Imperial College London
Department of Computing

Dense Visual SLAM

By

Richard A. Newcombe

December 2012

Supervised by
Prof. Murray Shanahan
Prof. Andrew J. Davison

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in Computing of Imperial College London and the Diploma of Imperial College London.

Declaration of Originality

This thesis is my own work and describes my own research except where explicitly indicated.

Copyright Declaration

Abstract

Visual SLAM systems aim to estimate the motion of a moving camera together with the geometric structure and appearance of the world being observed. To the extent that this is possible using only an image stream, the core problem that must be solved by any practical visual SLAM system is that of obtaining correspondence throughout the images captured. Modern visual SLAM pipelines commonly obtain correspondence by using sparse feature matching techniques and construct maps using a composition of point, line or other simple geometric primitives. The resulting sparse feature map representations provide sparsely furnished, incomplete reconstructions of the observed scene.

Related techniques from multiple view stereo (MVS) achieve high quality dense reconstruction by obtaining dense correspondences over calibrated image sequences. Despite the usefulness of the resulting dense models, these techniques have been of limited use in visual SLAM systems. The computational complexity of estimating dense surface geometry has been a practical barrier to its use in real-time SLAM. Furthermore, MVS algorithms have typically required a fixed length, calibrated image sequence to be available throughout the optimisation — a condition fundamentally at odds with the online nature of SLAM.

With the availability of massively-parallel commodity computing hardware, we demonstrate new algorithms that achieve high quality incremental dense reconstruction within online visual SLAM. The result is a live dense reconstruction (LDR) of scenes that makes possible numerous applications that can utilise online surface modelling, for instance: planning robot interactions with unknown objects, augmented reality with characters that interact with the scene, or providing enhanced data for object recognition.

The core of this thesis goes beyond LDR to demonstrate fully dense visual SLAM. We replace the sparse feature map representation with an incrementally updated, non-parametric, dense surface model. By enabling real-time dense depth map estimation through novel short baseline MVS, we can continuously update the scene model and further leverage its predictive capabilities to achieve robust camera pose estimation with direct whole image alignment. We demonstrate the capabilities of dense visual SLAM using a single moving passive camera, and also when real-time surface measurements are provided by a commodity depth camera. The results demonstrate state-of-the-art, pick-up-and-play 3D reconstruction and camera tracking systems useful in many real world scenarios.

Acknowledgements

There are key individuals who have provided me with all the support and tools that a student who sets out on an adventure could want. Here, I wish to acknowledge those friends and colleagues, that by providing technical advice or much needed fortitude, helped bring this work to life.

Prof. Andrew Davison's robot vision lab provides a unique research experience amongst computer vision labs in the world. First and foremost, I thank my supervisor Andy for giving me the chance to be part of that experience. His brilliant guidance and support of my growth as a researcher are well matched by his enthusiasm for my work. This is made most clear by his fostering the *joy of giving live demonstrations* of work in progress. His complete faith in my ability drove me on and gave me license to develop new ideas and build bridges to research areas that we knew little about. Under his guidance I've been given every possible opportunity to develop my research interests, and this thesis would not be possible without him.

My appreciation for Prof. Murray Shanahan's insights and spirit began with our first conversation. Like ripples from a stone cast into a pond, the presence of his ideas and depth of knowledge instantly propagated through my mind. His enthusiasm and capacity to discuss any topic, old or new to him, and his ability to bring ideas together across the worlds of science and philosophy, showed me an openness to thought that I continue to try to emulate. I am grateful to Murray for securing a generous scholarship for me in the Department of Computing and for providing a home away from home in his cognitive robotics lab.

I am indebted to Prof. Owen Holland who introduced me to the world of research at the University of Essex. Owen showed me a first glimpse of the breadth of ideas in robotics, AI, cognition and beyond. I thank Owen for introducing me to the idea of continuing in academia for a doctoral degree and for introducing me to Murray.

I have learned much with many friends and colleagues at Imperial College, but there are three who have been instrumental. I thank Steven Lovegrove, Ankur Handa and Renato Salas-Moreno who travelled with me on countless trips into the unknown, sometimes to chase a small concept but more often than not in pursuit of the bigger picture we all wanted to see. They indulged me with months of exploration, collaboration and fun, leading to us understand ideas and techniques that were once out of reach. Together, we were able to learn much more.

Thank you Hauke Strasdatt, Luis Pizarro, Jan Jachnick, Andreas Fidjeland and members of the robot vision and cognitive robotics labs for brilliant discussions and for sharing the

thrill of seeking the next real-time demonstration! I thank Thomas Pock and Christopher Zach for guiding me at an important time towards understanding a different world of computer vision. I'm extremely grateful to MSR Cambridge for my fruitful research internship with Shahram Izadi and colleagues. At MSR I was given early access to the first commodity depth camera that was about to create a revolution in computer vision.

Prof. Andrew Zisserman and Prof. Daniel Rueckert provided me with precise feedback on examining my thesis for which I am grateful, those remarks led directly to a more in-depth analysis of the techniques I demonstrate. I am extremely appreciative to Daniel Canelhas, Simon Fuhrmann, and my Dad, Christopher, who each provided brilliant proof reading of chapters. Thank you for wading through original unedited regions and for your useful comments.

My time in London now feels like the most amazing dream. I thank my long suffering friends from Flowers Mews and long before that: Renzo, Sarah, Pouria and Steven. They filled my time with immense fun and closeness that were the very best reasons to leave the lab at night and return to our great old house at the bottom of the hill.

I thank Jessica Robles for being on this adventure with me throughout these years and for her intimate understanding of the paths we have chosen and continue to have fun travelling on together.

Finally, I wish to state my gratitude for the love and unbounded support from my Mum and Dad, and for their continual interest in whatever I do. They have provided me with a place of tranquillity in my mind and a complete understanding that, no matter how difficult things may sometimes appear, a happy and loving home is always there for me. Whenever I wanted to escape the research, I was provided with tales from home of the growing adventures of my wonderful nieces, Libby and Alanna, brought into this world by my brother James and his wife Gemma. I thank all of you for letting me share that piece of home.

My wonderful colleagues, friends and family, you have filled these years with fun – I continue to be inspired by all of you!

I dedicate this thesis to my Mum and Dad
— my first teachers.

Contents

1

Introduction

Contents

1.1 Robot Perception

What are the limits on scene perception? This question has been investigated in a familiar form since the inception of the field of artificial intelligence in the middle of the 20^{th} century. We are captivated by the amazing capabilities of humans and other animals that effortlessly navigate and interact with the environment, sensing the world through their embodied visual, tactile and auditory modalities, which provide noisy and incomplete measurements of their environment. In a scenario where a robot needs to walk, drive or fly into a scene we want to know what should be inferred about the surrounding environment for the robot to achieve such navigation and interaction abilities; is there a generic way to represent the available information, and what are the limits on what can be inferred with the given limits in measurement quality and computing resources in practice?

Computer vision research tackles an aspect of one the most challenging pieces of the problem, trying to invert the image formation process that occurs when a $3D$ scene is projected into a camera creating its projected $2D$ image over time. The task of recovering the structure of the original scene given only passive images has resulted in decades of computer vision research trying to understand the fundamental nature of the inverse problem. Such an understanding is as important for unravelling the abilities found in animals using vision as for engineering solutions using computer vision since, while algorithms produced might be biologically implausible, the results demonstrate what is in principle retrievable from the data given a defined set of assumptions. New understanding has been interleaved with massive engineering efforts which, together with the explosion in computing resources, have demonstrated inspiring results. Solutions to the scene inference problem have been central to such feats as mapping at street level cities and towns across the planet to the advent of robust driver-less vehicles.

A robot, such as a driver-less vehicle, must solve a tougher version of the scene inference task that relates more acutely to the challenges faced by animals in that the problem must be continuously solved on-line in real-time. It would be of little use if a robot car coming to a particularly busy junction took several seconds to analyse the scene if an imminent danger required action within the next second: it must swerve to miss the cyclist!

The core perception challenge for any autonomous robot whose main task is navigation comprises two characteristic problems: environment mapping and robot localisation. To solve the robot localisation problem using only on board sensors a map must be available from which the robot can locate its relative location. But to be able to extend a map the robot must *know* its current relative location. This fundamental simultaneous localisation and mapping (SLAM) problem has been at the center of decades of robotics research.

Removing the complexities of a complete robot but keeping the core problem intact, researchers in computer vision and robotics have attempted to understand how to solve the visual SLAM problem where given only a sequence of images obtained from a moving camera, a scene reconstruction together with a current camera pose can be estimated. The most widely investigated class of visual SLAM systems estimate the pose of the cameras together with a sparse $3D$ point based representation of the scene, illustrated in Figure (1.1). The utilisation of a sparse point cloud model being sufficient for recovery of the real-time camera pose whilst also being computationally efficient to obtain and update. However, whilst such a sparse scene representation suffices for conservative forms of robot navigation, it is insufficient for the majority of future applications that will require some form of *scene interaction*.

Within a model based robotics paradigm it is clear that successfully achieving a particular

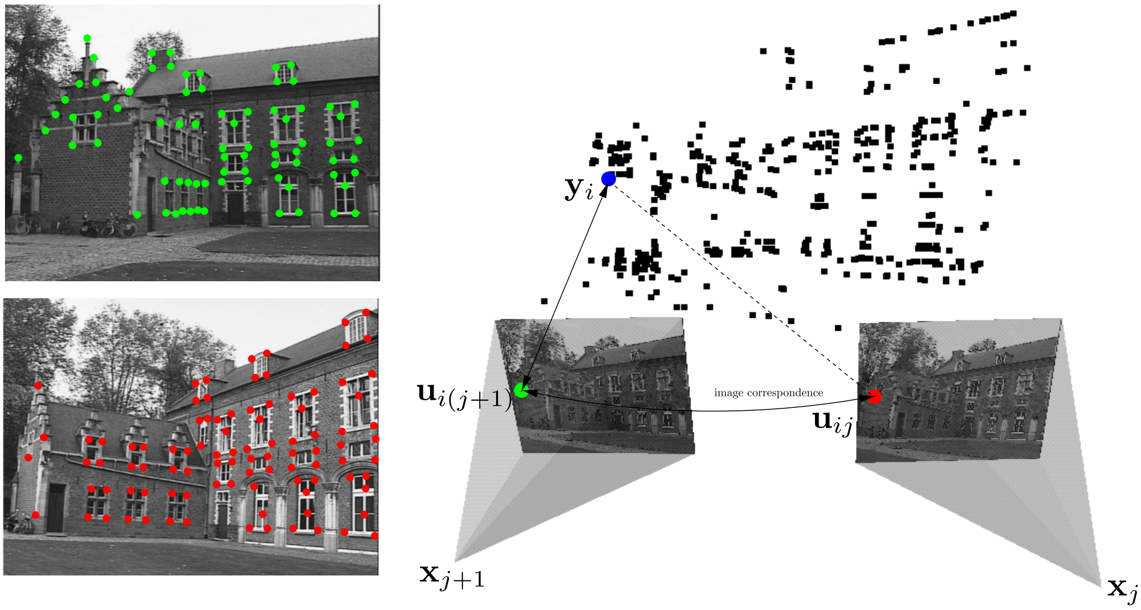


Figure 1.1: Illustration adapted from Pollefeys et al. (1999) of the resulting point cloud and camera pose estimates from a sparse feature based visual SLAM or SFM pipeline consisting of extracting and matching observations of points $\mathbf{u}$, across frames (2-views are shown here). Joint estimation of the camera poses $\mathbf{x}$, and point cloud structure $\mathbf{y}$, is performed using global optimisation, minimising the distance between predicted image points and observations. The resulting point cloud structure and camera poses are used by the authors in offline 3D model reconstruction.

task within the environment is to a large extent contingent on the availability and richness of information, represented in the model of the scene, necessary to achieve that task. If the task is to pick up an object on a table, the modelled scene must enable a planning algorithm to take into account the interaction between the object being manipulated and an end effector, such as information about the objects surfaces delineated from the reset of the scene. A probing question is *should object recognition come before interaction with the scene?* While it is the case that given knowledge that the object is a cup can lead to a greatly simplified interaction through expert knowledge on the subject of cup lifting, it should not be a necessary condition for interaction. More interestingly, the problem of interacting with unrecognised surfaces is clearly a prior condition on learning novel instances for object recognition in the robots future. Understanding what can be represented and inferred from a stream of images without strong prior assumptions on what is present is of major importance in building robots capable of exploring unknown environments and dealing with the challenges of a more general form of novel interaction with the world.

A point cloud is an impoverished representation of the world. In this work take a step towards understanding the limits of scene inference in the on-line setting by moving beyond the sparse point cloud representation of scenes to a denser representation of surfaces enabling much more of the image data available in the sensor stream to be used in both the mapping and localisation problems of visual SLAM. Denser SLAM leads to more robust continuous tracking, and ultimately produces richer predictive models useful in both robotics and augmented reality applications.

In the remaining sections of this introduction we outline the standard sparse visual SLAM approach used to obtain a point-cloud model of a scene together with the live camera pose estimate. We then illustrate the problems that arise in using a sparse scene representation and sketch three core components of an alternative dense visual SLAM system that uses a surface model representation of the scene in an attempt to capture and utilise all of the available information in the video stream. Finally we outline the contributions of this thesis in achieving a progression from sparse to dense visual SLAM.

1.2 Sparse Visual SLAM

In this section we first look at the specific formulation of the localisation and mapping problem previously illustrated in Figure (1.1). We assume that we are given M input images acquired from different locations overlooking a scene. In theory the images might have been acquired from M different cameras placed about the scene, but here we assume that they were captured from a single moving camera at different times.

We now make an abstraction from the image data: we model the structure of a scene as N *3D points* which can be partially observed in the M images. The projection of a scene point $\mathbf{y}_i \in \mathbb{R}^3$, into a camera with 6DoF pose $\mathbf{x}_j \in SE_3$ results in an image point $\bar{\mathbf{u}}_{ij} \in \Omega \subset \mathbb{R}^2$ that *could* be observed in that camera. If a measurement of the predicted point is actually observed, $\mathbf{u}_{ij} \in \Omega$, the error induced between the predicted and observed point is:

$$\Delta u_{ij} = \bar{\mathbf{u}}_{ij} - \mathbf{u}_{ij} \ . \tag{1.1}$$

In probabilistic terms, the probability density function over the error is often assumed to be a multivariate Gaussian distribution with diagonal covariance matrix $\sigma_{ij} \in \mathbb{R}^{3\times 3}$:

$$p(\bar{\mathbf{u}}_{ij}|\mathbf{x}_j, \mathbf{y}_i) \propto \exp(\frac{1}{2}\Delta u_{ij}^{\top} \sigma_{ij}^{-1} \Delta u_{ij}) \ . \tag{1.2}$$

If we further assume that observing multiple scene points across multiple cameras is an independent process, then for structure and motion parameters, $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_i, \dots, \mathbf{x}_M\}$, $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_i, \dots, \mathbf{y}_N\}$ and valid observations $\mathbf{u} = \{\mathbf{u}_{ij}|c_{ij} = 1\}$, the probability density function over all observations is:

$$p(\bar{\mathbf{u}}|\mathbf{x},\mathbf{y}) \propto \prod_{i=1}^{N}\prod_{j=1}^{M} p(\bar{\mathbf{u}}_{ij}|\mathbf{x}_j,\mathbf{y}_i)\ , \tag{1.3}$$

where we have used $c_{ij} = 1$ to indicate that camera j did observe point i. The core of the sparse visual SLAM pipeline attempts to estimate the unknown structure and motion parameters given the available observations. By Bayes rule we have:

$$p(\mathbf{x},\mathbf{y}|\bar{\mathbf{u}}) \propto p(\bar{\mathbf{u}}|\mathbf{x},\mathbf{y})p(\mathbf{x},\mathbf{y})\ . \tag{1.4}$$

Here $p(\mathbf{x},\mathbf{y})$ is prior over the structure and motion parameters. Hence, the most likely structure and motion parameters can be estimated by maximising the posterior distribution given in Equation (1.4). In practice, we can minimise the energy function resulting from the negative log of $p(\mathbf{x},\mathbf{y}|\bar{\mathbf{u}})$, known as bundle adjustment:

$$\hat{\mathbf{x}}_j,\hat{\mathbf{y}}_i \overset{MLE}{=} \min_{\mathbf{x}_j,\mathbf{y}_i} \sum_{i=1}^{N}\sum_{j=1}^{M} \begin{cases} \psi\left(\Delta u_{ij}\right) & \text{iff } c_{ij} = 1 \\ 0 & \text{otherwise}\ . \end{cases} \tag{1.5}$$

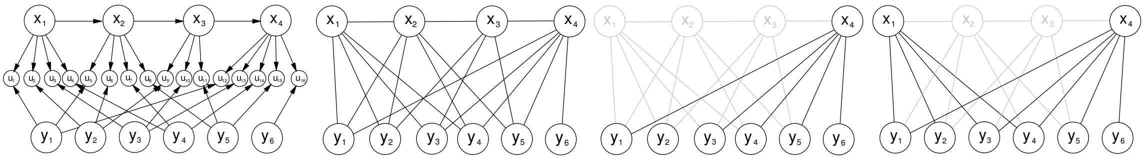
Here $\psi(\Delta u_{ij}) = \Delta u_{ij}^{\top}\sigma_{ij}^{-1}\Delta u_{ij}$ for the Gaussian distributed observation error in Equation (1.2), and in general is a positive penalty function designed to be robust to outliers in the point correspondences, ideally matched to the error distribution of the observations. Optimisation over the parameters is performed using a non-linear iterative minimisation scheme, requiring an initial estimate of the point positions and camera poses[1].

Given only the original input images a number of challenges arise: (1) parameter initialisation or bootstrapping, since the above non-linear optimisation is generally non-convex in the parameters an initial estimate of the structure and motion variables is required; (2) obtaining correspondence of the observed points across multiple images: the above bundle adjustment made the assumption that point correspondences were available, but initially we only have photometric image data; (3) timely optimisation of the resulting bundle adjustment Equation (1.5). We now look more closely at these challenges.

From Bundle Adjustment to Online Visual SLAM

In particular, it is of fundamental importance for many real world applications of visual SLAM that a live camera pose can be estimated in real-time. Shown in Figure (1.2a), a Bayesian network can be used to represent the causal relationships between a camera with pose $\mathbf{x}_j$ viewing the scene geometry abstracted to a point $\mathbf{y}_i$ resulting in a 2D point

[1] Equation (1.5) can be further extended to take into account a prior assumption about the smoothness of the camera trajectory.



(a) Bayesian Network (b) Markov Random Field (c) Probabilistic Filter (d) Keyframe BA

Figure 1.2: Graphical representations of the visual SLAM problem adapted from **Dellaert and Kaess** (2006) and **Strasdat et al.** (2010). The causal relationship between scene structure $\mathbf{y}_i$, camera poses $\mathbf{x}_j$ and the resulting observations $\mathbf{u}_{ij}$ is captured in a Bayesian network (a). The same problem can be expressed in an undirected Markov random field (b). The number of pose variables grows linearly over time and presents a computationally infeasible inference task for an online application of SLAM. Approximations to the full inference task are therefore required if estimation of the newest camera pose and observed structure is to be achieved: filtering solutions jointly estimate the full structure variables together with the live camera pose only (c); and keyframe based bundle adjustment approaches prune away all but a select number of judiciously chosen pose variables, which together with the observed scene structure and live camera state present a sparser inference task that can be solved in an available window of time.

observation $\mathbf{u}_{ij}$, captured in the probabilistic model from Equation (1.3), the graph also shows each new pose can be further constrained by the previous pose state. The *full* SLAM problem aims to solve for poses and scene points given a set of observations, as presented in the bundle adjustment problem. A further graphical abstraction shown in Figure (1.2b), removes the observations from the graph, and presents the constraints between a given camera pose and the scene points that it observed. This undirected Markov random field (MRF) is a direct representation of the structure of the bundle adjustment problem in Equation (1.5) where for $c_{ij} = 1$, each edge between a point i and pose j in the graph is associated with a reprojection error under a given penalty function.

To achieve online SLAM operation, strategies are required to reduce the linearly growing number of pose variables in the inference task, which quickly makes estimation of the latest camera pose and observed scene structure an impossibly expensive computation to be achieved in a fixed window of processing time. Illustrated in Figure (1.2c), probabilistic filters exploit the modelled Markov chain governing the camera state over time, enabling the inference to be written in a recursive form involving estimation over only the newest camera pose and the complete scene structure. An alternative solution shown in Figure (1.2d) instead maintains a sparse subset of the camera poses known as keyframes, together with the structure variables co-observed by these frames, maintaining a globally consistent joint estimate over the scene structure and keyframe poses using bundle adjustment. Online estimation of the current camera poses is then performed relative to the currently estimated structure given estimated correspondences between the scene model points and the image.

The Correspondence Problem

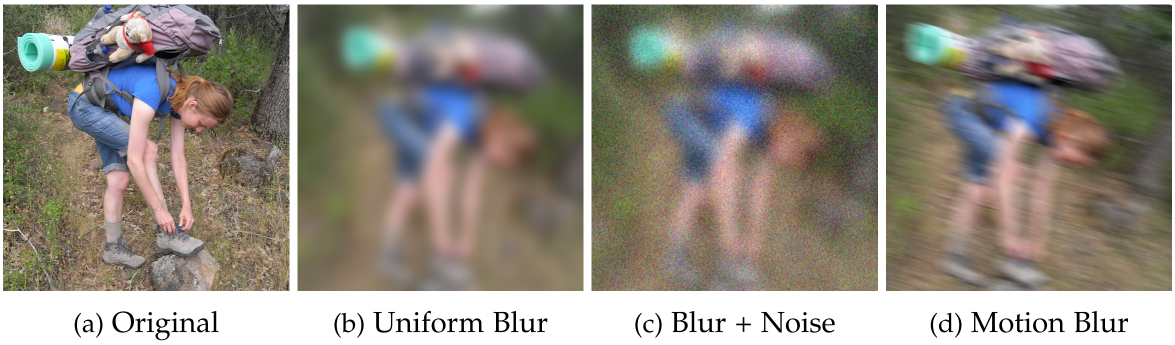
The correspondence problem appears in a sparse visual SLAM system in two different forms. The primary problem of obtaining $2D - 2D$ or image to image correspondence without knowledge of the camera poses and takes the form of tracking $2D$ image points across 2 or more views. This occurs in bootstrapping of both the structure and camera parameters, where the challenge of obtaining a $3D$ point estimate simultaneously with an estimate of the camera poses must be achieved first. We omit the techniques from this introduction for brevity, but note that solutions to the bootstrapping problem are an essential component of a fully automated visual SLAM system, and constitute one of the great achievements in computer vision. A treatise on the subject of visual geometry is available in **Hartley and Zisserman** (2004).

Assuming an estimate of $M \geq 2$ camera poses has been achieved, the bootstrapping problem continues for insertion of new structure points when there is gross uncertainty over the camera poses. When knowledge of the associated camera poses is available the space of correspondences of a feature in a second image is restricted to lie on the projection of the ray from the first images projection center and through the corresponding pixel called an epipolar line. Given correspondence between 3 or more views in total the constraints formed by the intersection of the epipolar lines resolves to a 3D point estimate of the observed geometry, which can then be projected into any other view with known pose.

The secondary problem is $3D - 2D$ or model to image correspondence, also called data-association, where given possibly incomplete knowledge of both the $3D$ point location $\mathbf{y}_i$ and a camera pose $\mathbf{x}_j$, correspondence is sought to obtain the $2D$ observation $\mathbf{u}_{ij}$. Typically, given the uncertain camera and structure estimates a restricted region within the image is formed within which the correspondence is expected to be observed.

Feature Detectors and Image Descriptors

A standard approach to obtaining $2D - 2D$ image correspondences follows a feature detection, description and matching framework. Given a first image a sparse selection of image locations is chosen at which local image patches are transformed into image *descriptors*. Finally, given $M \geq 2$ other images with associated descriptors, correspondence is obtained by searching for matching image descriptors amongst the frames. Data association or $3D - 2D$ correspondence is achieved in a similar way. Descriptors associated with a $3D$ model point are typically extracted from an image obtained when solving for the initial map point using $2D - 2D$ correspondences.



(a) Original (b) Uniform Blur (c) Blur + Noise (d) Motion Blur

Figure 1.3: Example synthetic motion with degraded images. The original image (a) is geometrically transform by a small in plane rotation and translation followed by three degradations to form images (b-d).

The goal of the feature detection, description and matching pipeline is to efficiently maximise the likelihood of valid correspondences to localisable $3D$ scene points in co-observing frames whilst minimising false correspondence occurrences for the space of image transforms over which the feature is to be observed. The general class of transforms for a static scene includes geometric (projective) distortion of the image region, and radiometric changes in appearance due to non-Lambertian materials, but in practice might include movement in the scene and dynamic lighting. We note that if ψ is chosen as a robust cost function, bundle adjustment minimisation can be made robust to erroneous correspondences and multiple data associations where a descriptor is matched in several mutually incompatible image locations.

1.3 Problems with Sparse Description

For the moment let us ignore the prevailing problem that arises when using sparse point cloud models in applications that instead require a dense surface estimate, and turn our attention to a key problem associated with robustness in the sparse visual SLAM algorithm. We show an image in Figure (1.3a) which is then rotated and translated prior to applying uniform blur, noise and motion blur to simulate degradation in the transformed image. A simplified version of the pose estimation problem can be illustrated with these images. Here we want to find the in-plane rotation and translation between the original image and one of the transformed images and to do using the sparse visual SLAM approach we must obtain correspondences between the images.

Since the sparse visual SLAM pipeline depends on the abstraction of the point features and their $2D$ correspondence it is crucial for both the image-to-image or model-to-image correspondence problem that such features can be reliably detected. This first step is used

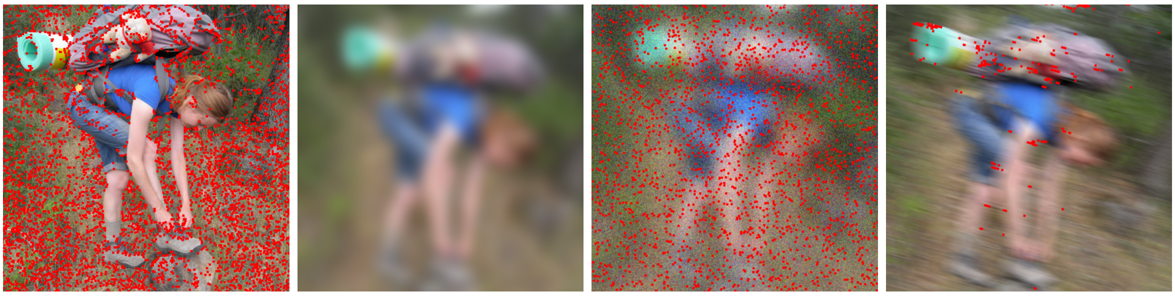


Figure 1.4: Result of running *FAST* corner detection on the images in Figure (1.3) using fixed thresholds. Typically, FAST is run over a multi-scale decomposition of the image to mitigate the effects image blurring in the original image.

to reduce the number of possible image locations down to a set that might be easily redetected under typical geometric and photometric image transforms. We demonstrate this using the popular FAST algorithm by **Rosten and Drummond** (2006) applied to the original and transformed images, the result is given in Figure (1.4). Unfortunately, as can be seen by comparing extraction on the original image with the degraded versions, the features detected vary considerably. Given an understanding of the feature detection mechanism it is not a surprising demonstration, and preprocessing of the image data together with tweaks to the detection algorithm parameters can be performed to improve repeatability of the detected locations.

Given detected feature locations, image regions must then be described and matched across frames. The vast majority of descriptors assume a locally planar surface around the feature point and while progress has been made in obtaining robust invariance of such patches across the geometric and photometric transformations typical in real images such invariance comes at cost, typically requiring larger image regions. In Figure (1.5) we demonstrate the result of running a popular feature detection, description and matching pipeline using the scale invariance feature transform (SIFT) keys developed by **Lowe** (2004) [2]. This example demonstrates that the extraction and matching pipeline produces sparser matching when faced with the degraded images. Again, this is not surprising since the feature extraction stage will produce fewer candidates for description and matching given the uniform and motion blurred images. However, it is exactly this need to find good parameters for the thresholds used in the extraction, description and matching pipeline, which often differs drastically depending on image quality, that can result in total failure of a real-time sparse visual SLAM system.

[2] Correspondences were computed using the software accompanying **Lowe** (2004) available from http://www.cs.ubc.ca/~lowe/keypoints/.

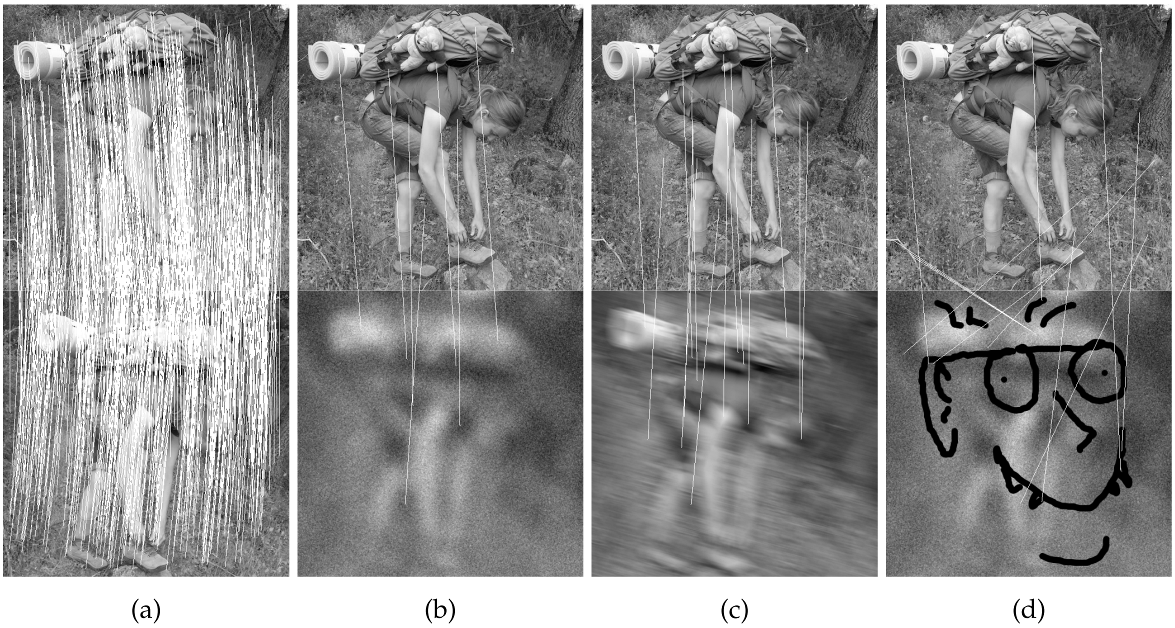


Figure 1.5: Example correspondences obtained by extracting and matching with SIFT, (**Lowe**, 2004). Each column shows the correspondences computed between the original image in Figure (1.3a) with a version of the transformed original image. (a) shows correspondences with a transformed original image without further degradation. Matching with the transformed and blurred noisy image is shown in (b), and with the motion blurred version in (c). In (d) we show matching against the blurred noisy image where we have added strong occluding outliers to the data. We note that while there are no correct descriptor correspondences in (d) for the given descriptor matching scheme, it could still be possible to estimate the correct transformation by using a RANSAC (**Fischler and Bolles**, 1981) style estimation of the transformation from a single inlier (since SIFT also encodes planar rotation information). Incorporating the knowledge of the manifold on which the correspondences exist would enable a more informative correspondence criteria.

1.4 Direct Approach: Dense Tracking and Mapping

We now break visual SLAM into its two characteristic problems. First, camera tracking: *Given a map, can we obtain the current camera pose?*. The simultaneous problem to be solved is: *Given known camera poses can we obtain an updated map?*. In the following three subsections we will look at alternative mechanisms that achieve camera tracking and structure estimation for visual SLAM but which do not make use of the explicit feature extraction and matching pipeline. Instead each component will formulate the tracking and mapping problem in a direct manner working directly over a function of dense image data available from the moving camera.

1.4.1 Dense Tracking

Given a map, can we obtain the current camera pose? In the previous section we looked at obtaining correspondences between frames required for estimation of the relative transform which can be solved with a non-linear least squares estimation over the parameter space:

$$\hat{\xi}_{ba} \overset{MLE}{=} \min_{\xi} \sum_{i=1}^{N} \psi \left(\mathbf{w}_{SE2}(u_a^i, \xi) - u_b^i \right) \ , \tag{1.6}$$

Here we defined the *warp function* $\mathbf{w}_{SE2}(u, \xi)$ which takes a pixel $u_a \in \Omega \subset \mathbb{R}^2$ in the frame of reference from image a and transforms it to a pixel u_b in image b using transform parameters ξ. The N explicitly given correspondences are defined in pairs between the frames (u_a^i, u_b^i). For the frame to frame tracking considered in figure (1.3), the warp function is an $SE2$ transformation parametrising a $2D$ translation and in plane rotation of the image.

If instead we formulate the inter frame motion estimation problem to directly optimise over the image intensity functions for frames and a and b we can remove the need to perform sparse feature extraction and matching. The formulation computes a cost over all pixels in the reference image $\mathcal{I}_a$:

$$\epsilon = \sum_{u \in \Omega} \psi \left(\mathcal{I}_b(\mathbf{w}(u, \xi)) - \mathcal{I}_a(u) \right) \ . \tag{1.7}$$

Here, an image interpolation function enables sub-pixel intensity values to be computed. The warp function together with image interpolation constitutes a generative model, and can take any form that predicts an image measurement from a set of parameters. As in the sparse bundle adjustment case, the direct approach can also utilise a robust error function to match a modelled likelihood of a potentially noisy observation. Figure (1.6) illustrates the shape of the cost function between the original image and three of the transformed and distorted counterparts used in SIFT key demonstration. In each of the presented cases a clear cost function minimum exists representing the solution to the optimisation problem. In particular, the image pair used in Figure (1.5d) that includes outliers in the data resulted in no correct correspondences using sparse feature extraction and matching pipeline, which precludes recovery of the frame-frame transform using Equation (1.6). In contrast the direct image error cost function shows a useful minimum for the same image pair near the correct transform parameters, shown in Figure (1.6f). Therefore, while the non-convex nature of Equation (1.7) prevents a guarantee of convergence to the correct parameters when using a gradient descent style optimisation, in practice we can often achieve convergence to the minimum if the initial estimate of the parameters ensures we are within a basin of convergence. The assumption under which such direct optimisation in Equation (1.7) is performed is that corresponding pixel values have brightness constancy, that

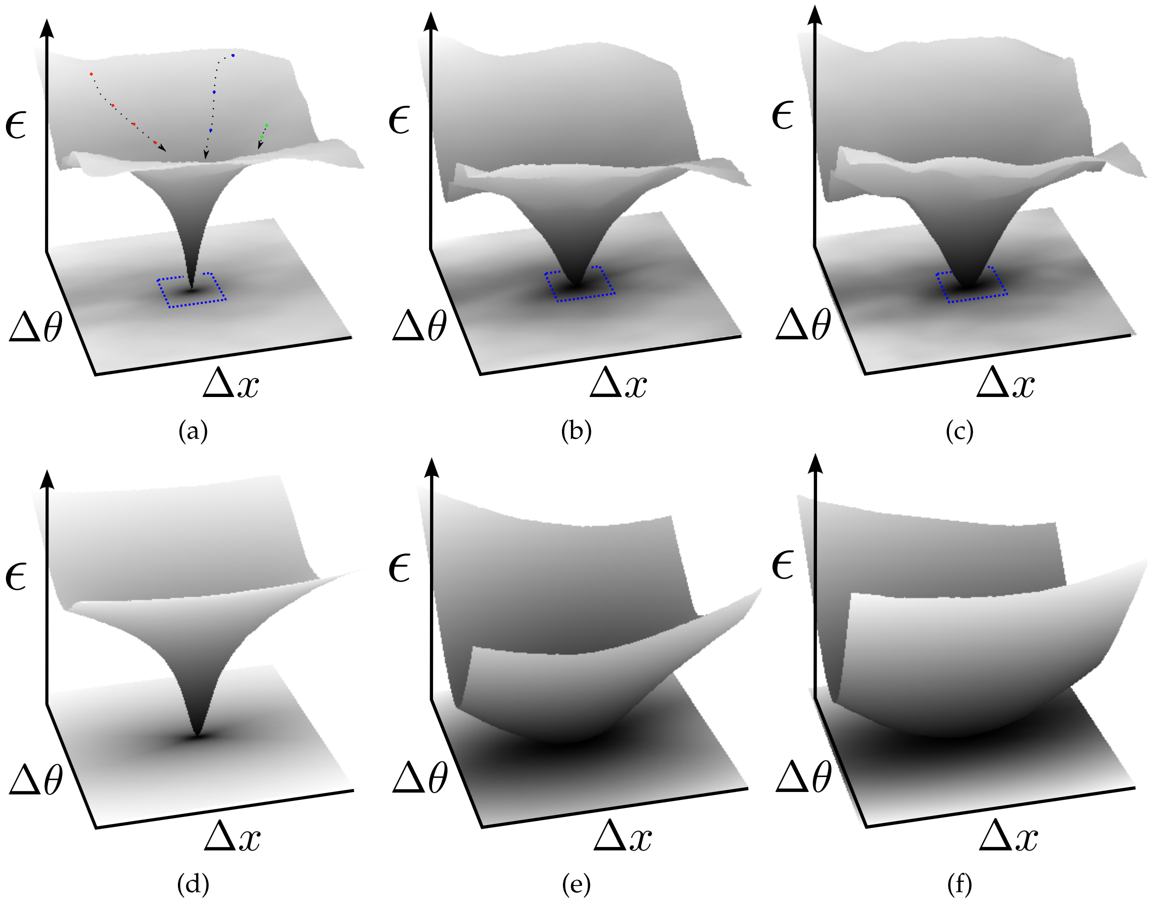


Figure 1.6: Cost function plots for the example transformed and degraded image pairs showing the parameter range with in-plane rotation $\pm\frac{\pi}{2}$ and translation $\pm 100 pixels$. (a) shows the cost as computed using the original image and its transformed version. In (b) between the original and transformed noisy blurred image (also shown in Figure (1.5b), and in (c) between the original and transformed noisy blurred image with outliers shown in Figure (1.5d). In (d-f) we show zoomed in plots for (a-c) with parameter range $\pm\frac{\pi}{20}, \pm 10 pixels$.

is $\mathcal{I}_b(\mathbf{w}(u, \xi)) \approx \mathcal{I}_a(u)$ given the correct parameters ξ, which although not generally true often holds in practice over short periods of time. Throughout this introduction we will assume this assumption does hold.

Direct Camera Tracking from a Dense Model

Assuming a static scene, a generative model can be formulated to compute an image prediction for an observation of a general scene under perspective projection. Assuming we have known perspective projection with calibration matrix K and the relative transformation between frames is ξ_{ba}, then using known geometry for example in the form of depth $\mathcal{D}(u) \in \mathbb{R}_+$ associated with each pixel u in image a, the pixel is transformed into frame b as:

$$\mathbf{w}_{proj}(u, \mathcal{D}(u), \xi_{ba}) = \pi\left(K\xi K^{-1}[u^\top; 1]^\top \mathcal{D}(u)\right) \ . \tag{1.8}$$

This generative model optimally describes the geometric distortions given the now 6DoF camera pose parameters ξ where the scene is jointly observable. As in the previous lower dimensional example, optimisation of the cost function can be performed through a form of gradient descent such as a non-linear least squares optimisation. The clear advantage of the direct approaches is that by minimising an image error instead of explicitly provided sparse correspondences we can hope to obtain a higher quality and more robust pose estimate, taking advantage of the massive redundancy in the image data without the preprocessing needed to achieve abstraction to $2D$ feature points.

It is important to note that researchers who pioneered the feature based tracking approaches originally asked the question *what makes a good feature for tracking?*, and modelled the problem as a sparsification of the direct tracking equation under a given geometric transform (i.e. affine (**Shi and Tomasi**, 1994)). Unfortunately such approaches are only successful for the class of transforms that can be encapsulated in the image intensity alone, since for optimal extraction under projective distortion of a non-planar scene the analysis must include knowledge of the surface resulting in a per frame optimal extraction of features.

1.4.2 Dense Correspondence and Depth Estimation

Given known camera poses can we obtain an updated map? The direct approach to camera tracking is only possible if the scene model is provided in the form of a dense surface model, enabling the warp function to densely predict the appearance of an image. We now look at how such a dense surface estimate can be computed without recourse to the sparse feature extraction and matching pipeline.

Given $M \geq 1$ correspondences from a pixel u_a in a reference frame to other frames where all poses are known, a new map point can be inserted by minimising the reprojection error of the $3D$ point that lies on the reference pixel ray with the constraint that the point must be in front of all observing cameras, illustrated in Figure (1.7). Using the previously defined warp function $\mathbf{w}_{proj}$ in Equation (1.8), we therefore parametrise the point as a depth $\mathcal{D}(u_a)$ in a *depth map* $\mathcal{D} : \Omega \mapsto \mathbb{R}_+$ in the reference frame:

$$\tilde{\mathcal{D}}(u) = \min_{d \in \mathbb{R}_+} \sum_{j=1}^{M} \psi\left(\mathbf{w}_{proj}(u_a, d, \xi_{ja}) - u_j\right) \ , \tag{1.9}$$

where u_j is the correspondence to pixel u_a found for frame j. Looking again at the extracted feature locations in the clean image in Figure (1.4), it can be seen that homogeneously textured image regions are feature sparse. From a mapping perspective this results in the insertion of new $3D$ points only in regions of high level of texture where features are detected.

We can instead attempt to obtain a dense correspondence field in a given reference image using a direct minimisation of photometric cost at each pixel. The photometric error for each pixel u in a reference image $\mathcal{I}_a$ and another frame $\mathcal{I}_b$, using the previously described parametrisation of the scene depth is:

$$\tilde{\mathcal{D}}(u) \overset{MLE}{=} \min_{d \in \mathbb{R}_+} \sum_{j=1}^{M} \psi\left(\mathcal{I}_b\left(\mathbf{w}_{proj}(u_a, d, \xi_{ja})\right) - \mathcal{I}_a(u_a)\right) \ . \tag{1.10}$$

Equation (1.10) takes the same form as the pose estimation cost function, simply fixing the previously unknown camera pose parameters and now estimating depth. The function is non-convex in the depth parameter, and so we are faced with two possibilities for its direct minimisation. The first route uses a gradient descent style optimisation. As described previously for the direct pose estimation, a non-linear iterative optimisation can be performed, linearising the photometric cost around a current depth estimate in each pixel to obtain a convex form that can then be minimised. An alternative method, taking advantage of the low dimensionality of the optimisation problem is simply to quantise $\mathcal{D}(u)$ into a finite number of depth hypotheses, from which the minimum of the cost function can be located by direct search. As the quantisation resolution tends to the sampling limit of the image functions this direct approach will result in finding the minimum of Equation (1.10). Figure (1.8) demonstrates the result of estimating the depth at each pixel in a reference frame by optimising the multi-view stereo cost function given a growing number of co-observing views for multi-view video dataset.

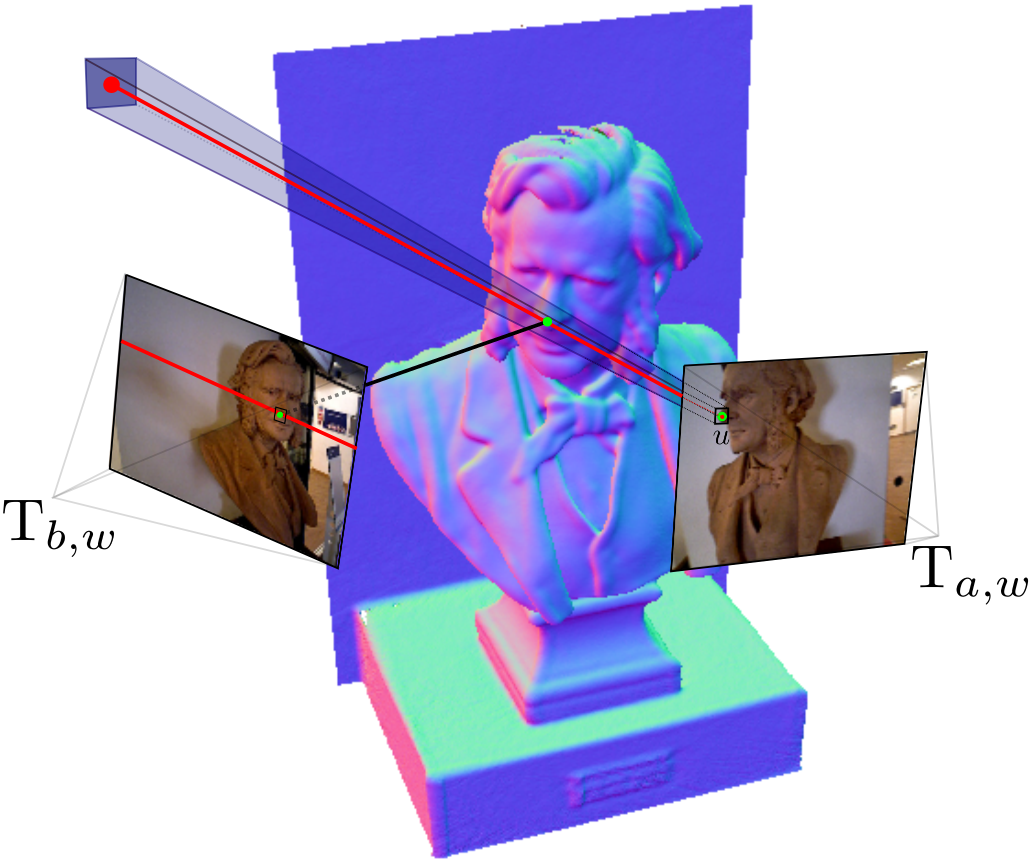
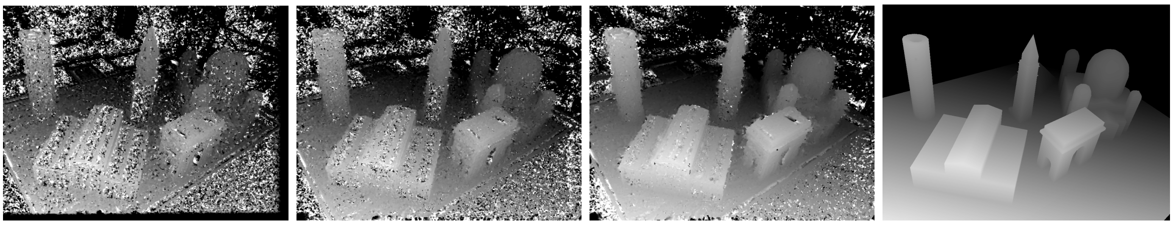


Figure 1.7: Multi-view stereo: Epipolar geometry constrains the correspondence problem for static scenes. The brightness constancy assumption for the pixel assumes that the correct surface location should result in a projection into co-observing frames which take on similar pixel values. Although a single pixel is not very discriminative in one view (since a similar value might be present along the epipolar line, shown here in red), it is more likely that across multiple views only the correct surface location should continue to project to pixels with similar values. This assumption can be wrong when the scene contains large homogeneous regions (of low or repeated textures).



(a) 2 views (b) 5 views (c) 20 views (d) Ground Truth

Figure 1.8: Estimated Multi-view depth map: Example per pixel minimum of Equation (1.10) using a quadratic function for ψ resulting in $\tilde{\mathcal{D}}$. Depth is illustrated with lighter values indicating points that are closer to the camera. Beginning with a two-view stereo data term (a), noise is reduced in the depth map as the number of views are increased (b-c). Noting the ground truth depth map(d), errors can be seen at depth discontinuities in the image and regions corresponding to surfaces with non-Lambertian materials breaking the brightness-constancy assumption. This example multi-view stereo estimation was performed on the Graz City of Sights video dataset presented in Chapter (4).

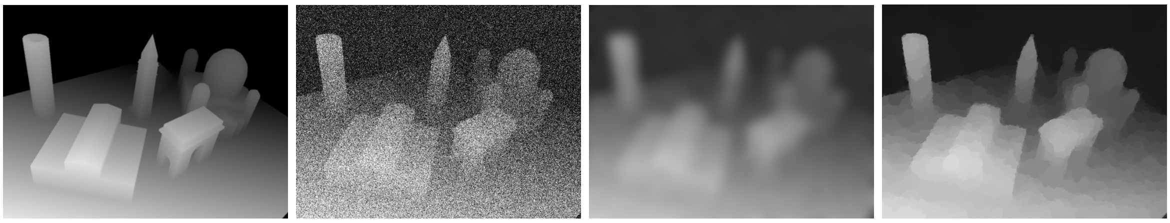
Global Optimisation

The depth estimation procedure discussed in the previous subsection yields erroneous surface estimates wherever the brightness constancy assumption does not hold, which occurs in a real world setting for a number of reasons including the presence of non-Lambertian surfaces and dynamic lighting in the scene as well as partial observability of a surface across the multiple views used in the optimisation. An insight into why the simple, per-pixel minimum, depth estimation procedure with the brightness constancy assumption falls short of obtaining higher quality surface estimates can be obtained by probabilistically modelling the process by which the images used were captured. In this subsection we now look to abstract from the image data and understand if, by modelling the noise present in the estimated depth maps, we can do better than the per-pixel minimum optimisation used to obtain the results in Figure(1.8). We note that a switch in notation is made in this subsection to enable the two co-ordinates of a pixel location to be referenced directly (as x, y).

Given a depth map (image) containing noise, we are interested in obtaining the denoised version (solution). We can model the formation of the noisy observed image $g : \Omega \mapsto \mathbb{R}$ as a degradation of the model solution $\mathcal{D}$ which is corrupted at each pixel $(x, y) \in \Omega$ by Gaussian noise with a variance σ^2:

$$g(x,y) = \mathcal{D}(x,y) + \mathcal{N}(0,\sigma) \ . \tag{1.11}$$

For example, let us assume that the depth map $\tilde{\mathcal{D}}$ in Equation (1.10) can be modelled by g. A statistical model of the forward process is called the likelihood, or statistical generative



(a) Model Depth $\mathcal{D}$ (b) $\mathcal{D} + \mathcal{N}(0, \mathbf{I}\sigma)$ (c) Quadratic (d) TV-Quadratic

Figure 1.9: Image denoising results for a depth map (a), corrupted by Gaussian noise (b). (c) Shows the resulting solution using a quadratic data term and regularisation with quadratic cost over first order edge magnitude, the result is an over smoothed solution. In (d) the TV-Quadratic denoising also uses the quadratic penalisation for the data term, but uses the $L1$ metric on the regularisation term, better matching the statistics of the 1^{st} order gradient smoothness. This results in better edge preservation in the solution, but also shows stepping artefacts. The solution is a piecewise constant function whereas the true solution is piecewise affine.

model. Specifically the above model states the independence of each pixel observation given the solution, and produces a conditional probability distribution:

$$p(g|\mathcal{D}) = \prod_{(x,y)\in\Omega} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(g(x,y) - \mathcal{D}(x,y))^2}{2\sigma^2}\right) . \tag{1.12}$$

We are interested in retrieving the noise free $\mathcal{D}$ given only the observation g and knowledge of the noise level σ^2, but unfortunately in its current form the problem is ill-posed; a number of possible solutions might exist but are not deterministically obtainable. The main tool in computer vision for obtaining a well-posed form of such problems is to restrict the space of possible solutions by introducing an image prior. A large class of priors have been investigated that assume the probability of the solution is proportional to its spatial smoothness. This is quite reasonable since we more often observe that the noise free depth map is composed of regions that vary smoothly across connected regions; scenes are composed of objects comprising surfaces that vary smoothly, changing more abruptly at object boundaries. Rather than jumping sporadically at pixel locations, a pixel depth therefore has an increased probability of taking on a similar value to its neighbours. A classic example is to model the prior as a Gaussian distribution over the magnitude of first order image derivatives $\|\nabla\mathcal{D}(x,y)\|$, with variance ν^2:

$$p(\mathcal{D}) = \prod_{(x,y)\in\Omega} \frac{1}{\sqrt{2\pi\nu}} \exp\left(-\frac{\|\nabla\mathcal{D}(x,y)\|^2}{2\nu^2}\right) . \tag{1.13}$$

With both the likelihood and prior specified we are now in a position to write the posterior distribution $p(\mathcal{D}|g)$ using Bayesian inference:

$$p(\mathcal{D}|g) = \frac{p(g|\mathcal{D})p(\mathcal{D})}{p(g)} \,. \tag{1.14}$$

Since $p(g)$ is independent of the solution u, $p(\mathcal{D}|g) \propto p(g|\mathcal{D})p(\mathcal{D})$ and the highest probability or maximum a posteriori (MAP) solution for the above model we find $\hat{\mathcal{D}}$:

$$\hat{\mathcal{D}} = \max_u \{p(\mathcal{D}|g)\} \tag{1.15}$$

$$\hat{\mathcal{D}} = \max_u \{p(g|\mathcal{D})p(\mathcal{D})\} \tag{1.16}$$

$$\hat{\mathcal{D}} = \max_u \left\{ \frac{1}{4\pi\mu\nu} \prod_{(x,y)\in\Omega} \exp\left(\frac{(g(x,y)-\mathcal{D}(x,y))^2}{\sigma^2} + \frac{|\nabla\mathcal{D}(x,y)|^2}{\nu^2} \right) \right\} \,. \tag{1.17}$$

We transform this probability maximisation problem into an energy minimisation form by setting $\mathsf{E}(\mathcal{D}) = -\ln p(\mathcal{D}|g)$:

$$\mathsf{E}(\mathcal{D}) = -\ln p(\mathcal{D}|g) \propto -\ln p(g|\mathcal{D}) - \ln p(\mathcal{D}) \,, \tag{1.18}$$

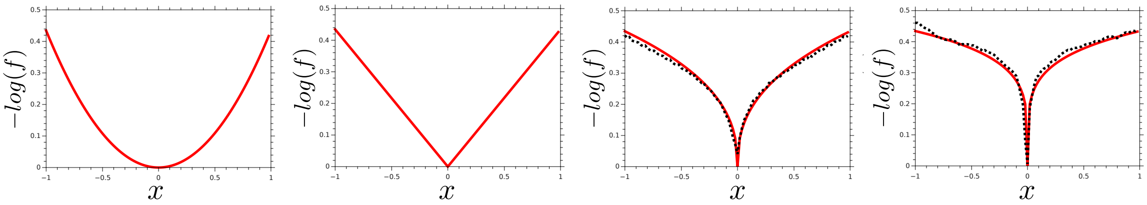
the interchange between $\max_{\mathcal{D}}\{p(\mathcal{D}|g)\}$ and $\min_{\mathcal{D}}\{\mathsf{E}(\mathcal{D})\}$ results in the maximum likelihood estimate:

$$\hat{\mathcal{D}} \overset{MLE}{=} \min_{\mathcal{D}} \left\{ \sum_{(x,y)\in\Omega} \left(\frac{1}{2}((g(x,y)-\mathcal{D}(x,y))^2 + \frac{1}{2\lambda}\|\nabla\mathcal{D}(x,y)\|^2 \right) \right\} \tag{1.19}$$

where λ combines factors relating to the variances ν^2, σ^2. $E(\mathcal{D})$ in Equation (1.19) is a sum of convex functions, and is therefore also convex. This is important since it ensures that there is a globally achievable solution, $\hat{\mathcal{D}}$. In the form of an energy in Equation (1.18), $-\ln p(g|\mathcal{D})$ is often called the data term while $-\ln p(\mathcal{D})$ is called the smoothness or regularisation term.

In Figure (1.9) we simulate the production of a noisy depth map by corrupting the ground truth depth map from Figure (1.8d) with Gaussian noise. Figure (1.9c) illustrates the resulting denoised solution depth map, obtained using the quadratic model described above. We can see that while noise is suppressed the depth boundaries are no longer sharp. Our probabilistic model contains two components that could be at fault: the prior or likelihood. However since in this synthetic example we have used an optimal likelihood model, given knowledge of the Gaussian form and σ, the problem lies in the image prior.

The prior model was arrived at by assuming a Gaussian distribution over first order solu-



(a) $\alpha = 2.0$ (b) $\alpha = 1.0$ (c) $\alpha = 0.8$ (d) $\alpha = 0.2$

Figure 1.10: Plots for the $-log$ of four instances of the generalised Gaussian distribution $f(x)$, Equation (1.20). Shown in (a) and (b) for the Gaussian and the Laplacian distributions. In (c) and (d) we show the generalised Gaussian model fit (in red) for the $-log$ of two sampled probability distributions (dashed lines): (c) for the gradients from a grey-scale image $\nabla\mathcal{I}$, captured from a moving camera, while in (d) for the corresponding depth image gradient $\nabla\mathcal{D}$, captured from the same camera trajectory taken over the course of a minute of scene browsing in an office environment.

tion smoothness, but *is a Gaussian a good match for the true distribution?* By capturing the real world derivative statistics for both natural images and depth images, researchers have found that a generalised Gaussian distribution with higher kurtosis and larger variance than is achieved with a Gaussian distribution provides a better fit:

$$f(x) \propto \exp(-\frac{|x-\mu|^{\alpha}}{\alpha\sigma^{\alpha}}) \ . \tag{1.20}$$

The Gaussian distribution is also captured with $\alpha = 2$, and yields the quadratic penalisation model through its negative log. To investigate what a more realistic distribution over derivatives is, we collected statistics over the pixel gradients of depth maps measurements for a dataset using a commodity structured light device in an office environment that includes cluttered desks and people within a working range of $0.4m$ to $4m$. Shown in Figure (1.10d), taking the negative log of the computed histogram over depth map gradients shows clearly that a non-convex penalty over the first order gradients is appropriate, with a good fit to the generalised Laplace distribution at $\alpha = 0.2$. Since a convex formulation confers serious advantages for fast global optimisation, an interesting prior model arises at the boundary between convex and non-convex where $\alpha = 1$, the closest convex model to the desired distribution, the Laplace distribution. The equivalent penalty function shown in Figure (1.10b) is an ℓ_1 norm. In comparison to optimisation under the quadratic penalisation, the ℓ_1 norm presents a robust cost which when applied as prior with the first order derivatives of the solution yields Total-Variation (**TV**) regularisation of the solution. The updated solution shown in Figure (1.9d) using the **TV** regularisation in combination with the previous Gaussian likelihood model shows an improvement in capturing the depth discontinuities whilst still suppressing noise.

Returning to the original dense correspondence problem, we are presented with two possible routes to increase the quality of the result. In the simplest case we can model the data term only depth map as a direct observation and define a likelihood model to describe the noise we observe, coupled with a suitable image prior. Alternatively we can go further back and model the likelihood directly over the photometric cost function and combine this with a smoothness prior.

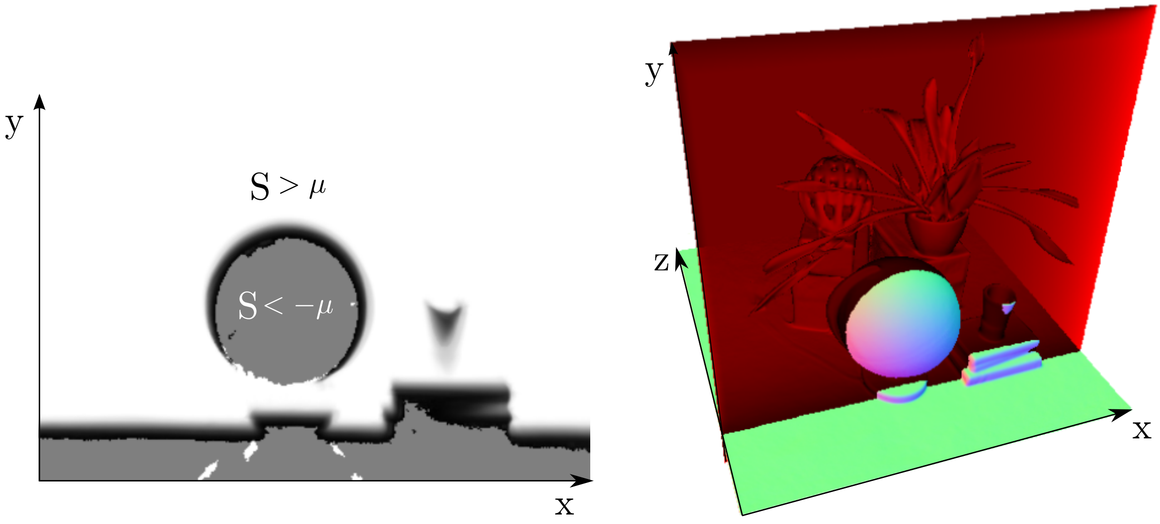
1.4.3 Combining Dense Depth Measurements into Dense 3D Maps

The surface measurement obtained by dense correspondence provides for each pixel an estimated $3D$ point in the camera frame of reference in which it was estimated, but as we have seen the depth maps contain errors. Moreover, each depth map covers only a partial view of the scene. In this subsection we assume that a number of depth maps have been computed from a moving camera browsing a scene. Given this set of calibrated depth maps, our task is to obtain a consistent map which explains these measurements. A point cloud representation of the scene, formed from the union of all depth measurements transformed into a global frame has limited use. Since points have neither direction nor area, a basic point cloud is unable to provide predictive capabilities such as surface visibility or occlusion in a given view. A simple approach to obtain a *surface* representation from a single depth map is to compute a triangle mesh by exploiting an assumed connectivity of neighbouring elements in the depth image. However, simply triangulating the set of depth maps can result in inconsistent reconstructions caused by connecting neighbouring depth map vertices which are not topologically connected on the real surface.

Fortunately, a depth measurement provides more than just an observation on the surface location in an image. It crucially also gives information about free space between the surface and the camera center. Assuming a Gaussian likelihood along a ray of measured depth, we can be relatively certain that the region in front of the measurement is free space, while our observation tells us nothing about the region behind the estimated surface past some threshold of uncertainty.

An extremely useful surface representation that enables the accurate representation of free space is the signed distance function (SDF) $S : \mathrm{R}^3 \mapsto \mathrm{R}$. Given a surface in $3D$ space, the signed distance function volumetrically defines the signed Euclidean distance $S(x)$ from a point in the volume $x \in \Lambda \subset \mathbb{R}^3$ to the nearest point on the surface, where the sign delineates regions of space that are closest to a front (positive distance) or back (negative distance) of the surface. The surface is therefore implicitly represented as the zero level set of the function, $S(x) = 0$.

In Figure (1.11a) we illustrate a *truncated* signed distance function (TSDF) representation for



(a) (b)

Figure 1.11: A slice through the truncated signed distance volume showing the truncated function $S > \mu$ (white), the smooth signed distance field around the surface interface $S = 0$ and voxels which are not defined with a valid signed distance $S < -\mu$.

the scene in Figure (1.11b). Unlike the SDF, the truncated version defines only a limited SDF near the surface interface and otherwise truncates the value where the unsigned distance is above a specified threshold, furthermore it also defines values which do not have a valid truncated SDF or SDF value through a second weighting function, defining the validity of TSDF value at each point in the volume. The importance of this truncated SDF will now be illuminated by its use in obtaining a global surface reconstruction from multiple calibrated depth measurements.

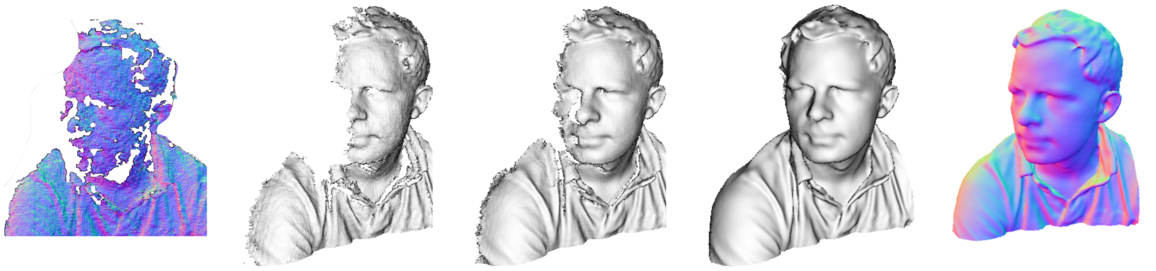
Let us assume that for any potentially noisy depth map $\mathcal{D}_i$ we have $\tilde{S}_i$, its TSDF. A solution to consistent model reconstruction can now be posed in terms of obtaining a denoised truncated signed distance volume S given m noisy overlapping depth map measurements in TSDF form. To that end, making the assumption that the surface measurements are independent:

$$p(\tilde{\mathcal{S}}_1, \tilde{\mathcal{S}}_i, \ldots, \tilde{\mathcal{S}}_m | S) = \prod_{i=1}^{m} p(\tilde{S}_i | S) \tag{1.21}$$

and for $\tilde{S}_i$, the likelihood $p(\tilde{S}_i(x)|S)$ is a Gaussian corrupted measurement of $S(x)$ independent of other points in $\tilde{S}_i$ with variance $\sigma_i^2(x) = 1/w_i^2(u)$,

$$p(\tilde{\mathcal{S}}_1, \tilde{\mathcal{S}}_i, \ldots, \tilde{\mathcal{S}}_m | S) = \prod_{x \in \Lambda} \prod_{i=1}^{m} p(\tilde{S}_i(x) | S) \tag{1.22}$$

Then for the simplest uniform prior over S, we can write a trivial posterior distribution



(a) Measurement (b) 2 Frames (c) 30 Frames (d) 100 Frames (e) Complete model

Figure 1.12: Example reconstruction using the truncated signed distance function averaging approach. (a) Shows the set of vertices computed from one depth image captured from a commodity structure light camera, rendered to show the surface normal orientation. In (b,c,d) the partially reconstructed surface is shown with Phong shading, computed from $2, 30$ and 100 calibrated surface observations of the subject acquired from a moving sensor. The complete model from approximately 20 seconds of modelling is shown in (e), rendered into the same camera pose as from measurement (a) highlighting the denoised and filled in reconstruction obtained.

over the desired surface:

$$p(S|\tilde{\mathcal{S}}_1, \tilde{\mathcal{S}}_i, \ldots, \tilde{\mathcal{S}}_m) \propto \prod_{x \in \Lambda} \prod_{i=1}^{m} \exp\left(-\frac{|S(x) - \tilde{S}_i(x)|^2}{2\sigma_i^2(x)} \right) , \tag{1.23}$$

Taking the negative logarithm of the distribution, we can obtain the maximum likelihood estimate of S by minimising the energy $\sum_{i=1}^{M} |S - \tilde{S}_i|^2$. Due to the independence assumptions made, given a discretisation over S this results in the weighted mean of the observations:

$$\hat{S} = \max_{S} p(S|\tilde{\mathcal{S}}_1, \tilde{\mathcal{S}}_i, \ldots, \tilde{\mathcal{S}}_m) \tag{1.24}$$

$$\hat{S} = \min_{S} \sum_{i=1}^{m} |S - w_i \tilde{S}_i|^2 \tag{1.25}$$

$$\hat{S} = \frac{1}{\sum_i^m w_i} \sum_{i=1}^{m} w_i \tilde{S}_i , \tag{1.26}$$

where $w_i : \Lambda \mapsto \mathbb{R}$ volumetrically defines the confidence over the TSDF values, $w_i(x) \propto \sigma_i^{-1}(x)$. We note the weighting function removes summation over regions of the TSDF which might not have a valid value simply by setting w_i in those regions to 0.

A useful property of the weighted mean is the ability to write it as update equation using a second cumulative weight volume $W : \Lambda \mapsto \mathbb{R}_+$,

$$\hat{S}_{i+1}(x) = \frac{W_i(x) + w_{i+1}(x)\tilde{S}_{i+1}(u)}{W_i(x) + w_{i+1}(x)} \tag{1.27}$$

$$W_{i+1}(x) = W_i(x) + w_{i+1}(x) \tag{1.28}$$

The effect of averaging truncated versions of the SDF representation limits averages to local regions in the multiple measurements. The narrow band of SDF represented values must therefore be small enough to eliminate interference between front and back facing surfaces, but must be wide enough to enable a useful average to be made between noisy observations of the same surface.

This simple update scheme enables a route to constant time dense mapping of calibrated depth measurements into an optimal surface reconstruction. A demonstration of this powerful technique is given in Figure (1.12). Depth map input to the example is produced from a Microsoft Kinect structured light depth camera with known sensor poses. Acquiring the many $100s$ of depth images used in the reconstruction from a moving camera, the measurements are integrated together and the resulting denoised surface is extracted as the zero level set of the weighted average TSDF.

1.5 From Sparse to Dense Visual SLAM

In the previous section we outlined tracking, depth estimation and surface reconstruction concepts that present approaches to solving separated localisation and mapping problems without the use of explicit feature extraction, matching and tracking. In particular we saw in Sections (1.4.1) and (1.4.2) the potential for direct optimisation of the pose and structure parameters (in the form of a depth map) to simplify the visual SLAM pipeline while providing a far richer mapping and potentially more robust tracking result. In this thesis we introduce an approximation to a recursive form of the SLAM posterior, that will enable us to use these dense tracking and mapping components in a straightforward way.

Given some form of an initial dense surface reconstruction $\mathbf{m}_{j-1}$ and a known camera pose $\mathbf{x}_{j-1}$, let us assume that we can alternate between estimation of a new camera pose $\mathbf{x}_j$, given the current dense map and that given that camera pose and the current map, we will estimate an updated map $\mathbf{m}_j$:

$$\begin{aligned} &p(\mathbf{x}_j, \mathbf{m}_j | \mathbf{x}_{\{j-1,\ldots,j-J\}}, \mathbf{Z}_{\{j,j-1,\ldots,j-J\}}) \approx \\ &p(\mathbf{x}_j | \mathbf{m}_{j-1}, \mathbf{Z}_j) \cdot p(\mathbf{m}_j | \mathbf{m}_{j-1}, \mathbf{x}_{\{j,j-1,\ldots,j-J\}}, \mathbf{Z}_{\{j,j-1,\ldots,j-J\}}) \ . \end{aligned} \tag{1.29}$$

Specifically this means that we will estimate a new camera pose $\mathbf{x}_j$, by exploiting the availability of a (partial) surface reconstruction $\mathbf{m}_{j-1}$. The availability of the dense model will enable us to use a direct tracking methodology. Then, given the surface model from $\mathbf{m}_{j-1}$, the new camera pose $\mathbf{x}_j$, its associated sensor measurement $\mathbf{Z}_j$ (i.e. a passive image) and a subset of J such historically calibrated estimated sensor measurements we will define a procedure to update a dense surface map of the environment $\mathbf{m}_j$. We will perform this optimisation by abstracting the input frames to surface measurements in the form of depth maps and optimally integrate these depth maps into a consistent global surface model.

Unfortunately, it is well known from SLAM research that such a partitioning of the joint distribution is not valid in general. This is due to the joint dependence of each measurement on pose and structure parameters as made explicit in the visual SLAM joint distribution from Equation (1.3). Typically we would expect error in pose estimates to ultimately lead to irreversible camera drift and errors being baked in the map, and we know that such an approximation in Equation (1.29) can only be valid if the estimates from the mapping and tracking components result in optimal parameters at each alternating step. It is therefore an interesting and surprising result of this thesis that we will demonstrate that the approximation does often hold in practice, at least for smaller maps. We note that scalable SLAM solutions exist based on sub-mapping and graph optimisation techniques (**Thrun et al.**, 2005), therefore we make use of this simplest of SLAM partitioning and focus on obtaining real-time *dense* SLAM that will enable new applications in augmented reality and robot interaction.

Selected Publications

The real-time systems developed during the course of this thesis contribute a progression of dense visual SLAM systems. In early work, we augmented feature-based visual SLAM point maps with a denser surface model composed of overlapping surface meshes obtained by computing dense correspondences between frames. This enabled real-time geometry aware augmented reality with a single moving camera: *Live Dense Reconstruction from a Single Moving Camera*, **Newcombe and Davison** (2010).

By exploiting the video rate data available from a single moving camera we demonstrated that the dense surface models can be computed more efficiently. We also replaced the feature based tracking pipeline with a direct tracking approach, utilising the dense geometric and photometric predictions made possible by the dense surface model: *DTAM: Dense Tracking and Mapping in Real-time*, **Newcombe, Lovegrove, and Davison** (2011c).

We then developed the full dense SLAM methodology that is central to this thesis. Taking advantage of newly available commodity depth cameras to reduce the complexity in the system development we focussed on enabling a truly incremental surface reconstruction.

The dense SLAM pipeline enables continually updating surface fusion, using all depth measurements in a live sensor stream and uses the current up to date surface model for dense tracking: *KinectFusion: Real-Time Dense Surface Mapping and Tracking*, **Newcombe, Izadi, Hilliges, Molyneaux, Kim, Davison, Kohli, Shotton, Hodges, and Fitzgibbon** (2011b). Further applications of the real-time capabilities enabled by the dense surface representation and real-time tracking were also investigated in *KinectFusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera*, **Izadi, Kim, Hilliges, Molyneaux, Newcombe, Kohli, Shotton, Hodges, Freeman, Davison, and Fitzgibbon** (2011).

In this thesis, we fully develop the form of alternating joint optimisation over the dense surface model and camera pose developed in KinectFusion, enabling real-time dense surface mapping and tracking with a single moving video camera, demonstrating the ability to achieve high quality dense visual SLAM.

1.6 Thesis outline

At the beginning of each chapter in this thesis we review relevant research areas specific to topics covered therein. In Chapter (2) we therefore take the opportunity to provide a background overview of related and interconnected areas from online visual SLAM through to multiple-view stereo looking in particular at techniques which were developed for use in a live operational setting.

In Chapter (3) we provide an introduction to the technical methods and notation used in the thesis. We describe the calibration procedures used to geometrically rectify and photometrically normalise the image data used in the visual SLAM systems, and provide an overview of the optimisation techniques that are at the core of the methods developed in later chapters.

In Chapters (4) and (5) we develop efficient convex optimisation based multiple-view depth map estimation algorithms. We investigate techniques that enable use of many small baseline images that are available from a video based data term, estimating depth maps as the solution to an energy minimisation problem comprising the multi-view stereo data term and a regularisation term. We select the most suitable algorithms that achieve a trade-off between accuracy and computation for use in the dense SLAM pipeline.

In Chapter (6) we describe the volumetric implicit surface representation that enables continuous integration of depth measurements into a consistent surface reconstruction. We also extend the model to capture a photometric representation of the scene and describe and develop the tools to enable efficient rendering of both the geometric and photometric predictions of the surface.

In Chapter (7) we combine the multiple view depth map estimation methods from Chapters (4) and (5) with the volumetric surface representation described in (6) and detail a pipeline for incremental surface reconstruction from video exploiting the full predictive capabilities of the dense surface model to reduce the complexity of estimating the depth map.

In Chapter (8) we develop the direct optimisation methods for estimating the pose of the live sensor using dense frame to model alignment. We describe the whole image alignment methods for both single passive and depth cameras that make full use of the dense surface predictions possible from the surface reconstruction pipeline.

In Chapter (9) we combine the dense tracking and mapping components detailed in the previous chapters into a number of dense visual SLAM systems. Importantly, we provide a video appendix at the end of the chapter which demonstrates each of the main systems and the components of which they are comprised.

Finally, in Chapter (10), we provide a summary of the work and the contributions made together with future research directions of the work.

2

Background

Contents

2.1 Feature Based Visual SLAM

We are interested in systems that are capable of providing a camera trajectory and scene maps without access to prior modelled artefacts or fiducial markers. This section provides a short background development of the current state of the art in real-time visual SLAM systems that succeed in producing accurate large scale maps in real-time with the associated camera trajectory. In all cases the visual SLAM systems described here produce models comprising simple geometric abstractions. Typically the models produced result in a sparse point cloud consisting of $100's$ or $1000's$ of 3D points, while more advanced systems extend the point cloud representation to include line segments.

A rich history of online or recursive estimation of the pose estimation and structure begins with the structure from motion community from computer vision. The study of obtaining

3D structure from 2D image motion has been extensively studied in the offline setting starting in the 1950s in the field of photography. Here we will only touch on research that begin to culminate in real-time, fully automatic systems that utilise the same basic principles found in the later visual SLAM systems that we will discuss in more detail below. **Jebara et al.** (1999); **Faugeras and Toscani** (1986); **Faugeras** (1993) provide a thorough overview of the subject detailing recursive structure estimation for causal image sequences. They note problems with small baselines having relatively little structure from motion information available but that correspondences are however easier to obtain between shorter baseline frames. The visual SLAM systems combined the research from structure from motion with online SLAM solutions such as Bayesian Filtering that provided a means of scaling up the optimisations while coping more explicitly with the uncertainties and ambiguities present in visual sensing, **Durrant-Whyte and Bailey** (2006); **Thrun et al.** (2005).

2.1.1 Sequential Filter Based Structure from Motion

Harris and Pike (1987), introduced a single camera tracking and mapping system: DROID, capable of real-time operation on very modest hardware. The environment is modelled as a 3D point cloud. Each 3D point has an associated covariance matrix to capture its uncertainty and can be seen as a forerunner of later visual odometry systems that accurately estimate the sequential motion of camera from real-time video by exploiting the uncertainty represented over the 3D point model to enable a constrained search for correspondences in the live frame.

Broida et al. (1990) detailed a filtering approach to 6DoF camera tracking using a single camera video stream. They initialise the system state comprising a map of 3D points and the camera pose using a batch bundle adjustment step. They also introduce the idea of reducing the search region for new features using the estimated covariance over the state space, however in their system they use manually defined fiducials which they then track by hand. **Soatto** (1997); **Chiuso et al.** (2000) emphasised the useful temporal constraints available in causal video data. Working independently of the SLAM community, they present an early Extended Kalman (EKF) Filter based system capable of joint structure and motion estimation using a single moving passive camera. **Jin et al.** (2000) demonstrated the principles of feature initialisation and detection that are fundamental to all filter based feature based visual SLAM systems.

Davison (2003), developed the *MonoSLAM* system, establishing a crucial footing for real-time single camera visual SLAM, building on earlier robot-based visual SLAM, (**Davison and Murray**, 1998). Using a joint state representing pose and a point cloud map within an EKF scheme, they use the joint uncertainty over predicted feature positions to reduce the computational cost of obtaining correspondences, the system's stability and agility surpassed previous systems, producing globally consistent maps over larger areas, leading to longer operation times than had previously been demonstrated. Early versions of the system needed a metric target to bootstrap the system from known features, and used a particle filter to coalesce new feature observations into an initial map point with Gaussian uncertainty. In a later version of the system by **Montiel et al.** (2006) the need for this heuristic mechanism was replaced with an inverse depth parametrisation of the initial map points that enabled representation of infinite uncertainty along the corresponding pixel ray, enabling for the first time fully automatic single camera monocular SLAM without a prior map.

Eade and Drummond (2006b) presented a novel approach based on camera tracking using filtering within local sub map nodes which are connected in a global map represented by a pose graph. Each node consists of an associated local frame of 3D points obtained through a Rao-Blackwelised particle filtering over the live camera state and tracked image features. A new node is introduced and connected to previous nodes in the graph via edges representing the relative pose between the local frames of reference. They perform pose optimisation over the graph propagating updates into the local nodes. **Eade and Drummond** (2008) further unify the loop closure and re-localisation components of the systems. **Pupilli and Calway** (2006) also used a particle filter on the state of camera pose demonstrating increased robustness to agile camera motion in comparison to the previous EKF based tracking systems.

2.1.2 Keyframe based Bundle Adjustment

Nistér et al. (2004) developed a real-time *visual odometry* system for stereo or single passive cameras using closed form solutions for camera pose estimation between frames and triangulation of 3D points from 2D image trajectories. They do not attempt to solve re-observation of historical map points leading to inevitable system drift. However by periodically restarting the system while keeping an initial pose estimate from the previous frame, they demonstrate a reduction in error build-up. **Engels et al.** (2006) showed that a carefully implemented bundle adjustment could run at real-time rates over a sliding window of recent video frames leading to increases in both camera trajectory and map accuracy, and importantly reducing error build-up that can lead to tracking failure when using more heuristic approaches.

Mouragnon et al. (2006) present a real-time single camera system using a local bundle adjustment approach with sparse visual feature tracking, capable of mapping $1000s$ of $3D$ points. Although no global optimisation is performed the system is robust on real-world urban sequences lasting around three minutes accruing a drift error of approximately 0.29 meters in comparison to a global bundle adjustment.

Klein and Murray (2007) developed *Parallel Tracking and Mapping*, a vastly different approach to the filter based monocular SLAM systems that came before it.

In place of an EKF where a joint representation of uncertainty over the map and pose restricted the achievable density of the map due to the $\mathcal{O}(N^2)$ complexity of the filter update step, Klein and Murray combine two novel ideas in their system. First they defined a separation between the task of tracking the camera given a known map, and updating the current map with new features. It is interesting to note a key insight from PTAM, that map building need not take place at frame-rate, is a product of the application area in which PTAM was built to succeed: Augmented Reality with a user in the loop, where browsing a scene is unlikely to lead to catastrophic system failure if part of the environment is not mapped. This can be contrasted with applications in the robotics community aiming for fully autonomous navigation in an unknown environment where an up-to-date representation of uncertainty over the pose and map of the scene are often a needed in online planning and control to ensure the robot does not fall into a state of physical dilemma.

In practice by dropping all explicit representations of uncertainty in the pose and map and decoupling the the tracking and mapping stages into a real-time tracking and slower offline bundle adjustment based map building component, an unprecedented level of performance over the filter based approaches was achieved in terms of tracking agility and accuracy as well as map scalability and density.

A second related innovation replaces the explicit extraction of descriptors to be associated with a map point with *keyframes* which are a selection of sparse source images decomposed into a scale-space pyramid with an estimated pose. A map point then holds a reference to a single source keyframe, together with the image location and scale where it was detected as a feature. This keyframe description of feature appearance simplifies the process of adding new features since no extra processing or storage is required, and many features may be present in a single image. The use of keyframes is also central to enabling efficient joint optimisation of the map structure in the SLAM system since sparsification of the live frame rate image stream into the keyframes enables practical use of bundle adjustment.

Robust point-based camera tracking is performed by projecting and matching features from the current map into each live frame, establishing data-association between the current model and live image data. The current camera pose is obtained by an iterative non-linear optimisation of the pose variables only, minimising the re-projection error of the data-associated features.

A mechanism for adding new keyframes, and with it new map points, works intermittently taking the current frame and pose and creating a new source keyframe. Correspondences from other keyframe points that can be data-associated are added to a growing list of keyframe correspondences. In a second thread global map optimisation is performed slower than frame-rate e.g. 2Hz. The set of the keyframes, along with the list of correspondences are bundle adjusted, to obtain a global estimate of the pose of the keyframes and the 3D point cloud model. A later investigation by **Strasdat et al.** (2010) compared the computational cost with achieved map and trajectory accuracy. The study showed that a keyframe based bundle adjustment approach in which more features are used for tracking without joint uncertainty typically leads to increased accuracy and stability over systems utilising joint estimation with uncertainty over sparser maps using a filtering framework.

Klein and Murray (2008) turned their attention to increasing the agility and robustness of the real-time pose estimation component of PTAM. Motivated in particular by the poor performance of point-based tracking in comparison to systems that rely on a prior built model, they argued that small image patches, as used in **Klein and Murray** (2007) and the earlier systems by **Davison et al.** (2007), are unable to handle large pixel motion due to motion blur artefacts that occur during rapid motion, and add edglets that in principle are more resilient to motion blur. They also add an inter-frame pose estimation mechanism that does not rely on the map, but instead uses a *whole image alignment* approach. Using the combined mechanisms they demonstrated a vast improvement in tracking higher velocity and acceleration camera motion.

They also introduced a simplified two stage re-localisation mechanism using the same direct alignment mechanism used in the frame-frame motion estimation. The effectiveness of the first stage was demonstrated by **Reitmayr and Drummond** (2006) within a known model tracking scenario. When tracking is lost, a zero mean sum of squared differences between the current frame pixel values and all keyframes is performed at coarse sub-sampled level of the scale-space pyramid. In the second stage, given the keyframe with the smallest photometric error, the $SE2$ whole image alignment optimisation is performed with an extra variable optimisation over the image mean to account for global illumination variation between the reference and live images. **Klein and Murray** (2007) point out that while the mechanism is not capable of arbitrary view relocalisation, i.e. when the camera is substantially rotated relative to the nearest keyframes, the low processing requirements of the optimisation together with user feedback enables the user themselves to easily move the camera to a nearby previous location increasing the likelihood of re-localisation. Such a simple mechanism contrasts with the more substantial approach taken in the original PTAM and developed by **Williams et al.** (2007) based on computing a minimal set of correspondences between the live frame and a prior learnt set of map features resulting in 3D-2D point correspondences that enable the pose to be estimated within the RANSAC frame of **Fischler and Bolles** (1981).

2.2 Live Dense Reconstruction

In this section we review systems that produce dense maps by exploiting the maps and real-time camera trajectories obtained with feature-based visual SLAM system. As in the previous section we are most interested in only those systems that attempt to achieve a real-time or incremental result. Unlike offline dense reconstruction systems that assume all camera poses are known prior to the dense reconstruction step, a live dense reconstruction (LDR) system must cope with increased or unknown uncertainty in the camera pose estimates. Furthermore, in the live setting the data input to the system is not fixed, hence LDR like visual SLAM systems must provide a solution should ideally posses a constant computational cost per frame enabling ongoing incremental reconstruction.

2.2.1 Extended Features in Visual SLAM

A natural extension for the feature-based visual SLAM systems is to incorporate an extension to richer geometric modelling primitives beyond points, perhaps the simplest example is the use of line or low dimensional parametric curves, (**Smith et al.**, 2006; **Eade and Drummond**, 2006a; **Klein and Murray**, 2008). An example map produced by PTAM, (**Klein and Murray**, 2008) is shown in Figure (2.1a) using the joint point and edglet scene representation.

Molton et al. (2004) used planar patches though not explicitly to increase map density but to increase feature correspondence by enabling better prediction of the map points over larger baselines due to representation of surface orientation provided by the patch. **Chekhlov et al.** (2007) coalesce co-planar map points to initialise planes represented in an EKF based monocular SLAM system, although observations of the planes remain point-based.

2.2.2 Free-Space Carving Approaches using Sparse Features

An important property of the previously introduced visual SLAM systems is the ability to maintain globally consistent maps through association of historical features with new observations obtained by data-association. Furthermore, the visual SLAM systems either explicitly represent map uncertainty (**Davison et al.**, 2007) or enable an estimate over joint map and pose uncertainty using the partial derivatives of the bundle adjustment error function (**Davis**, 2006).

A number of researchers have demonstrated dense reconstruction based on a 3D triangulation of the sparse point clouds available from real-time visual SLAM systems. An elegant incremental reconstruction approach can be achieved by using estimated correspondence information associated with each 3D point to provide constraints on free-space in the scene. Assuming a noiseless map point, the space along a ray emanating from a camera center and ending at the point must be empty requiring any represented solid that tessellates the space, such as tetrahedra, to be carved away.

Hilton (2005) introduced a provable theory for the reconstruction of dense geometric models consistent with all induced visibility constraints produced by a point map. They develop an efficient recursive algorithm which is proven to have a constant computational cost per new frame, but critically relies on point set noise distribution devoid of outliers, with only a small amount of measurement noise being tolerated for correct reconstruction.

Lovi et al. (2010) used PTAM (**Klein and Murray**, 2007) as the basis of an incremental free-space carving approach for fast rough estimation of scene geometry. The system includes the ability to handle map point insertion, deletion and point refinement whilst maintaining a run time cost proportional to the number of points visible in any key-frame. However, the visibility constraint does not take into account measurement uncertainty leading to possible incorrect carving of large structures, and a generally noisy surface reconstruction. A similar tetrahedral space carving approach using free space constraints induced by the point cloud map and selected camera frames in MonoSLAM is discussed by **Lovegrove** (2011).

Pan et al. (2009) developed *ProForma*, a probabilistic feature-based model system that generates good quality planar faceted models in real-time with modest commodity computing requirements. ProForma also use a keyframe based real-time structure from a motion based system to obtain a point cloud in real-time similar to PTAM. Unlike the incremental approach of **Lovi et al.** (2010), upon keyframe addition, ProForma reconstructs a full Delaunay tetrahedralisation of the updated point cloud and then uses an efficient probabilistic space carving algorithm to obtain a reconstruction consistent with all available visibility constraints.

While the space carving algorithms produce models that are consistent with the free-space constraints induced by a given point based map, the resulting models are typically very coarse and rough, seemingly as a consequence of the density of points in the constructed maps produced by the feature-based visual SLAM systems.

2.2.3 Real-time Dense Reconstruction on Commodity Hardware

If we can assume that the estimated poses obtained from the feature-based visual SLAM systems are within some bounded acceptable error we can separate the problem of estimating the camera pose, achieved using the feature-based approach, from that of reconstructing the observed scene with dense surface representation. Estimation of dense surface structure given multiple calibrated camera imagery has been extensively researched in the computer vision field of multiple-view stereo, (**Szeliski and Scharstein**, 2004; **Seitz et al.**, 2006), which has produced a large number of techniques to achieve high quality reconstruction. In this subsection we look at systems that achieve live dense reconstruction by separating the camera pose estimation and dense reconstruction tasks together with important developments that enabled the computationally demanding dense reconstruction components to operate on commodity hardware in a live or real-time setting.

Pollefeys et al. (1999, 2004) demonstrated one of the earliest, complete, single camera dense reconstruction pipelines. Although their system was not capable of real-time processing at the time of publication it contained the core of what has become one of the most successful dense reconstruction pipelines for both real-time and offline reconstruction applications. Their pipeline consists of first estimating the pose of a sequence of camera frames using SfM with a bundle adjustment refinement. Dense multi-baseline stereo is then computed on rectified temporally neighbouring image pairs, followed by multiple view linking of the dense correspondences to increase depth accuracy and reject low quality correspondences. A dense geometric model is computed by fusing the multiple view depth maps within the volumetric signed distance function fusion framework of **Curless and Levoy** (1996) followed by extraction of the surface mesh from the zero level set using the marching cubes algorithm by **Lorensen and Cline** (1987). Finally the mesh is simplified to facilitate efficient rendering of the reconstructed model within an augmented and mixed reality application. The resulting pipeline required several minutes for reconstruction from a sequence of five images obtained from a hand-held camera but established the result of acquiring a dense surface model from passive imagery using commodity computing hardware.

Pollefeys et al. (2008) addressed a number of computational issues associated with the pipeline outlined above, producing the first real-time capable dense reconstruction system using a single passive camera. Their work aims at a practical solution for reconstructing street size urban scenes viewed at car level. While the full incarnation of the system uses GPS and inertial measurement to obtain a real-time trajectory for up to four (non overlapping) cameras, the pipeline can utilise the structure from motion with bundle adjustment framework based on the systems of **Nistér et al.** (2006) and **Engels et al.** (2006) to provide camera pose estimates from a single camera video stream. Specifically, given the sparse visual SLAM results of systems such as PTAM from **Klein and Murray** (2007) or sliding window bundle adjustment by **Engels et al.** (2006), which provide real-time camera pose and sparse point-cloud estimation, the main difficulties overcome by **Pollefeys et al.** (2008) concern computing and fusing dense multiple view correspondences into a consistent surface reconstruction in particular by making use of newly available passively parallel commodity general purpose graphics processing (GPGPU) hardware.

Depth Map Estimation from a Single Camera

Collins (1996) introduced the elegant *plane-sweep* algorithm to obtain correspondences across multiple calibrated views as the minimum of a quantised disparity-space cost function induced in a chosen reference frame, resulting in a depth map for that frame. The plane-sweep approach directly enforces the epipolar geometry between the reference and comparison views equivalent to the direct search approach discussed in the introduction Equation (1.9).

Yang et al. (2003) introduced a real-time implementation demonstrating the effectiveness of GPGPU from several pre-calibrated static cameras for use in real-time teleconferencing applications. Previously such real-time capabilities had been available only through the use of specialised processing hardware, typically for fixed stereo pairs. The planesweep algorithm maps well to GPU hardware due to the trivial parallelisability of the stereo cost function computation removing the burden of depth map estimation from the host CPU, freeing up resources for other tasks in the real-time dense reconstruction pipeline. Many further developments increased the quality of the depth map estimation from multiple views, while more efficiently utilising the available commodity computing hardware.

Further rapid developments provided increases in depth map estimation quality while further utilising the growing capabilities of commodity GPU hardware. These included reduced noise in the estimated depth maps by spatial aggregation over the data term and occlusion handling, (**Woetzel and Koch**, 2004); the addition of spatial regularisation, (**Cornelis and Van Gool**, 2005); as well as addition of gain-adaptive data terms producing robustness to illumination changes across frames that break the brightness constancy assumption in the basic stereo data term, (**Kim et al.**, 2007).

Gallup et al. (2007) introduced a planesweep with multiple sweeping directions to address the issues associated with errors induced in the fronto-parallel plane sweep framework in which the error function computed for slanted surfaces visible across multiple views, being warped incorrectly, leads to erroneous local minima and decreased depth map accuracy. **Gallup et al.** (2007) also include an explicit occlusion handling mechanism (**Kang et al.**, 2001) in which the minimum of two temporally separated subsets either side of the plane-sweep reference frame is used, replacing the sum over all comparison views and leading to a decrease in the degradation of the disparity space cost function at depth discontinuities. While depth error increases quadratically in surface depth for the above systems **Gallup et al.** (2008) later introduced *variable baseline/resolution* stereo to achieve a constant depth error by dynamically altering the depth quantisation and camera baseline to keep a constant triangulation angle for any estimated depth.

Surface Reconstruction

The second major computational difficulty for the real-time dense reconstruction pipeline is the generation of a consistent surface reconstruction given the multiple view depth maps. The urban reconstruction approach of **Pollefeys et al.** (2008) addresses the issue by replacing the computationally expensive volumetric fusion approach of **Curless and Levoy** (1996) used in (**Pollefeys et al.**, 2004) with an explicit mesh representation of the scene constructed by compositing together multiple depth maps into a global frame using the visibility based depth map fusion approach by **Merrell et al.** (2007).

The depth map fusion approach from **Merrell et al.** (2007) generates small base-line depth maps, computed at frame rate on the input video using a GPU accelerated planesweep. These are back projected into the global frame where the depth map is triangulated. Meshes are fused in real-time using an efficient quad-tree structure in the image space by projecting neighbouring meshes into each others reference frames and updating the meshes to obtain a surface that reduces view consistency violations.

Zach et al. (2006) produced one of the first multiple-view stereo pipelines capable of dense reconstruction at interactive rates. While the resulting system did not demonstrate a complete live pipeline at the time of publication, the components present in the system and the emphasis on integration of all data available from a live video source makes the system the starting point for work in this thesis. By using components that efficiently exploit the massive computational resource presented by the GPU their aim was to produce a system with a constant computational cost associated with a new frame. This was in contrast the majority of offline multiple view stereo algorithms where performance scaled worse than linearly in the number of input frames (**Seitz et al.**, 2006). The system consists of computing a plane sweep stereo in a sliding window of frames using the gpu approach introduced by **Woetzel and Koch** (2004) but extended to include the more robust zero mean normalised cross correlation cost function. Their system integrates the short baseline depth maps into a dense volumetric reconstruction using a variant of the robust signed distance function averaging by (**Curless and Levoy**, 1996) again making efficient use of the GPU resource resulting in an order of magnitude reduction in integration time compared to a CPU implementation. A final mesh model extracted from the implicit surface zero crossing using a

CPU implementation of the marching cubes algorithms, but an interactive visualisation of the current reconstruction result is available using volume rendering techniques working directly on the GPU bound reconstruction. They demonstrate the full pipeline on a number of offline calibrated image sequences.

Vogiatzis and Hernández (2011) describe a video based multiple view stereo system using a per-pixel probabilistic depth estimation in which a posterior depth distribution is updated on every new frame, which unlike the majority of offline multiple view stereo system utilises hundreds of measurements possible from a video stream. The approach is similar to the extended Kalman filter based iconic depth map approach of **Matthies et al.** (1989), but crucially utilises a novel mixture model over the depth estimate increasing robustness to outliers and perpetual aliasing on repeating texture, similarly to the particle filter used in map point insertion mechanism in MonoSLAM (**Davison et al.**, 2007). Real-time camera pose estimates are obtained using a fiducial marker based tracking system, later replaced in **Woodford et al.** (2011) with live camera pose estimates obtained by PTAM. The result is a dense point cloud obtained in real-time for image regions with high enough texture to initialise a 2D image based feature tracker over a short baseline. When considering the possibility to regularise the correspondence field computed in the live image frame, **Vogiatzis and Hernández** (2011) argue that such early spatial regularisation, while leading to increased model completeness, reduces accuracy of the final model. Therefore for low texture regions, the reconstruction is sparse and an offline post-processing of the point cloud is required to generate the final surface reconstruction.

2.3 Global Optimisation and Regularised Stereo

Many problems in computer vision can be cast as global energy functions and solutions can be obtained by energy minimisation. This was previously demonstrated in Chapter (1) for a denoising problem, and the optimisation framework can be applied to estimate stereo directly without first extracting a depth map. The most widely researched class of energy minimisation approaches set up an energy summing two terms, defining the energy induced by a depth map solution D, the global energy is:

$$\mathsf{E}(D) = \mathsf{E}_{data}(D) + \mathsf{E}_{smooth}(D) \ , \tag{2.1}$$

where $\mathsf{E}_{data}(D)$ computes a cost (or energy) over a given data term for a possible solution. This is summed together with a regularisation $\mathsf{E}_{smooth}(D)$ term that penalises the non-smoothness of the solution in some way. A solution depth map is then obtained by searching for D that yields a minimum energy. Such regularisation of ill posed problems has been studied in many forms in computer vision in an attempt to understand how to state and solve low level vision problems in a useful, efficient and principled way. This work has also often been motivated by a desire to understand the principles behind the human visual system (**Marr**, 1982; **Poggio et al.**, 1985; **Blake and Zisserman**, 1987; **Szeliski**, 1991).

The particular form of the smoothness term is critical to obtaining high quality solutions. In general, because natural scenes are comprised of piecewise smooth surfaces it is important that the regularisation is discontinuity preserving. The seminal work by **Geman and Geman** (1984) originally proposed the Bayesian interpretation of many energy functions and formulated regularisation terms that are discontinuity-preserving.

We now briefly look at *discrete* optimisation approaches which have been extremely successful in providing solutions to computer vision problems in the form of Equation (2.1).

2.3.1 Discrete optimisation

The most widely researched techniques for minimising the global energy attempt to solve a discrete labelling problem where the solution for each pixel in a depth image is assigned a discrete label $D : \Omega \mapsto \{Q_0, Q_1, .., Q_K\}$. In the simplest case a label can correspond at each pixel to a discrete depth value. However, labels can instead specify pixel membership to some local parametric estimation of a region.

Combinatorial Optimisation: The difficulty in optimisation of global energies depends on the particular form of labelling and smoothness term used. For example, for a $1D$ label set of finite size such as for stereo labelling, if the smoothness term is restricted to a convex function of the solution space then an exact solution exists which can be found using the group of combinatorial optimisation techniques, known in computer vision as graph cuts (**Boykov et al.**, 2001). Use of a more general smoothness term unfortunately renders the problem NP-hard (**Veksler**, 1999)

Tappen and Freeman (2003) and **Szeliski et al.** (2008) provide an extensive comparison of state of the art discrete optimisation approaches including graph cuts and belief propagation **Sun et al.** (2003), with application in the stereo setting. Unfortunately, although Graph cuts achieves high quality (near global minimum) solutions efficiently in comparison to the previously non-deterministic search approaches such as simulated annealing (**Barnard**, 1989), the solutions even in the two view stereo setting are not real-time applicable, often taking on the order of minutes to solve even for space of $K = 256$ labels per pixel with a convex regularisation function. While developments in practical belief propagation have been achieved (**Felzenszwalb and Huttenlocher**, 2006), yielding simple and efficient parallel implementations, they too are outside of the range of real-time operation.

Convex Relaxations: More recently **Pock et al.** (2008a) introduced equivalent continuous optimisation based convex relaxation formulations for a subset of the multi-label problems which include stereo. A number of practical advantages are gained in the continuous setting over the discrete counterpart, including efficient parallel implementation leading to an order of magnitude speed-up using commodity GPU hardware over CPU based implementations, and also a reduction of the metrication errors that result from the approximation used for distance computation in a local neighbourhood used in the discrete MRF models which only approximate a discretisation of the gradient operators which measure smoothness and are easily implemented in the continuous formulation (**Pock**, 2008; **Klodt et al.**, 2008). Unfortunately the speed-up gained due to efficient parallel implementation is still far from useful in a real-time depth estimation setting.

Semi-Global Matching: In a rectified two view stereo setting, each scan-line can be treated as an independent $1D$ problem comprising a solution smoothness or consistency constraint along with the data term cost. Such scan line optimisation can be efficiently solved using dynamic programming. However streaking artefacts in the solution result from the lack of smoothness constraints existing between pixels on neighbouring scan lines.

Hirschmüller (2005) provides a solution to this in the form of semi-global matching (SGM) which introduced a very efficient alternative to the full global optimisation of discrete labelling problems. By splitting the global energy into several $1D$ optimisation problems, where the data and smoothness cost minimum is computed along several directions, he obtained a good approximation to the global minimum energy by selecting the minimum energy computed amongst all paths. Because each of the decoupled $1D$ optimisations can be solved using the dynamic programming approach SGM is computationally efficient. Evaluation comparing SGM to the combinatorial optimisation approaches show similar performance but with a speed-up of approximately $50\times$, making the approach potentially useful in a real-time application. Extension to the multiple view setting can be performed via any form of multiple input depth map denoising approaches where separate pairwise depth maps are computed and then combined into a higher quality estimate.

2.3.2 Continuous Optimisation

Within continuous optimisation based depth estimation, an energy functional is devised, mirroring Equation (2.1) consisting of a sum of two terms:

$$\min_{D} \left\{ \int_{\Omega} \sum_{k \in K} \psi_{\mathcal{D}} \left(\epsilon_k \left(x, D \right) \right) \, dx + \int_{\Omega} \lambda \psi_{\mathcal{R}} \left(\mathrm{A} \left(D \left(x \right) \right) \right) \, dx \right\} . \tag{2.2}$$

When the error function $\epsilon_k(x, D)$ is linear in D and the data and smoothness norms are chosen to be convex, then the functional is convex and a solution with a globally minimum energy can be obtained in practice. In particular, the importance of such a continuous convex energy functional lies in the assurance that a local minimum of the solution is also the global minimum, and can be achieved independent of the solution initialisation using a gradient descent style optimisation. This is extremely important in practice, since we can then exploit the massive compute power of modern GPGPU to obtain real-time performance, which has been found to be much harder for the previously described discrete combinatorial optimisation approaches.

Linearised Data Terms: Unfortunately multi-view stereo data terms for ϵ in Equation (2.2) are non-convex, for example for two-view stereo given a reference image $\mathcal{I}_r$ into which a depth map will be estimated and a second view $\mathcal{I}_l$, the error function:

$$\epsilon(x, D) = \mathcal{I}_r(x) - \mathcal{I}_l(\mathbf{w}(x, D)) \, , \tag{2.3}$$

where $\mathbf{w}$ is the warp function from Equation (1.8), is generally non-convex in D. The general solution used by nearly all continuous optimisation approaches is to linearise the data term around a given solution estimate:

$$\rho(x, D) = \mathcal{I}_r(x) - \mathcal{I}_l(\mathbf{w}(x, D_0)) - (D - D_0)\nabla_D I_l(\mathbf{w}(x, D_0)) \tag{2.4}$$

Horn and Schuncks Optical Flow: Multiple-view variational depth estimation can be seen as a constrained $1D$ form of the optic flow problem where the data term can be extended over multiple images. In the seminal optical flow paper **Horn and Schunck** (1981) introduced the vision community at large to the variational formulation of two view optical flow. In doing so they demonstrated that superior performance in obtaining dense correspondences could be achieved by solving a global optimisation problem involving a local smoothness assumption and simple point-wise minimisation of the data term. Their functional comprised a linearised brightness constancy data term, together with a quadratic penalisation of the solutions 1^{st} order gradient.

Course to Fine Solution: To ensure that linearisation of the data term holds, gradient descent based continuous optimisation methods often embed the solution in a *coarse-to-fine* framework. Initially, each of the input images are sub-sampled into separate image pyramids. Coarse to fine optimisation then proceeds to solve the lower resolution solution by minimising the global energy functional on the coarsest scale of images. The result of the coarse scale estimate is then up-sampled (and scaled appropriately) to the next highest resolution resulting in a linearisation point for the subsequent set of gradient descent iterations. Up-sampling followed by gradient descent continues until the original image scale is reached.

Variational Depth Map Estimation

Early variational formulations of the depth estimation mirrored earlier approaches in the optical flow research community performing gradient descent on the Euler-Lagrange equations of the energy functional. A number of innovations were introduced using robust data terms in combination with discontinuity preserving regularisation. An equivalent two view stereo version to Horn and Shuncks optical flow approach combining quadratic penalisation of the linearised brightness constancy data term together with quadratic penalisation of the first solution variable first order gradients,

$$\min_{D} \left\{ \int_{\Omega} \rho(x, D)^2 \, dx + \int_{\Omega} \lambda |\nabla D|^2 \, dx \right\} . \tag{2.5}$$

which is trivially extended to multiple views by replacing the two view linearised data term with its SSSD linearised counterpart.

Robert and Deriche (1996) used the Aubert function $\sqrt{1 + (s/k)^2} - 1$ over the basic quadratic penalisation of $s = \nabla D$ which provides a depth discontinuity preserving smoothness term. They also demonstrated the power of the image driven anisotropic regularisation (IDAR) previously introduced by (**Nagel and Enkelmann**, 1986) in optic flow community and originating in image restoration research. Image driven regularisation enables the strength of a solution smoothness prior to be modulated by another image source on a per solution point basis. Using an edge operator computed over the reference intensity, a weight can be derived to reduce the regularisation power at image boundaries. Since depth discontinuities often align with image boundaries a better depth discontinuity preserving regularisation is obtained. **Robert and Deriche** (1996) also show that the quality of reconstruction obtained using three images over two, where the comparison views captured are captured from camera translation above or below *and* beside the reference frame reduces the data term ambiguity that exists when depth boundaries align with the image axis.

Faugeras and Keriven (1998) provide novel variational formulations for multiple-view stereo reconstruction. Their solution parametrisation was over a complete surface manifold represented within the level-set frame-work and so is not strictly in keeping with the depth map estimation formulations we are detailing here, they utilised an NCC patch data term which can be expressed in a very similar form for any other non-parametric depth representation. They also detail the importance of computing the data term gradient at the linearisation point taking into account the surface normal there to reduce the fronto-parallel bias introduce by the patch data term.

Álvarez et al. (2000) also make use of the linearised brightness constancy data term and IDAR regularisation using a quadratic cost on ∇D, and solve the resulting PDE from the Euler-Lagrange equations within the linear scale space framework (**Weickert et al.**, 1999) resulting in improved correspondences over larger baselines.

Strecha et al. (2003) introduced a number of innovations focussing on wider baseline multiple-view depth estimation using colour images. Using an inverse depth parametrisation of the scene, they formulate a novel energy functional comprising a linearised brightness constancy data term with a per pixel gain under quadratic penalisation. Each pixel within a view also has an associated correspondence variable used to down weight erroneous pixel values. While estimation of the data association is heuristic, the per pixel gain adaptation is globally optimised with a quadratic penalisation of the fields first derivative. They also utilise IDAR under quadratic penalisation of ∇D. After discretisation of the Euler-Lagrange equations they iteratively solve the PDE using a novel inhomogeneous time diffusion process enabling a per solution point time step leading to faster convergence near image regions where prior sparse feature matching used in the camera calibration step are available.

Kim and Sohn (2003) present a rectified two view variational approach using combining the quadratic penalisation of the linearised brightness constancy with an *inhomogeneous isotropic image driven regularisation* introduced by **Geman and McClure** (1985) $g(s^2) = 1/(1+s^2)^2$ instead of the IDAR, although still within a quadratic penalisation of ∇D. Rather than solving the resulting PDE using a course to fine gradient descent they initialise the solution on the full resolution images using a per pixel data term minimum solution obtained using block matching.

In their paper *"Optic flow goes stereo"*, **Slesareva et al.** (2005) use the epipolar constraints available between two views and reformulate the state of the art of optical flow technique from **Brox et al.** (2004) in the constrained $1D$ setting. They introduce the ℓ_1 penalisation based on a combined data term using both a linearised brightness constancy and linearisation of the image gradients to increase robustness to illumination change. Similar to the discontinuity preserving smoothness used by **Robert and Deriche** (1996), they introduce the total-variation regularisation using the ℓ_1 in place of quadratic penalisation of the ∇D which had been used to great success in image restoration **Rudin et al.** (1992). They remove the discontinuity present in the ℓ_1 costs using ϵ regularisation of the norm, and solve the resulting fully convex functional in the coarse to fine framework used in the equivalent optic flow formulation. Since the PDE is non-linear due to the denominator of the ℓ_1 norm derivative, they perform a nested fixed point iteration by lagging the denominator to the previous iteration resulting in a sparse linear system which can be solved using successive over relaxation (SOR). **Liu et al.** (2009) make use of the same formulation for their multi-view stereo continuous depth estimation framework and point out the advantages of the single pixel data terms for obtaining reconstruction of fine details.

Ben-Ari and Sochen (2007) focused on extending the sophisticated variational framework developed by **Shah** (1993) that explicitly models solution discontinuities and occlusions using the piecewise smooth segmentation approach developed by **Mumford and Shah** (1989) (MS). Regularisation within the MS framework takes the form of a cost based on both the solution smoothness such as quadratic variation $|\nabla D|^2$ in combination with a measure of boundary length of disparate regions within the solution. They make use of a robust ℓ_1 joint colour and gradient constancy data term that is defined over the set of non occluded pixels determined by the segmentation result. They also use the total-variation regularisation in place of the quadratic variation used in **Shah** (1993) to improve further the discontinuity preservation. The coupled segmentation and disparity estimation is then solved using alternate minimisation of each functional within a coarse to fine framework.

Slesareva et al. (2007) develop a novel robust data term computing gradient constancy on logarithmically transformed input images. Global illumination changes are transformed into additive perturbations by the image logarithm, and since the gradient operation is invariant to such transforms the result is invariance to such changes. They use an $\epsilon - \textit{regularised}$ ℓ_1 penalisation over the data term together with the quadratically penalised IDAR regularisation and solve the non-linear PDE obtained from the Euler-Lagrange equations using the lagged denominator based linearisation approach.

Zimmer et al. (2008) derive an *anisotropic solution driven* two view stereo formulation bridging the gap between the isotropic solution driven approaches (i.e. Total Variation) and the popular image driven anisotropic regularisation. They couple the regularisation with an ℓ_1 penalisation of brightness and gradient constancy data terms and solve using a multi-grid gradient descent on the resulting non-linear PDE.

Kosov et al. (2009) introduce a novel multi-level adaptive technique (MLAT) that enables efficient approximate solution of the Euler-Lagrange equations by introducing a measure of stability for both the data term and smoothness term energy in a given iteration. Starting at a coarse scale grid resolution they refine the computational grid over which the solution is defined only in solution regions which have not converged, thereby reducing the total number of iterations required for regions with strong data terms or that adhere quickly to the local smoothness measure. Using a linearised brightness constancy data term they also make use of an adaptive regularisation term which is switched between either a Charbonnier or Perona-Malik regularisation, both of which are non-convex functions and lead to discontinuity preservation. Switching is achieved on a per pixel basis by learning a mapping between the best regularisation to use and a current value $\|\nabla D\|^2$ estimated in an offline manner on stereo data sets.

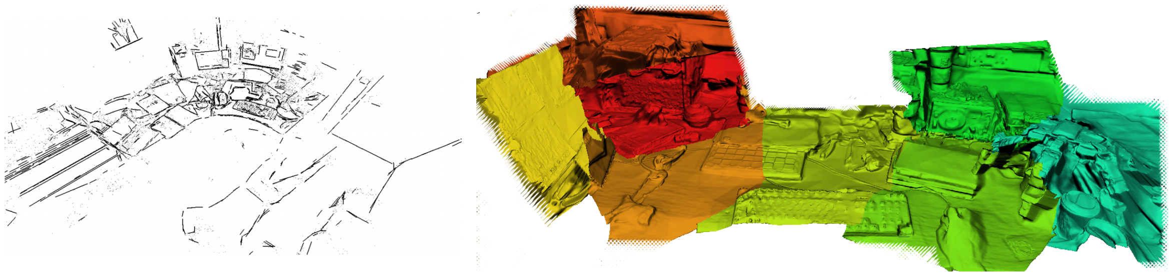
2.3.3 Fast Convex Optimisation for Dense Reconstruction

A great amount of research effort has been applied to efficiently obtain depth maps of higher quality than can be obtained by searching for the stereo data term minimum alone. **Scharstein and Szeliski** (2001) produced a taxonomy of the main algorithms available early on in the research specifically looking at the two view stereo case which often are also applicable in the multiple view setting. Research has continued to grow and produce improvements on benchmark experiments for accuracy and robustness, but an important distinction for application within a live dense reconstruction setting is the computational efficiency of the algorithms. Many of the state of the art results which can not trivially make use of GPGPU hardware typically require minutes or hours to compute a depth map from a single stereo pair.

Pock et al. (2007a); **Pock** (2008) established a paradigm shift in real-time computer vision with the application of continuous convex optimisation techniques efficiently implemented on commodity GPUs to achieve state of the art real-time image denoising. In a series of papers, a number of top performing algorithms were introduced in image denoising (**Pock et al.**, 2007a), optical flow (**Zach et al.**, 2007a), segmentation **Zach et al.** (2008); **Unger et al.** (2008), and dense reconstruction **Zach et al.** (2007b), providing real-time results on commodity hardware using principles from continuous convex optimisation.

The rigorous convex optimisation framework used provides globally optimal results by posing the solution to each problem as the result obtained by minimising a convex energy functional comprising a data term and some spatial smoothness term. Within this optimisation setting the minimisation is obtainable using a first order gradient descent style algorithm on the discretised functional (or by discretising the resulting gradient descent equations). While more sophisticated optimisation techniques exist to minimise the energy functional, these gradient descent equations can be trivially parallelised leading to an efficient mapping on commodity GPU architectures enabling rapid computation of the solution.

Zach et al. (2007b) developed further the depth map fusion pipeline (**Zach et al.**, 2006), producing a globally optimal range fusion approach to dense reconstruction. The technique transforms input depth maps into the truncated signed function form which are treated as 3D images. They produce a denoised TSDF as a minimum of a convex combination of the data term error computed under an $L1$ norm from all input TSDF, under total-variation



(a) Point and edge map from PTAM (Klein and Murray, 2008)

(b) Mesh surface model LDR, (Newcombe and Davison, 2010)

Figure 2.1: PTAM point and edglet map of an office scene, a person is lying on the floor. Live Dense Reconstruction from a Single Moving Camera, using a hybrid approach applied to a desktop scene. While no edglets or points are consistently mapped using the sparse visual SLAM approaches, shape information is readily available from short baseline multiple view stereo.

regularisation of the solution. **Zach** (2008) introduced a novel representation for errors in the data-term, resulting in constant time optimisation with the total variation regulariser of the TSDF independent of the number of input depth maps. The algorithm was implemented efficiently on commodity GPU hardware and evaluated on offline pre-calibrated image data achieving state-of-the-art multiple view stereo performance in quality but with vastly decreased computation times as seen in the middleburry MVS evaluation website, (**Seitz et al.**, 2006).

Graber et al. (2011) used PTAM to produce a live running version of this efficient depth map fusion approach. For each new keyframe produced by PTAM, a spatially local set of neighbouring keyframe images with associated bundle adjusted poses are used in a concurrently running depth map estimation thread using a GPU implemented planesweep. The system interleaves global optimisation with addition of new depth maps into the data term when they become available. The current implicit surface is visualised using ray-casting, and texture mapped using a set of nearby keyframes through projective texturing. Since depth maps are only produced at keyframes, the system does not use all data available in the live video stream, typically requiring tens of seconds before a sufficient set of initial keyframes are available for a single depth map to be generated.

Newcombe and Davison (2010) developed a *live dense reconstruction* pipeline built on top of PTAM described in more detail in Chapter (9). A surface patchwork of overlapping depth maps is computed at automatically selected reference frames covering the currently observed scene. The depth maps are estimated from multiple spatially local views around each reference frame using a dense correspondence field computed between frames using a dense variational optical flow, (**Zach et al.**, 2007a) computed on the GPU in coarse-to-fine manner. To enable correspondences across wider views, increasing the quality of resulting triangulation into a depth map, the optical flow estimation is initialised using a predicted correspondence field induced between views by a coarse geometry proxy obtained by fitting a surface to the PTAM point cloud. Figure (2.1) shows an example desktop scene reconstruction.

Stuehmer et al. (2010) demonstrated a real-time GPU implemented depth map estimation also within the convex optimisation framework of **Zach et al.** (2007a), but further exploit the epipolar constraint between the multiple views given the static scene assumption to remove the unnecessary burden of the pairwise optic flow computation used in **Newcombe and Davison** (2010) with a direct parametrisation on depth. We will return to modern convex optimisation based multi-view stereo in Chapter (5).

2.4 Dense Tracking and Mapping

The live dense reconstruction pipelines of the previous section enhance the 3D point maps of sparse visual SLAM systems with a dense surface reconstruction. However, while the surfaces are useful in a variety of applications, they are not used within the central scene representation used to solve the SLAM problem. In this section we look at systems that go beyond acquiring the model as the end result to pipelines that use live *dense* model reconstruction within the SLAM pipeline.

Specifically given a dense model representing not only the scene geometry but also surface appearance, it is possible to predict a whole image view simply by rendering the model in a given camera frame. This is in contrast to the predictive capabilities of the sparse visual SLAM systems that predict only the location in the image projections of geometric features.

We outlined in the introduction Section (1.4), the potential of direct tracking when a generative model of image appearance is available. These techniques originally developed for non-projective transformations of the image pixels, (**Lucas and Kanade**, 1981), and are often used as a sub-pixel refinement step in the feature based tracking and mapping systems such as PTAM and monoSLAM. **Baker et al.** (2004b) describe the basic extension of the forward compositional luckas-kanade approach to 6DoF pose estimation when a dense textured model is available. The improvement to the frame-frame tracking robustness of **Klein and Murray** (2008) comes by mitigating the need to obtain binary data association of sparse features, all of the following systems share this characteristic but go further by making use of technique as the central tracking methodology.

A dense tracker is capable of using all of the pixels in a live measurement to align the model prediction, having the potential to add massive redundancy into the 6DoF optimisation problem without having to decide beforehand which features are good to track. This results in more robust camera tracking. We will discuss the technique in detail in Chapter (8).

Silveira et al. (2008) introduced a *direct visual SLAM* approach which departs from the traditional sparse visual SLAM systems from the previous sections to unify the tracking and mapping of the scene with the extraction, tracking and matching of sparse features. Their key observation is that reduced drift in camera trajectory can be obtained if a map's features can be observed throughout wider baselines as the camera moves. Specifically, they parametrise the scene as a set of planes and define a joint optimisation over the plane parameters and camera pose parameters. The joint estimation is solved using a direct image-intensity based error minimisation over the plane induced homograph between a set of keyframes. They formulate the minimisation using an efficient second-order method (ESM) for optimisation over the homography parameters, (**Malis**, 2004) providing faster convergence in comparison to the Gauss-Newton approximation first-order accurate formulation. The sparse visual SLAM based systems that have typically only small regions of stable observability over the map features with a given descriptor, due to geometric appearance changes which are not modelled. In comparison, the large planar regions can be continuously tracked for longer periods of time over larger baselines leading to greatly reduced drift.

Lovegrove (2011) also uses the plane induced homography based ESM optimisation but parametrises the scene as a single plane where a photometric prediction is obtained using a set of keyframes historically placed as the camera is tracked in real time using a frame-frame whole image alignment. The system performs loop closure detection and obtains globally consistent maps through a pose-graph optimisation using the relative pose constraints estimated from keyframe-keyframe alignment.

Comport et al. (2007) details the quadrifocal tensor based visual odometry framework for a calibrated stereo camera. A dense depth map obtained from the stereo pair along with the reference frame image provides a dense model at keyframe locations. The dense prediction can be warped via a quadrifocal tensor into estimated live camera frame using a 6DoF camera transform between the keyframe and the estimated frame. The live stereo pair is therefore aligned by optimising the relative camera transform to minimise the prediction error between the predicted and live views using an efficient second order gradient descent. Their system demonstrates the advantages of using all of the image data available in a new frame and the model, to provide reduced drift without the traditional binary data association based feature tracking and matching visual SLAM systems while working on a non-parametric representation of the scene.

Newcombe et al. (2011c) developed *DTAM: Dense Tracking and Mapping in Real-time*, demonstrating the first *single camera* system to use both a dense non parametric scene mapping and dense tracking pipeline showing increased robustness to agile camera motion. The system represents the scene as a composite of overlapping dense textured meshes and consists of two core components: using the current dense model of the scene they use the 2.5D dense whole image alignment method of **Baker et al.** (2004b) to track camera motion at frame-rate. Interleaved, given a sliding window of images from the tracked camera, they update and expand the model by building and refining dense textured depth maps using a novel convex optimisation based multiple view stereo technique developed in Chapter (5). The DTAM system is boot strapped using the PTAM system to obtain the poses required for an initial texture depth map before running independently. They directly compare the tracking robustness to the PTAM system for desktop sized scenes demonstrating the ability to track through motion and focal blur while providing a dense surface prediction for use in geometry aware augmented reality. The system is described in further detail in Chapter (9).

Tykkala and Comport (2011) develop a dense visual SLAM method for a passive *stereo camera* to compute a 3D textured dense point cloud model of the scene. A dense tracking component estimates the live camera poses using a variant of the Lucas-Kanade 2.5D alignment approach, (**Comport et al.**, 2007). The model keeps a 1D Gaussian uncertainty associated with each model point represented in a reference key-frame. The point is then updated by exhaustively searching for a correspondence as the minima of the photometric cost function induced along the associated epipolar lines within neighbouring images. Importantly they use a bounded region within which to search using the associated point uncertainty to make the search tractable, and to reduce mismatches due to repeating texture or a reduced signal to noise ratio. New map points are initialised for image regions where there are gaps in the model using a standard planesweep technique. The system demonstrates high quality pose estimation with a low level of drift due to the massive redundancy obtained in tracking from a dense model.

2.5 The Advent of Commodity Depth Cameras

Commodity RGB + Depth sensors like the *Microsoft Kinect* and *Asus Xtion Pro* (both based on the *Primsense* structure light device) provide a real-time high resolution dense depth map alongside the traditional passive RGB video. The availability of such sensors has led to an explosion in practical SLAM being used in a variety robot systems for uses in navigation, object recognition, grasp planning and augmented reality applications. The commodity depth sensors provide the real-time depth map as output without the need of the large computational resource required for the real-time stereo approaches outlined in Section (2.2.3).

Prior to the available commodity depth cameras, **Rusinkiewicz et al.** (2002) pioneered the first demonstrably live dense 3D reconstruction system comprising a structured light based depth measurement built using a commodity projector and single passive camera. While the specifics of their hardware and application to small model acquisition reversed the roles of the moving camera and static scene, their system pipeline demonstrated the core for dense reconstruction using real-time depth images alone. As they move an object in front of the depth camera, they align new depth scans into a mesh surface model using a fast iterated closest point optimisation. The partial scans are visualised in a global frame using an efficient splat rendering (**Rusinkiewicz and Levoy**, 2000), enabling feedback to the user of where the model is currently incomplete. Finally they use the volumetric signed distance function integration approach by **Curless and Levoy** (1996) to fuse the depth scans into a consistent surface reconstruction.

Since the dense depth map mitigates a large computational cost in obtaining correspondences for initialisation of 3D maps, the sensors can trivially replace the single passive only camera used in all of the sparse visual SLAM systems discussed in section 2.1 leading to improvements in mapping density and subsequent improvements in tracking quality. Recently, **Strasdat et al.** (2011) utilised the depth camera to ease correspondence computation for sparse visual feature based tracking within a keyframe based monocular SLAM framework speeding up initial map point estimation. They enable a dense registered coloured point cloud visualisation by hanging the depth maps from optimised keyframe poses, but do not the use the depth map and available dense point cloud within pose estimation.

Henry et al. (2010) developed a full visual SLAM pipeline for medium to large indoor environments. They combine a feature-based visual SLAM pipeline enabling global consistency with a dense ICP based frame-frame tracking mechanism using the available depth maps to increase robustness of the pose estimation when the number of visual features available is small. They represent the scene as a series of dense key-frames for use in a loop closure and pose graph optimisation. They also perform an *offline* processing of the point cloud obtained from the keyframes into a higher quality surface reconstruction using a *surfel* representation (a surface element comprising a location, scale and 3D orientation), (**Pfister et al.**, 2000). While it would be desirable for the surfel representation to be computable online, they note that incremental updating of the representation is prohibitively expensive since points used in computing a given surfel are dependent on the keyframe poses which continue to change during global optimisation. The system demonstrates impressive performance on real-world datasets constructing maps within whole offices, but requires relatively slow camera motion to ensure the sparse visual SLAM based correspondence is achieved and tracking maintained.

More recently a number of researchers have taken advantage of the frame-rate and high quality RGB-D measurements from the commodity depth sensors. Returning to visual odometry systems that accurately estimate the camera pose using methods based on directly minimising the dense $2.5D$ frame-model alignment error, they demonstrate very low levels of drift due to massive redundancy in the optimisation problem which can be robustified to moving objects and illumination changes, (**Comport et al.**, 2011; **Audras et al.**, 2011; **Steinbrucker et al.**, 2011).

Newcombe et al. (2011b) developed the *KinectFusion* system, demonstrating high quality, real-time surface reconstruction from a single moving depth camera. KinectFusion interleaves dense surface estimation using a real-time implementation of the volumetric signed distance function integration method from **Curless and Levoy** (1996). The trivial paralleslisability of the weighted average update rule introduced in Section (1.4.3) is leveraged to perform the surface fusion on commodity GPGPU hardware. The up-to date surface reconstruction provides the implicit surface estimate as the current zero-level set of the SDF, extracted in KinectFusion using direct raycasting on the SDF volume. Real-time camera pose estimation is achieved by aligning a new depth using dense ICP with a prediction from the current surface reconstruction. Utilising all surface measurements in a tightly interleaved surface reconstruction and camera tracking pipeline leads to reduced drift in the system in comparison to frame-frame pose estimation. Furthermore, despite no explicit joint estimation of the camera pose with the surface reconstruction, KinectFusion is capable of consistent drift free dense reconstruction within workspaces ranging in size from desktops to small rooms. The incremental surface reconstruction components are detailed in Chapter (6) and the dense camera tracking framework is detailed in Chapter (8). Results of the complete system together with further extensions to enable larger scale mapping are discussed in Chapter (9).

3

Technical Introduction

Contents

In this Chapter we provide an overview of mathematical notation, camera calibration models and optimisation tools used throughout the thesis. We begin in Section Section (3.1) by giving definitions of the geometric notation we will use to describe the point transfer between frames of reference, parametrised with a Euclidean transform in $3D$ space. In Section (3.2) we outline the models used for both geometric and photometric calibration of a camera enabling simplifying assumptions in various techniques later developed. In Sections (3.3) and (3.4) we provide an introduction to the two main optimisation tools used in development of the dense visual SLAM tracking and mapping components in this thesis, and motivate the simplicity of their implementation on modern parallel hardware in Section (3.5).

3.1 Geometry

We define a rigid body transformation comprising a translation t_{ba} and rotation component R_{ba} as $T_{ba} \in SE_3$, where the special Euclidean group, SE_3 is:

$$SE_3 \triangleq \left\{ T_{ba} = \left(\begin{array}{c|c} R_{ba} & t_{ba} \\ \hline 0_{1\times 3} & 1 \end{array} \right) | R_{ba} \in SO_3, t_{ba} \in \mathbb{R}^3 \right\} \ , \tag{3.1}$$

and the rotation component R_{ba} is in the special orthogonal matrices, SO_3:

$$SO_3 \triangleq \left\{ R_{ba} \in \mathbb{R}^{3\times 3} | R_{ba}^{\top} R_{ba} = I, det(R_{ba}) = +1 \right\} \ . \tag{3.2}$$

We transform a point $x_a \in \mathbb{R}^3$ represented as a 3-element column vector from a frame of reference a into a second frame of reference b using the rigid body transform:

$$\dot{x}_b = T_{ba} \dot{x}_a \ . \tag{3.3}$$

Here, the *dot* notation defines the homogeneous point $\dot{x} \equiv \left[\begin{smallmatrix} x \\ 1 \end{smallmatrix} \right] \in \mathbb{R}^4$, enabling multiplication by the 4×4 transformation matrix. When multiplying by the transformation matrix, unless specifically stated otherwise, we will imply that dehomogenisation has been performed after multiplication of the homogenised vector, equivalent to explicit rotation followed by translation:

$$x_b = R_{ba} x_a + t_{ba} \ . \tag{3.4}$$

This notation allows us to chain transforms together, since:

$$T_{ca} = T_{cb} T_{ba} \ . \tag{3.5}$$

The inverse of T_{ba} is:

$$T_{ab} \equiv \left(\begin{array}{c|c} R_{ab} & t_{ab} \\ \hline 0_{1\times 3} & 1 \end{array} \right) = \left(\begin{array}{c|c} R_{ba}^{\top} & -R_{ba}^{\top} t_{ba} \\ \hline 0_{1\times 3} & 1 \end{array} \right) = \left(\begin{array}{c|c} R_{ba} & t_{ba} \\ \hline 0_{1\times 3} & 1 \end{array} \right)^{-1} = T_{ba}^{-1} . \tag{3.6}$$

Given two rigid bodies a and b, with transforms relative to a common frame of reference w we can compute the relative transformation between a and b as:

$$T_{ba} = T_{wb}^{-1} T_{wa} \ . \tag{3.7}$$

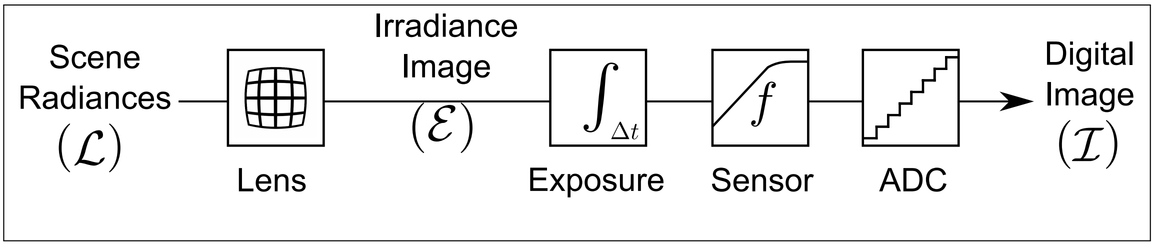


Figure 3.1: Basic components in the image acquisition pipeline. We calibrate for geometric lens distortion and also for the photometric sensor function when knowledge of the exposure time of an image is available. We perform geometric calibration to remove any non-linear geometric transformation in the image plane. We also perform photometric calibration given knowledge of imaging exposure time and sensor response functions. Calibration enables useful simplifications in dense tracking and mapping components.

3.2 Camera Calibration

In this section we describe the basic camera model that is used throughout the thesis. First we describe the geometric calibration that enables practical use of the simple camera model, transforming a $3D$ point in the frame of reference of the camera into a $2D$ point in the camera image. We then describe the photometric calibration possible for an image captured with knowledge of the exposure time of the frame. This enables a simplifying brightness constancy assumption relating the value measured in two or more views of a static Lambertian surface under fixed lighting. An outline of the simplified geometric and photometric transformations that take place during imaging is shown in Figure (3.1).

3.2.1 Geometric Calibration

Given a point $x_a \equiv \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}_a \in \mathbb{R}^3$ we define perspective projection to a point $u \in \Omega \subset \mathbb{R}^2$:

$$\dot{u} = \pi(x_a) \equiv \frac{1}{x_2} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}_a . \tag{3.8}$$

The *intrinsic calibration* matrix K is defined as:

$$K \equiv \begin{pmatrix} f_0 & 0 & p_0 \\ 0 & f_1 & p_1 \\ 0 & 0 & 1 \end{pmatrix} , \tag{3.9}$$

with focal length $(f_0, f_1)^\top$ and principle point $(p_0, p_1)^\top$. We can obtain the image co-ordinates of a projected point p_a:

$$\dot{p}_a = K\pi(x_a)\ . \tag{3.10}$$

We *back-project* the image co-ordinates p_a, back to a point x_a using a depth $z \in \mathbb{R}$ by:

$$\dot{x}_a = \left(K^{-1}\dot{p}_a\right) z\ . \tag{3.11}$$

where K^{-1} is computed explicitly by:

$$K \equiv \begin{pmatrix} \frac{1}{f_0} & 0 & -\frac{p_0}{f_0} \\ 0 & \frac{1}{f_1} & -\frac{p_1}{f_1} \\ 0 & 0 & 1 \end{pmatrix}\ . \tag{3.12}$$

Perspective projection (3.8) ensures that straight lines in the world project to straight lines in an image, but to relate elements in the geometric model to a pixel value in a real image, we must first remove any non-linear geometric transformation produced by the camera lens to obtain a *rectilinear* image. Relative to the rectlinear image which contains only a linear transformation from projected point locations in a camera to pixel co-ordinates, we will refer to the non-linear geometric transformation in the image plane to geometric *distortion*.

In particular, the lenses used in cameras throughout this thesis have a wide *field of view* (FOV) where the main component of the image distortion is radial. Assuming that the radial distortion is symmetric about a pixel $d \in \Omega$, then a function $R : \mathbb{R} \mapsto \mathbb{R}$, takes a Euclidean distance $r_u = \|d - u_u\|_2$ in a rectilinear image to the distance in a radially distorted image $r_d = \|d - u_d\|_2$. The mapping between a pixel $u_u \in \Omega$ from a rectilinear image to the corresponding radially distorted pixel location u_d is:

$$u_d \quad = \quad (u_u - d)\frac{R\left(r_u\right)}{r_u} + d. \tag{3.13}$$

The inverse function from a pixel in the radially distorted image back into a rectilinear image is:

$$u_u \quad = \quad (u_d - d)\frac{R^{-1}\left(r_d\right)}{r_d} + d. \tag{3.14}$$

Devernay and Faugeras (2001) modelled an ideal fish-eye lens by assuming that the distance between an image point and the principal point is proportional to the angle between the corresponding rectilinear ray connecting the imaged $3D$ point with the optic center and the optic axis. The corresponding radial distortion function is:

$$R(r_d) = \frac{1}{\omega} \arctan\left(2r_u \tan\left(\frac{\omega}{2}\right)\right) , \tag{3.15}$$

and its inverse is:

$$R^{-1}(d) = \frac{\tan(r_d\omega)}{2\tan\left(\frac{\omega}{2}\right)} . \tag{3.16}$$

We use the calibration procedure defined in **Devernay and Faugeras** (2001) to estimate the parameter ω together with the intrinsic calibration camera parameters (f_0, f_1, p_0, p_1) used in K, Equation (3.9). We assume that the center of distortion d is equal to the principal point (p_0, p_1) and compute a look-up table for the inverse function from undistorted to distorted pixel co-ordinates. Given a new distorted image, we proceed to obtain its rectilinear version by bilinear interpolation of the distorted image using the pre-computed inverse map at each pixel location u_u.

3.2.2 Photometric Calibration

The ability to photometrically calibrate imagery from a camera is often understated or neglected within standard visual SLAM systems. This is typically because the first operation of a sparse visual SLAM system is to condense an image to a set of discriminative points (corners, FAST, Harris etc.) which can be robustly described for use in a binary data-association framework.

In this work we would like to use all of the image data available in every frame, and as a first step it is very useful to recognise the physical image formation process that occurs in video capture. This enables us to transform the image pixel values into a form independent of the camera exposure time that we will call irradiance values. This is possible in practice because of an ability in modern digital video cameras to set or read the exposure over time.

Assuming a static scene, let $Z_j(u_i) \in \mathbb{R}$ denote the grey scale pixel value captured at pixel location $u_i \in \Omega$ at time j. The value is formed by the irradiance E at the corresponding sensor bucket being integrated over the exposure time Δt_j and transformed by the sensor response function f:

$$Z_j(u_i) = f\left(\Delta t_j E(u_i)\right) . \tag{3.17}$$

The function f models both the physical response of the sensor including specific properties of the imaging hardware (which includes saturation) and user defined operations such as applying a gamma curve, contrast or brightness change. In reality sensor manufacturers attempt to ensure that f is as linear as possible to achieve a faithful measurement of the physical energy arriving at the sensor.

Auto exposure control on the camera can be utilised to attempt to trade-off the two problems associated with fixed exposure video capture in natural environments. If the light levels are low, a longer exposure is required to ensure sufficient signal to noise to ratio; whereas if light levels are high, a shorter exposure may be necessary to remove value saturation. This naive point assumes a static imaging scenario, since the trade-offs involved with a moving scene become complicated by the fact that as image exposure increases, so does motion blur. Understanding these trade-offs presented by a moving camera within a visual SLAM setting is an area of current research, (**Handa et al.**, 2012).

If f^{-1} is known and the exposure time is available then we can transform the pixel values into irradiance:

$$f^{-1}\left(Z_j\left(u_i\right)\right) = E\left(u_i\right)\Delta t_j \ . \tag{3.18}$$

$$\Rightarrow E\left(u_i\right) = \frac{f^{-1}\left(\left(u_i\right)\right)}{\Delta t_j} \ . \tag{3.19}$$

The importance of obtaining the quantity E is that for a scene composed of Lambertian surfaces with constant illumination we can assume that pixel irradiance values can be associated across frames:

$$E_j\left(u_i\right) = E_k\left(\left(u_i\right) + v\right) \ , \tag{3.20}$$

where the displacement v maps the corresponding surface projections from image frame j into frame k. This simple irradiance constancy assumption is useful when searching for short baseline pixel correspondences between frames in a setting where the camera exposure is changing, as occurs when using automatic exposure control. Under irradiance constancy, given knowledge of the exposure across frames, a similarity measure between frames can be computed as a function of pixel differences.

Computing the response function f

There are two main approaches to estimating the response function f. Chart based approaches use a calibration chart of colours of specified irradiance under known lighting that can then captured over a range of exposure times using the sensor to be modelled. Assuming static capture of the chart by the camera, f^{-1} can be trivially constructed as a discrete table of 2^b entries where b is the bit depth of the pixel value. Each entry maps from measured pixel value to the known antecedent quantity $E\left(u\right)\Delta t_j$ given from the chart and pixel exposure time. Since the pixel value measurements will contain noise, it is more useful to robustly fit a parametric curve to a large number of irradiance-pixel value pairs.

Chart based calibration techniques, when performed with known illumination, can be used to produce a metric, physically meaningful sensor measurement. If no calibration chart is available, or the available illumination when performing calibration is fixed but unknown, we can only obtain the response function up to scale factor, noting that irradiance constancy still holds in this case. In the chartless self-calibration approach of **Debevec and Malik** (1997) f^{-1} is estimated by exploiting the assumed monotonicity of f and taking the logarithm of Equation (3.18):

$$g\left(Z_j\left(u_i\right)\right) \equiv \ln f^{-1}\left(Z_j\left(u_i\right)\right) = \ln E\left(u_i\right) + \ln \Delta t_j \tag{3.21}$$

Given a finite pixel value depth of b bits $Z_j\left(u_i\right) \in \left\{1, 2, \ldots, 2^b\right\}$, a non-parametric approximation of g can be estimated by jointly optimising for discretised g and log irradiance values $E\left(u_i\right)$. Given N pixel locations with observed values over a P known exposure times, we can minimise a quadratic error under a 2^{nd} derivative penalty on the solution of g to enforce smoothness of the function, yielding the following energy function:

$$\sum_{i=N}^{i=0} \sum_{j=P}^{j=0} \left(g\left(Z_j\left(u_i\right)\right) - \ln E\left(u_i\right) - \ln \Delta t_j\right)^2 + \lambda \sum_{Z=1}^{Z=2^b} \frac{\partial^2}{\partial^2 Z} g(Z)\ . \tag{3.22}$$

Expressing the unknown 2^b elements of the inverse function g concatenated with the unknown log irradiance values in vector form:

$$\mathbf{p} = \begin{pmatrix} \hat{\mathbf{g}} \\ \hat{\mathbf{E}} \end{pmatrix} \tag{3.23}$$

$$\hat{\mathbf{g}} = \left(g(1), g(2), \ldots g(2^b)\right)^\top \tag{3.24}$$

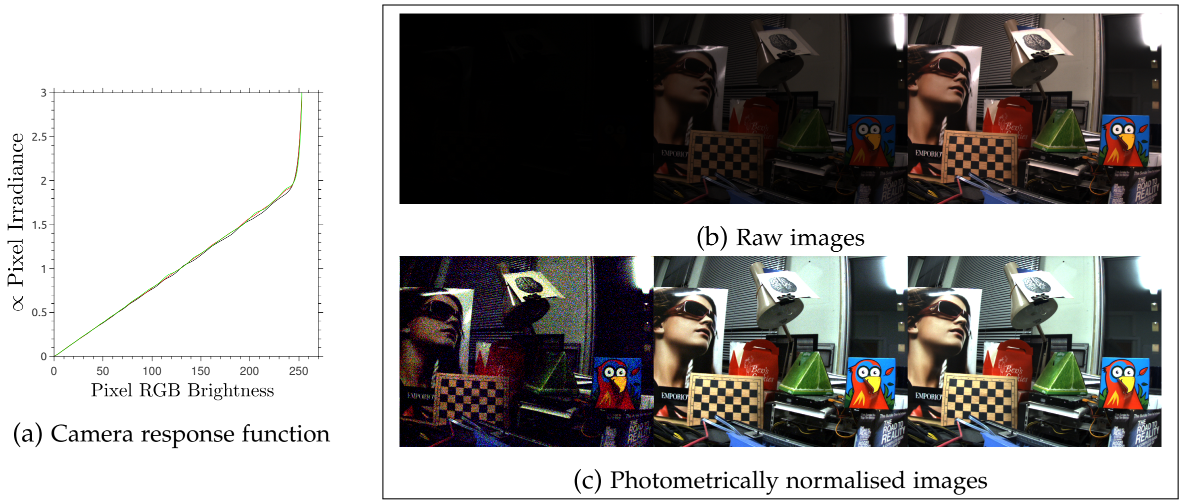
$$\hat{\mathbf{E}} = \left(\ln E_0, \ln E_1, \ldots \ln E_N\right)^\top\ , \tag{3.25}$$

we can write the data term and smoothness constraint as a linear system $\mathbf{Ap} = \mathbf{b}$:

$$\left(\begin{array}{c|c} \mathbf{D} & \mathbf{V} \\ \hline \lambda \boldsymbol{\nabla}^2 & \mathbf{0} \end{array}\right) \begin{pmatrix} \hat{\mathbf{g}} \\ \hat{\mathbf{E}} \end{pmatrix} = \begin{pmatrix} \mathbf{T} \\ \mathbf{0} \end{pmatrix}\ , \tag{3.26}$$

where the sub matrices $\mathbf{D} = \left[d_{r,c}\right]_{NP \times 2^b}$ and $\mathbf{V} = \left[v_{r,c}\right]_{NP \times N}$, $\mathbf{T} = \left[t_r\right]_{NP \times 1}$ and $\lambda \boldsymbol{\nabla}^2 = \left[\partial_{r,c}\right]_{N \times N}$ is a scaled 2^{nd} order derivative operator implemented as a finite difference matrix contributing the smoothness constraint on g.

Each pixel location i observed at exposure j forms a row $r = jN + i$ in $\mathbf{D}$ and $\mathbf{V}$ and a



(a) Camera response function

(b) Raw images

(c) Photometrically normalised images

Figure 3.2: Capture of three raw images with a change of one order of magnitude in exposure time between consecutive frames shown in (a). The images are normalised in (b) using photometric calibration parameters for the camera.

corresponding element in **T**:

$$\begin{aligned} d_{r,c} &= \begin{cases} w_{i,j} & \text{if } Z_j\left(u_i\right) = c \\ 0 & \text{otherwise,} \end{cases} && (3.27) \\ v_{r,c} &= \begin{cases} -w_{i,j} & \text{if } c = i \\ 0 & \text{otherwise,} \end{cases} && (3.28) \\ \Delta t_r &= w_{ij} \Delta t_j && (3.29) \end{aligned}$$

If $N(P-1) > 2^b$ then the system is overdetermined. In this case the least squares solution of $\|\mathbf{Ap} - \mathbf{b}\|_2^2$ for the estimated function $\hat{f}^{-1}$ can be recovered, together with the estimated irradiance values, by solving for $\mathbf{p}$ using the pseudoinverse of $\mathbf{A}$. Finally, element-wise exponentiation of $\mathbf{p}$ inverses the log transformation.

Example Photometric Calibration

Calibrating each channel of an (RGB) enables photometric normalisation of each frame from a video stream, resulting in irradiance constancy for pixels captured under different exposures in a static scene. A response function for a $flea2$ (rgb) colour camera is plotted in Figure (3.2a) together with a calibrated image sequence from multiple exposures of the same scene in Figure (3.2c). The saturation region for the sensor can clearly be seen in the response function. When using photometrically calibrated video we discard the highest value from the sensor across all colour channels, since pixel values which map through the saturation region will, through value quantisation, map onto the same irradiance value, breaking the irradiance consistency assumption.

3.3 Parametric Optimisation

In Chapter (1) we discussed the basic advantages of the direct image alignment approach introduced by **Lucas and Kanade** (1981) for obtaining an unknown transform $\hat{\mathbf{x}}$ between two image frames by minimising a whole image error:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \{E_w(\mathbf{x})\} \ , \tag{3.30}$$

$$E_w(\mathbf{x}) = \sum_{u \in \Omega} \psi \left(e(u, \mathbf{x})\right) \ . \tag{3.31}$$

Here ψ is a robust function chosen to reduce the cost associate with potential outliers resulting in photometric and geometric differences that are not modelled by the basic single pixel error function and:

$$e(u, \mathbf{x}) = \mathcal{I}_l(\mathbf{w}(u, \mathbf{x})) - \mathcal{I}_r(u) \ , \tag{3.32}$$

where the warp function $\mathbf{w}(u, \mathbf{x})$ is responsible for transforming a pixel $u \in \Omega$, from image frame r into frame l. In this section we outline the iterative solution to problems expressed in form of Equation (3.30).

3.3.1 Iterative Gauss-Newton Gradient Descent

We often need to perform the minimisation of an energy function which is not convex due to the data term being non-linear. For example, letting $\mathbf{w}$ be a similarity transformation in the image plane with $\mathbf{x} = (\theta, v_x, v_y)$ then:

$$\mathbf{w}(u, \mathbf{x}) = \begin{pmatrix} -\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} u + \begin{pmatrix} v_x \\ v_y \end{pmatrix} \ , \tag{3.33}$$

and so the error is clearly a non-linear function of the parameters. Furthermore, even when $\mathbf{w}$ is a linear map, the image data $\mathcal{I}_l(\mathbf{w}(u, \mathbf{x}))$ is not generally a linear function of $\mathbf{x}$. **Lucas and Kanade** (1981) proceed by Gauss-Newton gradient descent non-linear optimization to iteratively solve (3.30).

We obtain a local convex approximation of the whole image error using a second order

Taylor series expansion of $E_w(\mathbf{x}_0 + \Delta x)$:

$$\tilde{E}_w(\mathbf{x}_0 + \Delta x) \approx E_w(\mathbf{x}_0) + \boldsymbol{\nabla}_{\mathbf{x}} E_w(\mathbf{x}_0)\Delta x + \tfrac{1}{2}\Delta x^\top \boldsymbol{\nabla}^2_{\mathbf{x}} E_w(\mathbf{x}_0)\Delta x \;, \tag{3.34}$$

where $H(\mathbf{x}_0) = \boldsymbol{\nabla}^2_{\mathbf{x}} E_w(\mathbf{x}_0)$ is called the Hessian. If the second order Taylor series expansion at $\mathbf{x}_0$ is accurate then the solution can be obtained by stepping to the extremum for a convex function, attained at $\boldsymbol{\nabla}_{\Delta x}\tilde{E}_w = 0$.

Often obtaining the second order partial derivatives of E_w can be very computationally demanding, and since higher order terms are multiplied with larger powers of Δx the contribution to the error from higher order terms is diminishing. We therefore further approximate the function using a Gauss-Newton approximation of the Hessian term, requiring only first order gradients of the cost function:

$$\tilde{E}_w(\mathbf{x}_0 + \Delta x) = E_w(\mathbf{x}_0) + \sum_{u\in\Omega} \psi'(e(u,\mathbf{x}_0))J(u,\mathbf{x}_0)\Delta x + \tfrac{1}{2}\sum_{u\in\Omega} \Delta x^\top J(u,\mathbf{x}_0)^\top J(u,\mathbf{x}_0)\Delta x \;, \tag{3.35}$$

where the per pixel gradient vector is evaluated through the chain rule as a product between the error metric derivative *wrt* to the error:

$$\psi'(e(u,\mathbf{x}_0)) = \left.\frac{\partial\psi(e(u,\mathbf{x}))}{\partial e(u,\mathbf{x})}\right|_{\mathbf{x}_0} \;, \tag{3.36}$$

and the derivative of the warp function *wrt* the parameter vector:

$$J(u,\mathbf{x}_0) = \left.\frac{\partial\mathcal{I}_l(\mathbf{w}(u,\mathbf{x}))}{\partial\mathbf{x}}\right|_{\mathbf{x}_0} \;. \tag{3.37}$$

The minimising argument to equation (3.35) is obtained at the extremum:

$$\boldsymbol{\nabla}_{\mathbf{x}_0}\tilde{E}_w(\mathbf{x}_0 + \Delta x) = 0 \;. \tag{3.38}$$

Taking the the derivative of (3.35) *wrt* $\mathbf{x}_0$ results in a linear system in Δx:

$$\sum_{u\in\Omega} J(u,\mathbf{x}_0)^\top J(u,\mathbf{x}_0)\Delta x = -\sum_{u\in\Omega} \psi'(e(u,\mathbf{x}_0))J(u,\mathbf{x}_0) \;, \tag{3.39}$$

$$\Rightarrow \Delta x = -\Big(\sum_{u\in\Omega} J(u,\mathbf{x}_0)^\top J(u,\mathbf{x}_0)\Big)^{-1} \sum_{u\in\Omega} \psi'(e(u,\mathbf{x}_0))J(u,\mathbf{x}_0) \;. \tag{3.40}$$

The increment is solved for in practice using a Cholesky decomposition of the summed Hessian approximation. The updated parameter estimate is obtained simply by adding the incremental update onto the current estimate resulting in a new linearisation point:

$$\mathbf{x} \leftarrow \mathbf{x}_0 + \Delta x \ . \tag{3.41}$$

This new estimate is then used in an updated version of the warp function, proceeding to another iteration of the gradient descent which continues until convergence is achieved. We can test for convergence in practice either by assessing the reduction in the computed cost or the rate at which the cost is reducing, stopping when either falls below a predefined value. We must also take into account the available computational budget of a live running system in which the optimisation procedure is being used. There is no guarantee of convergence to the global minimum for the method since the original cost function is non-convex. Therefore, without a specified cost threshold which might indicate a solution for the application at hand, in practice, we run the optimisation for the maximum number of iterations possible given an available computational window, checking for any degeneracy that might occur in solving Equation (3.40).

3.4 Convex Optimisation

The previous section outlined an iterative optimisation approach applicable to general continuous non-convex cost functions comprising a sum of errors. However, in contrast to the low dimension parameter estimation required for whole image alignment, we now look to solve optimisation problems with hundreds of thousands of variables which arise when estimating the solution of every pixel in an image as required in image denoising and dense correspondence problems.

In this section we detail the optimisation tools that enable the structure present in these computer vision problems to be exploited, obtaining solutions that can be computed extremely efficiently in practice using modern parallel hardware.

Probabilistically modelling computer vision problems using a generative model of the available measurements conditioned on the model solution, together with some prior assumption on solution smoothness, has an equivalent energy form. Indeed, many vision problems can be cast within this framework where the solution $u \in \mathbb{R}^{M \times N}$ is obtained as the minimisation of an energy $\mathrm{E}(u)$:

$$\mathrm{E}(u) \triangleq \mathbf{D}(u,g) + \lambda \mathbf{R}(u) \tag{3.42}$$

$$\hat{u} = \min_{u \in \mathbb{R}^{M \times N}} \{\mathsf{E}(u)\} \ . \tag{3.43}$$

Here the data term $\mathbf{D} : \mathbb{R}^{M \times N} \mapsto \mathbb{R}$ is derived from the likelihood function $\propto -\ln(p(g|u))$

given some measurement g, and a regularisation term $\mathbf{R} : \mathbb{R}^{M\times N} \mapsto \mathbb{R}$ is derived from the model prior $\propto -ln(p(u))$, which takes on lower values for solutions with given desirable characteristics. The data term comprises an error function e, which is the error induced by generative model given the solution u and the available measurement g, and a positive penalisation function ψ_D:

$$\mathbf{D}(u,g) = \sum_{x\in\Omega} \psi_D\left(e\left(u(x), g(x)\right)\right), \tag{3.44}$$

and the regularisation term is a function of the solution smoothness, together with a second positive penalisation function ψ_R:

$$\mathbf{R}(u) = \sum_{x\in\Omega} \psi_R(s(u)(x)). \tag{3.45}$$

In a continuous domain and computing solution smoothness through $s(u) = \boldsymbol{\nabla} u$, we can define the variational optimisation problem:

$$E(u) = \int_\Omega \psi_D\left(e\left(u(x), g(x)\right)\right) \mathbf{d}x + \lambda \int_\Omega \psi_R\left(\boldsymbol{\nabla} u\right) \mathbf{d}x\ . \tag{3.46}$$

The calculus of variations provides a general condition to obtaining minima for this type of functional involving u and its derivatives by minimising the Euler-Lagrange Equation of the energy functional: $\frac{\partial E(u)}{\partial u} = 0$.

If error function e is linear in u and both ψ_R and ψ_D are convex functions, then since $\boldsymbol{\nabla} u$ is also a linear function of u the resulting energy functional is a sum of convex terms, resulting in a globally convex energy. In this case any solution obtained for $\frac{\partial E}{\partial u} = 0$ is also the global minimum. This has important ramifications for ensuring that a solution can be obtained very efficiently and is the focus of the remainder of this chapter.

3.4.1 Convex norms

In Chapter (1) we saw that the *quadratic cost* x^2 results from a Gaussian likelihood over x in probability form, while the absolute function is at the convex-concave boundary and provides the closest convex model for the associated probability distribution over spatial gradients of both depth maps and intensity images. The associated vector norms using the quadratic and absolute penalisation functions are summarised in table (3.1).

Minimisation under ℓ_1 norm is more robust to outliers, in that the maximum likelihood estimate of a random variable can be achieved when up to half of the samples present are outliers to the distribution. This is in contrast to the quadratic cost which is strongly influenced by outliers. However, the ℓ_1 penalty $|x|_1$, is non differentiable at $x = 0$. As we

PDF(x)	Cost	Primal	Vector Norm
Gaussian	Quadratic	$\lvert x\rvert_2^2 = x_i^2$	$\lVert\mathbf{x}\rVert_2 \triangleq \sum_{i=1}^{n} \lvert x_i\rvert_2^2$,
Laplacian	ℓ_1	$\lvert x\rvert_1 = \sqrt{x_i^2}$	$\lVert\mathbf{x}\rVert_1 \triangleq \sum_{i=1}^{n} \lvert x_i\rvert_1$
Hybrid	Huber	$\lvert x\rvert_h = \begin{cases} \lvert x\rvert_2^2 & \text{if } \lvert x\rvert \leq \alpha \\ \lvert x\rvert - \frac{\alpha}{2} & \text{if } \lvert x\rvert > \alpha, \end{cases}$	$\lVert\mathbf{x}\rVert_h \triangleq \sum_{i=1}^{n} \lvert x_i\rvert_h$

Table 3.1: Convex penalty terms.

will see below a solution to this is required to obtain the full power of the robustness of this norm.

The *Huber* penalisation (**Huber**, 1981), is a piecewise mixture of the quadratic and absolute functions:

$$|x|_h = \begin{cases} |x|_2^2 & \text{if } |x| \leq \alpha \\ |x| - \frac{\alpha}{2} & \text{if } |x| > \alpha, \end{cases} \tag{3.47}$$

measuring x^2 for small values of x and an absolute function for larger values. Hence, the Huber vector (pseudo-norm) combines both the differentiability of a quadratic cost function with the robustness to data outliers obtained when optimising under the ℓ_1 norm.

We also define the norms for use over vectors comprising elements of partial derivatives, we define the *quadratic* cost over an element in $\boldsymbol{\nabla} u$ by:

$$|\boldsymbol{\nabla} u(i,j)|_2^2 = \partial_x u_{i,j}{}^2 + \partial_y u_{i,j}{}^2 \ , \tag{3.48}$$

and for the ℓ_1 norm:

$$|\boldsymbol{\nabla} u(i,j)|_1 = \sqrt{\partial_x u_{i,j}{}^2 + \partial_y u_{i,j}{}^2} \tag{3.49}$$

with the Huber norm defined piecewise using both. We note that while the ℓ_1 coincides with the Euclidean norm for a scalar variable, when placed within the summation or integral over the domain of u which is a vector, the Euclidean norm takes the square-root of the sum of the squares, while the ℓ_1 sums the absolute values of elements in the norms argument. In using the ℓ_1 norm of a vector field of $\boldsymbol{\nabla} u(i,j)$ where each element $\in \mathrm{R}^2$ we first apply the ℓ_2 norm to each element and sum the resulting (absolute) values, which corresponds to the mixed $\ell_1 - \ell_2$ norm.

3.4.2 An Example of Convex Optimisation in Computer Vision

A classic convex optimisation based computer vision solution was presented by **Rudin, Osher, and Fatemi** (1992), with a variational solution to the image de-noising problem introduced in Chapter (1). The model demonstrated in Section (1.4.2), combines the quadratic norm on a data term corresponding to a Gaussian likelihood and the ℓ_1 norm over $\boldsymbol{\nabla} u$, which we saw previously is obtained by taking a Laplacian distribution in the image prior:

$$E_{ROF} = ||\nabla u||_1 + \lambda ||u - f||_2^2 \, . \tag{3.50}$$

The associated partial differential equation obtained from the Euler- Lagrange equation for the model is:

$$\frac{\partial E_{ROF}}{\partial u} = \boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|} \right) + \frac{1}{\lambda}(u - g) = 0 \, , \tag{3.51}$$

where $\boldsymbol{\nabla}\cdot$ is the divergence operator. Unfortunately we can see a number of problems in obtaining the solution $\frac{\partial E_{ROF}}{\partial u} = 0$. First, the derivative of the ℓ_1 is undefined at 0, second the non-linearity of the derivative results in there being no direct closed form solution. The simplest solutions to both of these problems were presented in the original paper, (**Rudin et al.**, 1992). First, non-differentiability of the ℓ_1 norm is resolved by using an $\epsilon - regularised$ form:

$$|x|_\epsilon = \sqrt{x^2 + \epsilon} \, , \tag{3.52}$$

where a small value ϵ removes the non differentiability of the norm. An explicit time marching gradient descent is then performed with time step τ,

$$\frac{u^{t+1} - u^t}{\tau} = -\boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{\nabla} u^t}{|\boldsymbol{\nabla} u^t|} \right) - \frac{1}{\lambda}(u^t - g) \, . \tag{3.53}$$

Such a gradient descent gets around a number of issues with the solution of the PDE, resulting in a simple point-wise update of the solution variable given an initialisation on u at $t = 0$. Unfortunately the method is very slow to converge and in any case results only in the solution of a modified version of the original energy in Equation (3.50).

More efficient iterative solutions exist for this model including the straightforward use of a semi-implicit time marching scheme (**Vogel and Oman**, 1996), in which the denominator in the ϵ regularised ℓ_1 derivative term is the only lagged component in the discretised gradient, resulting in a sparse linear system of equations. In the next section we return to the original discrete energy and introduce a modern solution to the *ROF* following the more recent results of **Chambolle and Pock** (2011), which has applicability to general convex energy optimisation problems. These solutions make use of techniques, described next, that mitigate the non-differentiability of the class of energy functional we are interested in, resulting in efficient solutions to convex optimisation problems without resorting to approximation. We note that there is large body of work on convex optimisation, and that at the time of writing the work by **Chambolle and Pock** (2011) which contains extensive comparison with historically related techniques, presents the state of the art for the type

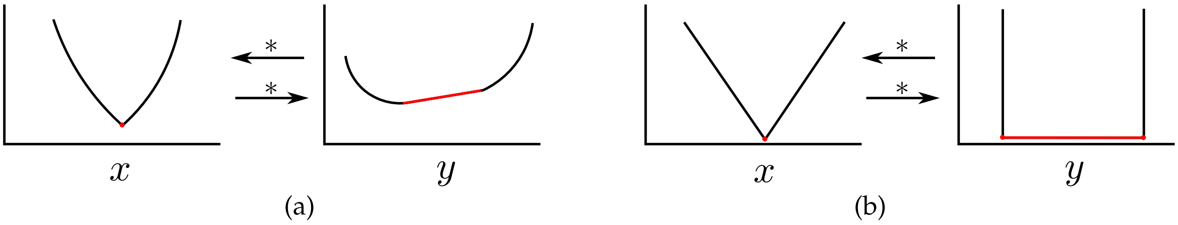


Figure 3.3: The Lengendre-Fenchel transform takes non-differentiable points in $F(x)$ to affine sections in F^*y. Figure adapted from **Touchette** (2005), an accessible introduction to the convex conjugate. In (a) and (b) non-differentiable points are transformed to affine regions in the functions conjugate. In (b) the ℓ_1 function is transformed to the indicator function.

and the size of the convex problems described here.

3.4.3 Duality and the Convex Conjugate

The convex conjugate $F^*(x)$ of a function $F(x)$ where $x \in \mathbb{R}^n$ is defined through the Legendre-Fenchel transform,

$$F^*(y) = \sup_{x \in \mathbb{R}^n} \{\langle x, y \rangle - F(x)\} \ , \tag{3.54}$$

where $\langle ., . \rangle$ computes the inner product between the so-called *primal* variable x and its *dual* y. The importance of the convex conjugate lies in the transformation of functions containing points of non-differentiability, such as in the ℓ_1 penalty, (**Touchette**, 2005). In this case, the action of the Legendre-Fenchel transformation creates affine parts in $F^*(x)$ resulting in a function which, over the domain of y, *is* differentiable. The convex conjugate of the ℓ_1 norm of a vector $x \in \mathbb{R}^n$, illustrated geometrically in Figure (3.3b), is the indicator function in dual variable y:

$$F^*(y) = \delta(y) = \begin{cases} 0 \text{ if } \|y\|_1 \leq 1 \\ \infty \text{ otherwise} \ . \end{cases} \tag{3.55}$$

Therefore we can write the ℓ_1 norm for the primal variable $x \in \mathbb{R}^n$as:

$$\|x\|_1 = \max_{y \in Y}(\langle x, y \rangle - \delta(y)) \ , \tag{3.56}$$

where the set Y is given by:

$$Y = \{y \in \mathbb{R}^n, \|y\|_\infty \leq 1\} \ . \tag{3.57}$$

The Huber-norm $\|x\|_h$ which is defined piecewise, comprises a quadratic region for $\|x\|_1 \leq \alpha$ which results in a quadratic conjugate function within that region:

$$F^*(y) = \frac{\alpha}{2}\|y\|_2^2 \quad \forall ||y|| \leq \alpha \ , \tag{3.58}$$

while the conjugate of the Huber-norm in the region $\|x\|_1 > \alpha$ leads again to an indicator function:

$$F^*(y) = \begin{cases} \frac{\alpha}{2} \text{ if } \alpha < \|y\| \leq 1 \\ \infty \text{ otherwise} \ , \end{cases} \tag{3.59}$$

therefore we can re-write the Huber-norm as:

$$\|x\|_h = \max_{y \in Y} \left(\langle y, \nabla x \rangle - \delta_Y(y) - \frac{\alpha}{2}||y||^2 \right) . \tag{3.60}$$

3.4.4 Discretisation

It will be convenient to use vector versions of the $2D$ solution or other images or variables $a \in \mathbb{R}^{M \times N}$. We will therefore use column vectors $\mathbf{a} \in \mathbb{R}^{MN}$ containing the stacked elements of a, where $i = x + My$,

$$a = [a_{m,n}]_{M \times N} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \mapsto \mathbf{a} = [a_i]_{MN \times 1} \begin{pmatrix} a_{1,1} \\ a_{1,2} \\ \vdots \\ a_{1,n} \\ a_{m,1} \\ \vdots \\ a_{m,n} \end{pmatrix} . \tag{3.61}$$

Within the energy minimisation setting, when the data and regularisation terms are linear in u, we can conveniently rewrite the discrete energy in matrix-vector notation:

$$E(\mathbf{u}) = ||D\mathbf{u} - \mathbf{g}||_D + ||R\mathbf{u}||_R \ . \tag{3.62}$$

We note that often it is convenient simply to imply the use of matrix-vector notation, when it is clear that the context of the formulation is discrete, hence we will not use the bold vector notation and simply use u and g instead.

Discrete Gradient and Divergence

We can define the discrete operator ∇ in matrix form $\nabla = [\boldsymbol{\nabla}_{i,j}]_{2MN \times MN}$, where the resulting partial derivative elements are stacked in a vectorial fashion:

$$\nabla \mathbf{a} \triangleq \begin{pmatrix} \frac{\partial}{\partial x} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \frac{\partial}{\partial x} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial}{\partial x} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \frac{\partial}{\partial x} & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \ddots & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \frac{\partial}{\partial y} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial}{\partial y} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \frac{\partial}{\partial y} & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \ddots & \frac{\partial}{\partial y} \end{pmatrix} \begin{pmatrix} a_{1,1} \\ a_{1,2} \\ \vdots \\ a_{1,n} \\ a_{m,1} \\ \vdots \\ a_{m,n} \end{pmatrix} = \begin{pmatrix} a^x_{1,1} \\ a^x_{1,2} \\ \vdots \\ a^x_{1,n} \\ a^x_{m,1} \\ \vdots \\ a^x_{m,n} \\ a^y_{1,1} \\ a^y_{1,2} \\ \vdots \\ a^y_{1,n} \\ a^y_{m,1} \\ \vdots \\ a^y_{m,n} \end{pmatrix} \tag{3.63}$$

Here, each dimensions partial derivative $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ is computed using a forward difference operation. We note that the divergence operator $\nabla \cdot$, in matrix form is simply the transpose of the gradient matrix i.e. $\nabla \cdot \equiv \nabla^\top$. Extensive details and definitions for such a discretisation are available in **Handa et al.** (2011); **Zhu** (2008); **Chambolle and Pock** (2011).

3.4.5 Primal-Dual methods

We now have the tools to look again at minimising the ROF denoising model in Equation (3.50) using convex optimisation. Using the Lengendre-Fenchel transform, we can replace the primal form of total variation regularisation $\|\boldsymbol{\nabla} u\|_1$ of the ROF model, with the primal-dual form in Equation (3.56). The resulting primal-dual energy minimisation is a saddle-point problem in the primal solution variable $u \in \mathbb{R}^{MN}$ and dual variable $p \in \mathbb{R}^{2MN}$:

$$\min_u \max_{p \in P} \left\{ E\left(u, p\right) \triangleq \langle p, \nabla u \rangle + \frac{\lambda}{2} \|u - f\|_2^2 - \delta_P(p) \right\}, \tag{3.64}$$

where $\delta_P(p)$ is the indicator function, and the set P is:

$$P = \left\{ p \in \mathbb{R}^{2MN}, \|p\|_\infty \leq 1 \right\} . \tag{3.65}$$

This concave-convex optimisation problem can be solved using a simple alternation of gradient ascent on the dual variable:

$$p^{n+1} = \Pi_1(p^n + \sigma \nabla u^n) \tag{3.66}$$

where the projection onto the convex set P is performed pointwise:

$$\Pi_{\zeta}(p) = \frac{p}{\max\left\{1, \frac{\|p\|}{\zeta}\right\}} \,. \tag{3.67}$$

This is then followed by fixing p^{n+1} and performing gradient descent on the primal variable,

$$u^{n+1} = \frac{u^n + \tau \nabla \cdot p^{n+1} + \tau \lambda g}{1 + \tau \lambda} \,. \tag{3.68}$$

Since we will use this primal-dual optimisation approach in several algorithms we give a basic derivation for the above scheme. The minimum of the convex function in Equation (3.64) is obtained at the function extremum where $\boldsymbol{\nabla}_{u,p} E(u,p) = 0$, which we solve by a sequence of alternating gradient descent steps:

1. Computing the derivative with respect to p i.e. $\partial_p E(u,p)$,

$$\partial_p E(u,p) = \partial_p \left(\langle p, \nabla u \rangle + \frac{\lambda}{2} ||u - g||_2^2 - \delta_P(p) \right) \tag{3.69}$$

$$\partial_p \left(\langle p, \nabla u \rangle \right) = \nabla u \tag{3.70}$$

$$\partial_p \left(\frac{\lambda}{2} ||u - g||_2^2 \right) = 0 \tag{3.71}$$

$$\partial_p \delta_P(p) = 0 \tag{3.72}$$

$$\Rightarrow \quad \partial_p E(u,p) = \nabla u \tag{3.73}$$

   Fixing the current value of variable u at u^n, we compute a gradient ascent on the dual variable:

$$\frac{p^{n+1} - p^n}{\sigma} = \nabla u^n \,, \tag{3.74}$$

   which is solved for p^{n+1} incorporating the constraint on p through projection onto the unit ball, given in Equation (3.66).

2. Computing the derivative with respect to u i.e. $\partial_u E(u,p)$,

$$\partial_u E(u,p) = \partial_u \left(\langle p, \nabla u \rangle + \frac{\lambda}{2} \|u-g\|_2^2 - \delta_P(p) \right) \tag{3.75}$$

$$\partial_u \left(\langle p, \nabla u \rangle \right) = \partial_u \left(-\langle u, \nabla \cdot p \rangle \right) = -\nabla \cdot p \tag{3.76}$$

$$\partial_u \left(\frac{\lambda}{2} ||u-g||_2^2 \right) = \lambda(u-g) \tag{3.77}$$

$$\partial_u \delta_P(p) = 0 \tag{3.78}$$

$$\Rightarrow \quad \partial_u E(u,p) = -\nabla \cdot p + \lambda(u-g) \tag{3.79}$$

Fixing the current value of variable p at p^{n+1}, we compute a gradient ascent step on the primal variable:

$$\frac{u^{n+1}-u^n}{\sigma} = \nabla \cdot p^{n+1} - \lambda(u^{n+1}-g) \; , \tag{3.80}$$

which is solved for u^{n+1}, given in Equation (3.68).

3.5 Parallel Computation

An important element in choosing the above optimisation strategies is the efficiency and certainly to some extent, the simplicity with which they can be implemented on modern commodity massively-parallel computing hardware. The modern general purpose graphics processing unit (GPGPU) is a descendent of graphics cards designed to efficiently perform the matrix-vector computations dominant in $3D$ graphics applications (**Nvidia**, 2008). In practice, modern GPGPU hardware and parallel programming languages provide efficient computation for problems which are trivially parallelisable, i.e. algorithms which can be modularised into independently operating local sub computations that make use of a small local region of memory. In this section we highlight issues for the optimisation schemes introduced above in relation to efficient implementation using GPGPU.

3.5.1 Computing Primal-Dual Updates for Convex Optimisation

The updates for the primal variable (Equation 3.68) and dual variable (Equation 3.66) use only sparse matrix-vector multiplications and element-wise operations that can be efficiently computed *in-place*. The core computation of the dual variable update needs the point-wise gradient of the primal variable, and also performs the projection of the variable onto the ball in Equation (3.67), computed using an element-wise `max(.,.)`. This computation is performed for all elements $i = x + My$ in parallel:

```
  float d_px_ = d_px[i] + du_x*tau;
  float d_py_ = d_py[i] + du_y*tau;
  float len = fmaxf(1, sqrtf( d_px_*d_px_ + d_py_*d_py_)/lambda);
  d_px[i] = d_px_/len;
  d_py[i] = d_py_/len;
```

Here `du_x` is $\frac{\partial u}{dx}$ computed for element $i = x + My$ using u^n as,

```
if (x==M-1) return 0;
else return u[i+1] - u[i];
```

Similarly we compute `du_y` from $\frac{\partial u}{dx}$. Fixing p the core computation for the primal variable update computed at all elements $i = x + My$ in parallel is simply:

```
  d_u[i]=(d_u[i]+tau*div_p+tau*lambda*d_g[i])/(1.0f+tau*lambda);
```

which requires a point wise divergence computed as,

```
  float div_p = dxm + dym;
```

where `dxm` is $\frac{\partial p_x}{dx}$ computed on p^{n+1},

```
  if (x==0) return p[i];
  else if (x==M- 1) return -p[i-1];
  else return p[i] - p[i-1];
```

and similarly for `dym` computing $\frac{\partial p_y}{dy}$.

3.5.2 Gauss-Newton Iterations for Parametric Optimisation

The GPGPU implementation of solutions using the iterative Gauss-Newton energy minimisation method outline in Section (3.3), is only slightly more involved. Looking at the Gauss-Newton update in Equation (3.40), computation of each element in the summand of the element-wise gradient with $\psi'(e(u, \mathbf{x}_0))$ and the Jacobian $J(u, \mathbf{x}_0)$ given the current parameter estimate can clearly be obtained in parallel at each pixel u. However, we require the summation of these elements that form the weighted normal equations, $\sum_{u\in\Omega} J(u, \mathbf{x}_0)^\top J(u, \mathbf{x}_0)$ and $\sum_{u\in\Omega} \psi'(e(u, \mathbf{x}_0))J(u, \mathbf{x}_0)$.

It is important to minimise global memory access across the parallel processors and threads holding each partial $J(u, \mathbf{x}_0)$ since such operations are expensive. Fortunately, as summa-

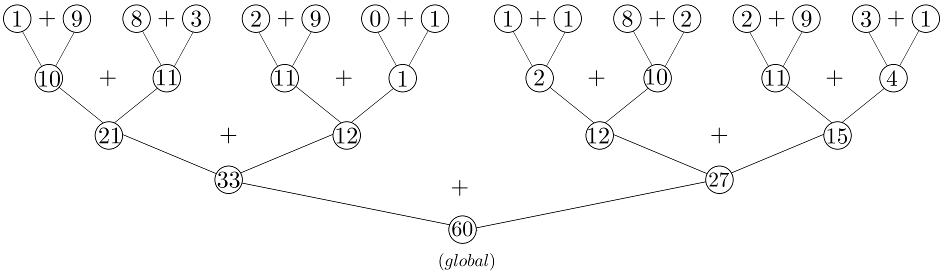


Figure 3.4: Example tree reduction. Since the addition operator is associative and commutative we can compute the summation between any pairs of the partial sum: $\mathcal{O}(\log(n))$ parallel steps.

tion over the vectors and matrices are both commutative and associative the result can be obtained via a tree reduction, also called a pre-fix parallel sum, as illustrated in Figure (3.4). We further note that for an n dimensional parameter vector, the Hessian approximation from $\sum J^\top J$ is a symmetrical $n \times n$ matrix, and therefore only the the upper or lower triangular matrix need be reduced. While the summation is performed efficiently using the tree-reduction, we then copy the resulting summed elements from the Hessian approximation and gradient vectors to the host computer memory, where an efficient solution to the linear system can be obtained.

4

Convex Optimisation Based Depth Map Denoising

Contents

In this chapter we investigate approaches for acquiring surface measurements from a real-time video stream where camera poses are known. In subsequent chapters we will develop the full dense reconstruction system making use of the surface estimation methods developed in this chapter, leading to a demonstration of the advantages that can be gained by coupling the passive surface estimation together with dense model reconstruction.

Standard sparse visual SLAM pipelines typically treat correspondences that can be obtained between a set of frames as point abstractions with each pixel independent from its neighbours. Here we will make use of the fact that dense correspondences result from the projection of continuous surfaces into the camera frames, and therefore have continuity in the image plane. This basic assumption enables correspondences to be obtained at pixels where explicit feature extraction and matching techniques would fail. Furthermore the techniques we explore for estimating depth maps, by using all of the available image data under the assumption of surface continuity, result in higher precision sub-pixel dense correspondences than traditional sparse feature-based matching techniques.

4.1 Outline

Multiple view depth estimation which subsumes the more traditional stereo estimation where a rectified and synchronised camera pair provides the input images, is one of the most widely studied topics in computer vision due to the abundance of applications that require surface measurements (**Scharstein and Szeliski**, 2001). While depth estimation is itself often a sub component for the more complete dense reconstruction of surfaces in a scene that will be developed in this thesis, the constrained correspondence problem that must be solved for a single depth map is to a large extent the core of any dense reconstruction pipeline. First we describe the specific representations used in the majority of multi-view stereo estimation methods, which generalise stereo disparity parametrisation for two-view correspondences used in the traditional stereo case.

4.1.1 Parametrisation of Depth

A depth map non-parametrically represents a scenes surface geometry as viewed by a camera in the form of an image $D : \Omega \mapsto \mathbb{R}$, where a single point in the image $D(u)$ corresponds to the distance to the surface. Typically the distance is measured perpendicular to the image plane, such that the correspondence of point x in the depth map frame observing distance $d = D(x)$ can be computed in a second co-observing camera with relative transform T_{kr} by $x' = \mathbf{w}(x, k, d)$, where:

$$\mathbf{w}(x, k, d) \triangleq \pi(\mathrm{K}\mathrm{T}_{kr}\mathrm{K}^{-1}\dot{x}D(x)) \ . \tag{4.1}$$

Alternatively there is an advantage in enabling D to instead represent inverse depth at each pixel $\xi(x) = \frac{1}{D(x)}$. Regular sampling in inverse depth leads to regular sampling along the epipolar lines in the supporting frames and is a generalisation of the common pixel disparity representation used in two view rectified stereo estimation. Given a known minimum and maximum depth in the reference frame, the inverse depth variable can be defined as $\xi : \Omega \mapsto [0, 1]$ to represent the fixed range $[d_{min}, d_{max}]$ as:

$$\xi(x) = \frac{1}{\xi_{range}D(x) + \xi_{min}} \ , \tag{4.2}$$

where $\xi_{range} \equiv \frac{1}{d_{min}} - \frac{1}{d_{max}}$ and $\xi_{min} \equiv \frac{1}{d_{max}}$. In this form, $D(x) = 1$ and $D(x) = 0$ respectively represent the nearest and furthest depth values in the scene.

Given this representation, the problem of multiple view depth estimation is then how to compute D in reference frame r given $m \geq 1$ other frames which are co-observing the scene.

4.1.2 Chapter Overview

Substantial reviews on the topic of stereo estimation are provided by **Dhond and Aggarwal** (1989); **Scharstein and Szeliski** (2001); **Brown et al.** (2003) each using similar taxonomies with which to categorise approaches. One critical categorisation occurs between *local* methods which attempt to solve for the depth map on a point by point or local area basis using only statistical properties of image intensities within the region, and *global* solutions which combine the local data term with a global regularisation term relating to the solution smoothness, forming an energy functional where minimum energy configurations relate to a depth map solution.

Stereo Data Terms

In Section (2.3) we reviewed the global optimisation background of this technically advanced field, discussing the salient problems and the key features of dense depth map estimation algorithms within the continuous optimisation setting in particular.

Since all multiple view stereo estimation techniques require the computation of a data term component, in Section (4.2), we detail the techniques involved in computing the dataterm, the assumptions made and associated problems that are common to approaches.

Depth Map Denoising using Convex Optimisation

In the second half of the Chapter we specifically investigate the depth map denoising methodologies previously introduced in Chapter (1). These techniques enable the incorporation of the smoothness assumptions over the solution depth map within a convex global optimisation framework. We begin in Section (4.3) with an overview of the general approach.

In section (4.4) we restate the basic winner takes all multi-view stereo data term. We use a basic patch based data term that is robust to local additive illumination changes together with a simple occlusion handling mechanism for short baseline video frames.

In section (4.5) we develop the global optimisation based depth map denoising approach providing several models to investigate the general performance possible using state-of-the-art convex optimisation based primal-dual formulations. We look at several smoothness assumptions and cover both single and multiple input depth map denoising frameworks. For each model we provide a primal-dual formulation resulting in a convex optimisation problem that is efficiently solved on the GPU using gradient descent. We are interested in large scale qualitative differences and gross inaccuracies that occur in the models which we highlight using a small-baseline multiple view stereo video dataset.

We continue our investigation of convex multiple stereo in Chapter (5), looking at the alternative strategy of linearising the data terms to produce a sequence of convex optimisation steps which can be solved using the same techniques developed in this section. In that chapter we will further evaluate the results obtained with the depth map denoising approach, and draw some conclusions about the applicability of the techniques for use in dense visual SLAM systems.

4.2 Data Terms and Local Approaches

The *brightness constancy assumption* underlies the development of many stereo data terms. Given two frames viewing a scene, where a reference frame image value $\mathcal{I}_r(x)$ has an associated depth estimate $D(x)$ at pixel x, the brightness constancy assumption states that the value at the corresponding pixel in the second image should be the same if $D(x)$ corresponds to a scene surface co-visible in both frames,

$$\mathcal{I}_k(\mathbf{w}(x,k,d)) = \mathcal{I}_r(x). \tag{4.3}$$

Given this assumption, we can define an error function computed over a set $m \geq 1$ of co-observing views, computed for each pixel in the reference frame evaluating dissimilarity of corresponding pixels induced by a depth estimate d:

$$\epsilon(x,d) = c\left(\mathcal{I}_r(x), \mathcal{I}_0(\mathbf{w}_0(x,d)), \mathcal{I}_k(\mathbf{w}(x,k,d)), \ldots, \mathcal{I}_m(\mathbf{w}_m(x,d))\right) \ . \tag{4.4}$$

In general the cost function c can be based either on combined statistics computed for all corresponding pixels such as value variance, or by summing individual pairwise costs computed between each view value and the reference value. The most basic stereo estimation procedures use the brightness constancy assumption directly to define a photometric error:

$$\epsilon(x,d) = \sum_{k=0}^{m} \psi(\mathcal{I}_r(x) - \mathcal{I}_k(\mathbf{w}(x,k,d))) \ . \tag{4.5}$$

An estimate of D can then be computed by selecting the depth at each pixel that obtains the minimum of ϵ:

$$\hat{D}(x) = \underset{d}{\operatorname{argmin}}\, \epsilon(x, d) \ . \tag{4.6}$$

This winner-takes-all approach was illustrated in Chapter (1), and can be efficiently computed using the plane-sweep algorithm given a particular discretisation over the parametrised depth variable, (**Collins**, 1996).

4.2.1 Problems with Brightness Constancy

The simple winner-take-all approach makes a number of assumptions about the scene being viewed and the imaging arrangement used in capturing the images. Foremost, it requires that the brightness constancy for corresponding pixels holds, which is possible when the surface material is Lambertian so that the appearance is not a function of viewing direction. Furthermore taking the single pixel data term minimum does not take into account ambiguity in the data-term given finite camera resolution and measurement accuracy, or the multiple sources of noise between imaging the surface in one frame and the next, especially when the camera is moving through the scene.

In reality most natural scenes comprise at best partially non-Lambertian surfaces; digital cameras have a fixed spatial resolution; the image exposure often changes to ensure reduced image blur while optimising for the signal to noise ratio in the image; and natural scenes are often imaged under light sources which are not static. Finally the imaging process contains a number of sources of noise, including value quantisation and sensor noise from thermal properties of the electronics. Furthermore, the original assumption of co-visibility of a surface region in all frames is often wrong due to object occlusions. For these reasons the $1D$ cost functions $\epsilon(x, d)$ do not present us with a clear single minimum and the resulting winner takes all strategy, using the single pixel dataterm based on the brightness constancy error, results in errors in correspondence.

Local stereo methods aim to obtain a better error function ϵ, by increasing the distinctiveness of regions at the correct solution, resulting in a more clearly defined single minimum in the function.

4.2.2 Error Function, Aggregation and Post-processing

The error and aggregation functions are defined over both the spatial extent of a reference image region and also across the multiple views. Improvements to local error functions can be broken into a number of distinctive elements with many algorithms making use of the vast permutations possible with them. Review articles on the subject have introduced established components used by practitioners through the earlier years and include the excellent review by **Scharstein and Szeliski** (2001) that systematised evaluation of the two view-stereo algorithms. Here we look at salient differentiators that have been instrumental in forming the modern local stereo algorithms and are most relevant to decisions in our own work.

Region Descriptor

Replacing the single pixel comparison, an area around the target pixel can be used and some descriptor formed within a small patch of pixels. Used in combination with a measure of the closeness of image regions under the descriptor, also called the penalisation function.

Linear descriptors: The most basic of these is a fixed rectangular patch of neighbouring pixels, with the error measured using a sum of squared difference (SSD). In a multiple view depth setting this leads to the sum of sum of squared differences (SSSD):

$$\epsilon(x,d) = \sum_{k=0}^{m} \sum_{j \in n(x)}^{m} (\mathcal{I}_r(j+x) - \mathcal{I}_k(\mathbf{w}(x,k,d)+j)^2 \,. \tag{4.7}$$

Here $n(x)$ defines the set of pixel translations that define the patch of pixels centred on x. An alternative form of the SSSD instead computes the difference on the warped patch:

$$\epsilon(x,d) = \sum_{k=0}^{m} \sum_{j \in n(x)}^{m} (\mathcal{I}_r(j+x) - \mathcal{I}_k(\mathbf{w}(x+j,k,d))^2 \,. \tag{4.8}$$

SSSD is not robust to illumination changes. For this reason a normalised cross correlation has been more widely used, (**Hannah**, 1974; **Faugeras et al.**, 1993). Alternative strategies for obtaining increased robustness to the failure of brightness constancy make use of an image pre-processing stage, applying the symmetrical Laplacian ($\boldsymbol{\nabla}^2$) operator to the image, or incorporate the invariance within the error measure, e.g. using the zero mean normalised cross correlation. A large number of variations on this approach have been introduced in the literature which aim to provide invariance to changes in the image intensity of corresponding pixels across frames, **Szeliski** (2010).

Robust descriptors: Further work established more robust descriptors based on a non-linear transformation of the values in a local region. The census transform (**Zabih and Woodfill**, 1994) produces a descriptor by computing the differences between each pixel value in the local patch and the central pixel value. A single bit binary representation of the regions pixels is then obtained by comparison of the sign of the difference. Such descriptors can be matched using the efficient Hamming distance metric and have been shown experimentally to be more robust than linear correlation strategies (**Bhat and Nayar**, 1996), since unlike the linear descriptors, such ordinal descriptors can tolerate any radiometric change in intensity so long as the relative ordering of pixel values remains the same. However, information is clearly lost in such transforms. In situations when only subtle variations in texture exist within a region the ability to increase discrimination by aggregation across views, possible with linear descriptors, is lost.

Penalisation Function

A second consideration involves the cost function or error norm, under which errors between pixels in a descriptor are accumulated (**Szeliski and Zabih**, 1999). In the case of the simple SSSD measure above, robustness can be increased by using an m-estimator over the quadratic photometric cost. Modelling the likelihood of the photometric error given a model solution results in a corrupted Gaussian distribution (**Black and Anandan**, 1993). The use of a truncated cost function provides robustness to the outliers that can exist in matches using rectangular patches, which we discuss further in the next section. By ensuring only high quality matches are accumulated, aggregation over the patch is more likely to be restricted to those neighbouring pixels which do lie on the same surface with a similar depth value.

Aggregation and Filtering

An implicit assumption of the traditional region descriptors operating over fixed spatial patches is that all depths within the patch are the same. Such an assumption leads to biases towards depth map solutions with regions that are fronto-parallel to the reference image plane. In practice the pixels in a region can straddle depth discontinuities, which can lead to occlusions in a subset of views. Moreover, the fronto-parallel surface assumption holds only for a very restrictive set of real world scenes, since most scenes are composed of various surfaces at out of plane orientations or containing high curvature and thin structures.

Slanted windows: If the local geometry projected into a reference frame pixel is explicitly parametrised as a local planar patch then an extra two degrees of freedom must be sampled over, resulting in an increased computation time if exhaustively searching over the cost volume for the winner-take-all minimum. However, such explicit modelling does lead to higher quality results.**Gallup et al.** (2007) achieved real-time multiple view stereo estimation, extending the plane-sweep approach to range over a selected set of sweeping orientations, providing a more accurate and discriminative cost function from which the winner-take-all solution can be selected.

Alternatively, segmentation of the reference image into local regions can be used together with a fronto-parallel patch based depth estimation within each region to initialise a set of planes. These can then be used to recompute an oriented patch based cost and aggregation (**Tao et al.**, 2001; **Zhang et al.**, 2008). Such post-processing can achieve better results for scenes comprising planes but reduces the ability to capture curved or thin structures. Also, if the initial depth estimation fails to capture the slanted or curved surfaces the secondary process will not necessarily lead to an improvement.

The *stereo patch-match* framework introduced by **Bleyer et al.** (2011) makes use of an efficient, iterative non-deterministic search through the parameters of a full continuous planar parametrisation by restricting the space to search first over parameters currently used by a pixels neighbour. The result is a useful propagation of neighbourhood information without altering the cost associated with a per pixel solution, as occurs in the global regularisation approaches we will discuss in the next section. Instead, since nearby pixels that lie on the same surface will have similar plane parameters, the method simply searches for a local minimum at locations in the parameter space biased by the solution its neighbour has taken on in a previous iteration, converging to the per pixel local minimum of the planar parametrised depth map. In practice the algorithm is halted prior to convergence, and produces results with qualities approaching the global optimisation approaches which use solution regularisation.

Weighted aggregation: A second highly successful track of work directly addresses the issue that some pixels within a patch do not belong to the same surface as the pixel at the center of the patch. Spatial aggregation is presented with a trade-off in setting the optimal size of the window used. A larger window is required to increase signal to noise ratio and discrimination amongst the many possible matches of the cost function. However, this increases the risk of aggregating cost over pixels which lie on different surfaces, and introduces error in the cost where regions are severely warped by projective transformation within the patch across the multiple images. In contrast smaller windows (and in the limit a single pixel) suffer less from incorrect aggregation, but are not very discriminative. By altering the patch size based on measurements of the quality of correlation obtained, researchers investigated adaptive size windows (**Kanade and Okutomi**, 1994; **Kang et al.**, 2001), leading to improvements in reconstruction of finer structures and depth boundary pixels. Alternatively, aggregation can be performed on the cost volume directly using a $3D$ weighting mask (**Scharstein and Szeliski**, 2001), enabling filtering across depth hypothesis as well as within the local image space.

Yoon and Kweon (2006); **Gong et al.** (2007) introduced adaptive support weights making the assumption that pixels within a patch which are closer to the center pixel and have the same colour as the center pixel are more likely to take on a similar depth value. Hence, an adaptive support weight is produced for each pixel in patch using a combination of pixel intensity or colour similarity and pixel proximity with the reference pixel. This approach proved extremely effective and unlike the pre-segmentation approaches that aggregate within regions of unbroken colour similarity, does not rely on the hard segmentation problem being solved. (Hosni et al., 2009) extended the adaptive support weights to bring the concept of unbroken paths from the explicit segmentation paradigm via weighting by geodesic distance within a patch, resulting in higher quality depth boundaries.

Hosni et al. (2011) presented a real-time capable approach based on an adaptive cost volume filtering paradigm. They achieve speed up over the previous cost volume filtering systems by replacing the bi-lateral filter used to compute the aggregation weights with the guided image filter of He et al. (2010). This can be efficiently implemented as a box-filter, enabling a trivially parallelisable implementation on the GPU platform, leading to a three order of magnitude decrease in run time over the original adaptive support weight method of Yoon and Kweon (2006).

Occlusion handling: Often performed in a post processing stage in two view approaches, given depth maps computed using the left and right as reference, a consistent depth in the first image should lead to a correspondence in the second which has an associated depth which maps back to the original first image pixel, due to the co-visibility of the surface. Occluded pixels can therefore be detected and discarded based on a left-right consistency check (Fua, 1991). Besides the adaptive weighting strategies that decrease errors due to occlusion, reasoning can be performed to remove pixels occluded in the multiple views from the cost aggregation. Specifically in the multiple view depth estimation scenario, when $m \geq 2$ possible observations of the same surface are given, the best subset of views can be selected on the basis of matching quality. Along these lines, Kang et al. (2001) proposed a temporal selection of the best half sequence. Assuming that the trajectory of a moving camera is locally smooth, occlusion in one half of the sequence may be resolved in the other with respect to the pivoting reference frame if the scene under observation forms a single depth discontinuity tangential to the camera motion.

Post-process: A number of strategies exist to improve the depth map *after* it has been extracted as the local cost function minimum. In particular Scharstein and Szeliski (2001) provide a review of the many algorithms which perform interpolation of the cost function to obtain higher precision estimates of the data term minimum. Other techniques attempt to use the local minimum as an initialisation for a local parametric optimisation as previously discussed for plane fitting within a segmented region. A parametric optimisation to obtain sub-pixel precision can also be performed using standard gradient descent style optimisations directly on the intensity data (Lucas and Kanade, 1981), although there is a clear link between interpolation of the cost volume and using interpolation in the image space during gradient descent which often makes the extra computation redundant. Alternatively, working directly on the estimated depth map extracted from the data term minimum, improvements can be obtained by denoising. In practice depth map denoising attempts to optimise a combination of the local data term minimum with a smoothness constraint common to all global stereo estimation approaches.

4.3 Depth Map Denoising Approaches

In contrast to the full global optimisation problem using the complete multi-view stereo data term, in this section we will instead make the simplifying assumption that the data term is convex in the solution by following the depth map denoising (DMD) approach from Section (3.4). While the resulting optimisation is clearly sub-optimal, it presents an important paradigm for practical depth map estimation due to the efficiency with which the sub problems can be solved: first we obtain the data term minimum and then solve a fully convex denoising problem. Therefore the input to the denoising algorithm is independent of both the number of images used to obtain the data term, and also the resolution in quantisation of the depth variable, both of which alter the computational complexity of minimisation with the original data term. We now describe the two main approaches to depth map denoising.

4.3.1 Single Depth Map

Pock et al. (2007b) highlighted that many of the high quality results obtained for publicly available two-view stereo datasets from (**Scharstein and Szeliski**, 2001), make use of accurate segmentation using the reference image, enabling boundaries to be accurately reconstructed. They proposed a unified framework for joint colour and depth image segmentation within the Mumford-Shah (MS) segmentation model **Mumford and Shah** (1989), in which multiple local minimum based data terms provide the depth map inputs which must be jointly segmented with the reference colour image. The result is a piecewise smooth approximation of both the colour and depth inputs.

A number of the local stereo approaches are based on the bilateral filtering method of (**Tomasi and Manduchi**, 1998), where image weighted aggregation within the photometric cost volume, is followed by extraction of the per pixel cost minimum. Despite their purely local computation, several variations have shown state of the art performance. Based on the insights gained in such systems, **Yang et al.** (2007) developed a super resolution approach for range images, showing that given only a low resolution depth map with a high resolution colour reference frame as input they can reconstruct an approximate higher resolution cost volume, perform bilateral filtering on the cost volume using a colour weight computed on the colour reference frame, and then extract a higher resolution denoised depth map as the per pixel minimum of the smoothed volume. They provide quantitative results using their approach as a post-processing for many of the original algorithms listed on the two-view stereo dataset comparison page from **Scharstein and Szeliski** (2001). Their results are a good demonstration of the practical benefits gained in assuming that the region continuity and edge information present in the reference image provides a strong indication of boundaries in real-world depth maps.

4.3.2 Multiple Depth Maps

An alternative approach to computing and denoising a single depth map estimated using all available frames is to compute multiple depth maps from subsets of the frames and then combine them into a single depth map using a multiple image denoising approach.

Koch et al. (1998) developed a correspondence linking approach for multiple view stereo in an attempt to combine the benefits associated with small and wide baseline stereo systems. They compute pairwise depth maps over an image sequence using a local patch based approach and then compute both forwards and backwards correspondences through all depth images. Correspondence is established based on projecting the depth estimate into the neighbouring depth frame and rejecting the link if the corresponding depth estimate at the projected pixel location is outside of a defined distance threshold. They gain easier correspondence with reduced occlusions from the short baseline stereo combined with an increased triangulation angle for correspondences linked over wider baselines, resulting in higher quality depth maps.

Merrell et al. (2007) made use of explicit reasoning with the visibility constraint that exists between multiple depth map measurements of a scene obtained from multiple neighbouring views. For a set of co-observing depth maps to be physically consistent, the transfer of points between any pair of depth maps should not result in the measurements from one frame occluding measurement in another. Due to noise in the measurement process this visibility constraint is often broken, and a heuristic approach can be taken to find each depth value in a reference frame which breaks the constraint in the least number of neighbouring views.

Pock et al. (2011) exploit the fact that for reconstruction of distant scenes, images captured of the scene are approximately modelled through orthographic projection. They compute multiple depth maps from temporal pairs of images using the semi-global matching approach of **Hirschmüller** (2005) and project each depth estimate onto a common plane. In contrast to the explicit fusion approach described above, they formulate a global multiple depth map image denoising problem, handling errors in the data term using a robust penalisation function, together with the TGV regularisation term discussed in Section (4.5.2).

In the remaining sections of this chapter we will now turn to detailing specific depth map denoising models that we have investigated for use in the dense visual SLAM pipeline. We begin in the next section with a description of the multi-view stereo data term that we will use both in a depth map denoising algorithm, and then again in Chapter (5), where we look at its use in a full global optimisation framework. Specifically, we will now look at making use of the convex optimisation framework outlined in Section (3.4) to produce efficient, novel, depth map denoising algorithms for use in our dense visual SLAM pipeline, exploiting the efficiency with which the models can be computed on parallel hardware, (**Pock et al.**, 2008b).

4.4 Small Baseline Multi-View Stereo Data Terms

We now detail the specific data term model we will use in the convex optimisation based depth map denoising approaches we investigate in the remaining sections of this chapter.

4.4.1 Normalised Patch Error

We obtain robustness to local additive image irradiance variations and noise by modifying the patch based SSSD from Equation (4.7), to a local mean subtracted similarity measure under a robust norm ψ_D. The penalised similarity measure between the reference r and image k at inverse depth d is:

$$\rho_P(x,k,d) = \sum_{j \in n(x)} \psi_D \left(\mathcal{I}_r(x+j) - \mu_r(x) - \mathcal{I}_k(x'+j) + \mu_l(x') \right) \ , \tag{4.9}$$

where $x' = \mathbf{w}(x,k,d)$ is the projection of into image k from reference pixel x at depth d, and μ is a pre-computed Gaussian convolved versions of an image $\mathcal{I}$ using variance σ_P^2:

$$\mu(x) = \left(\mathcal{N}_{\sigma_P^2} * \mathcal{I} \right)(x) \ . \tag{4.10}$$

The multiple view data term is then simply:

$$\rho_P(x,d) = \sum_{k \in \mathcal{K}} \rho_P(x,k,d) \ . \tag{4.11}$$

We note that this operation is similar to the gain compensated patch data term used in (**Gallup et al.**, 2007), and it is not the same as computing the sum over warped patches where the images have had the local average subtracted in a pre-processing step.

As previously discussed in Section (4.2.2) aggregation over a patch as defined in Equation (eqn:normalisedpatch), assumes all frames are viewing a fronto-parallel surface, and ignores inter-frame rotation. We have found that when using an image collection from a temporal sliding window with a video rate stream captured from a hand held camera, that this assumption holds quite well in practice.

4.4.2 Pixel-wise Minimum

We discretise the depth variable into m points such that $d \in \mathcal{M}$ are steps, linear in inverse depth: $\mathcal{M} = \left\{0, \frac{1}{m}, \frac{2}{m}, ..., \frac{m-1}{m}, 1\right\}$. A depth map is then trivially obtained as pixel-wise minimum searching over the possible inverse depth values:

$$d_{\mathcal{K}}^{min}(x) = \underset{d \in \mathcal{M}}{\operatorname{argmin}} \, \rho(x, d) \tag{4.12}$$

Depth Confidence: We obtain a measure of confidence for each pixel's depth estimate following the approach of **Matthies et al.** (1989) by scaling the data term cost at the minimum by the curvature of the data term at that point:

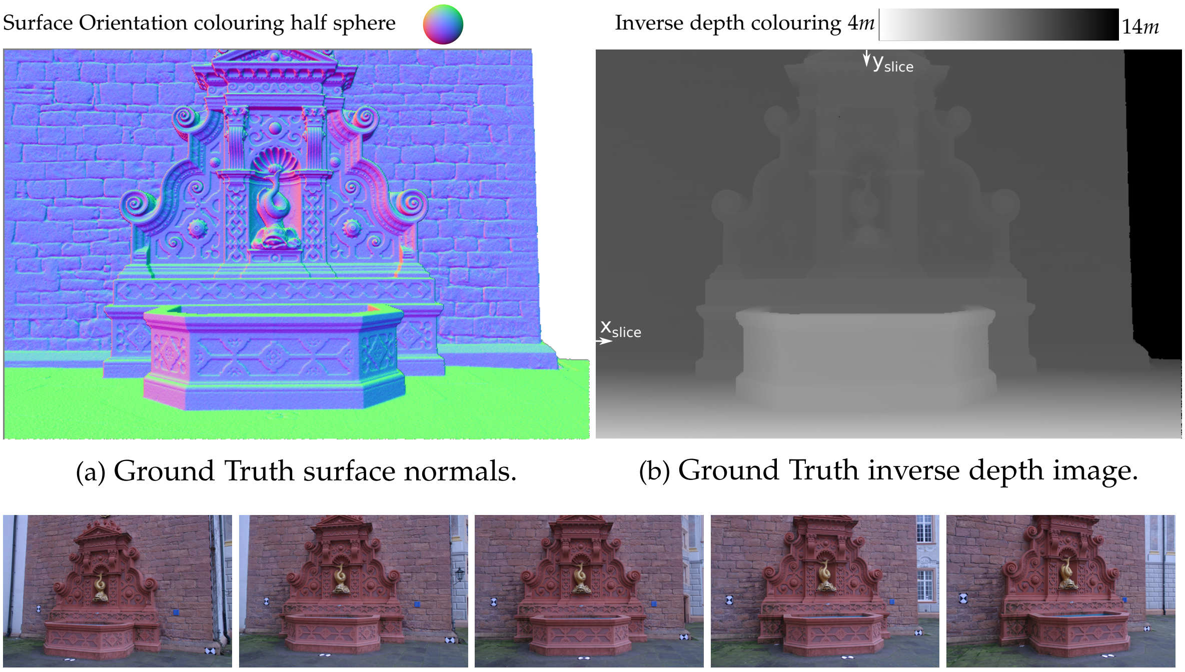
$$\sigma_{\mathcal{K}}^2(x) \propto \frac{\rho(x, d_{\mathcal{K}}^{min}(x))}{\nabla_d^2 \rho(x, d_{\mathcal{K}}^{min}(x))} \,, \tag{4.13}$$

where we compute the discrete ∇_d^2 on the data term using a 3 point central difference.

Data Term Interpolation: We obtain an interpolated depth solution using a 3-point parabola fit centred at the data term minimum which is equivalent to performing a Newton iteration style numerical gradient descent step using the first and second derivative :

$$d_{\mathcal{K}}^{min}(x) + \frac{\nabla_d \rho(x, d_{\mathcal{K}}^{min}(x))}{\nabla_d^2 \rho(x, d_{\mathcal{K}}^{min}(x))} \,. \tag{4.14}$$

We note that both the depth confidence and data term interpolation make an assumption about the locally convex nature of the data term minimum. We therefore first ensure that the interpolation of the minimum is valid by testing that the gradient descent step is less than one whole inverse depth interval. If this is not the case we reject the interpolation and use the discrete minimum instead, also setting a low default confidence for the depth estimate at that pixel.


Surface Orientation colouring half sphere
Inverse depth colouring $4m$
$14m$
y_{slice}
x_{slice}

(a) Ground Truth surface normals.

(b) Ground Truth inverse depth image.

Figure 4.1: Ground truth depth map data and views of the calibrated fountain-P11 sequence from **Strecha et al.** (2008) which we use to compare features of the depth map denoising models. (a) and (b) show the ground truth surface normal and inverse depth values of reference image 5 from the data set. Also marked on (b) are locations used later in this section at which slices through the solutions of depth map denoising algorithms are extracted for comparison.

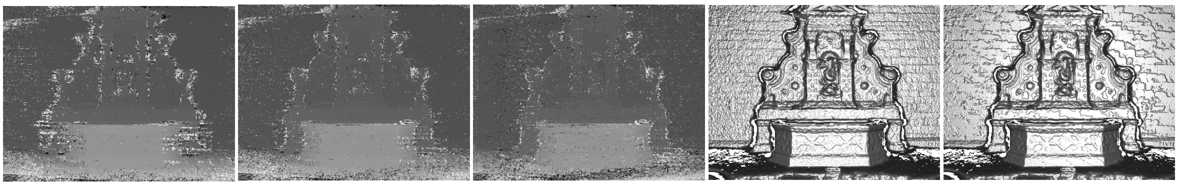

(a) (b) (c) (d) (e)

Figure 4.2: Illustration of the data term only depth map estimation using the occlusion robust data term minimum proposed in Equation (4.19). The data term is computed using the five frame image input from the downsampled fountain-P11 data from **Strecha et al.** (2008) shown in Figure (4.1). (a) Occlusion from complete data term minimum $s = 0$, (b) trading off the minimum from either the complete data term or from the left or right set $s = 0.3$, (c) using only the minimum from the left or right set $s = 1$. We demonstrate the result of the weighted-Huber denoising model from Section (4.5) with (d) and without (e) the cost function interpolation from Equation (4.14). In this experiment we downsample the original dataset from 3072×2048 to 480×320 pixels, used for real-time capable evaluation of a GPGPU implementation for most of the models outlined in this Chapter. A quantitative analysis of the stereo consistency check is performed against the ground truth in Figures (4.3) and (4.4).

4.4.3 Handling Occlusions

Since this simple data term does not explicitly take into account occlusion of the surface, errors can accumulate when a reference pixel is observable only in a subset of frames. Fortunately our N frames are extremely short baseline images from moving video camera, reducing the size of occlusions within a small window of frames about the reference. Furthermore, we can make a simple assumption regarding the visibility of the surface. Choosing a reference frame in the middle of a linear trajectory with the other frames on either side (which is reasonable for short baseline video) the simplest form of depth discontinuity, tangential to the camera motion, will lead to either the left or right half of the frames having co-visibility of the surface with the reference. The left and right half sequence data term costs are:

$$\begin{aligned} \rho_{\mathcal{A}}(x,d) &= \sum_{k=1}^{\frac{N}{2}} \rho(x,k,d)), \\ \rho_{\mathcal{B}}(x,d) &= \sum_{k=\frac{N}{2}}^{N} \rho(x,k,d)). \end{aligned} \tag{4.15}$$

Kang et al. (2001) proposed to take the minimum cost produced by the error induced by the left or right frame set:

$$d_{\mathcal{H}}^{min}(x) = \underset{d \in \mathcal{M}}{\operatorname{argmin}} \left\{ \min(\rho_{\mathcal{A}}(x,d), \rho_{\mathcal{B}}(x,d)) \right\} . \tag{4.16}$$

While this best half sequence approach improves errors at discontinuities, it unfortunately reduces the quality of the data term minimum for surfaces that are co-visible across all frames by reducing the number of observations by half. We can instead take the best of the three possibilities: either the best solution is given by the left or right half sequence, or by the estimate obtained using all of the images. Given the winning estimates from the left and right half sequences:

$$d_{\mathcal{A}}^{min}(x) = \operatorname*{argmin}_{d \in \mathcal{M}} \rho_{\mathcal{A}}(x, d) \tag{4.17}$$

$$d_{\mathcal{B}}^{min}(x) = \operatorname*{argmin}_{d \in \mathcal{M}} \rho_{\mathcal{B}}(x, d) \ , \tag{4.18}$$

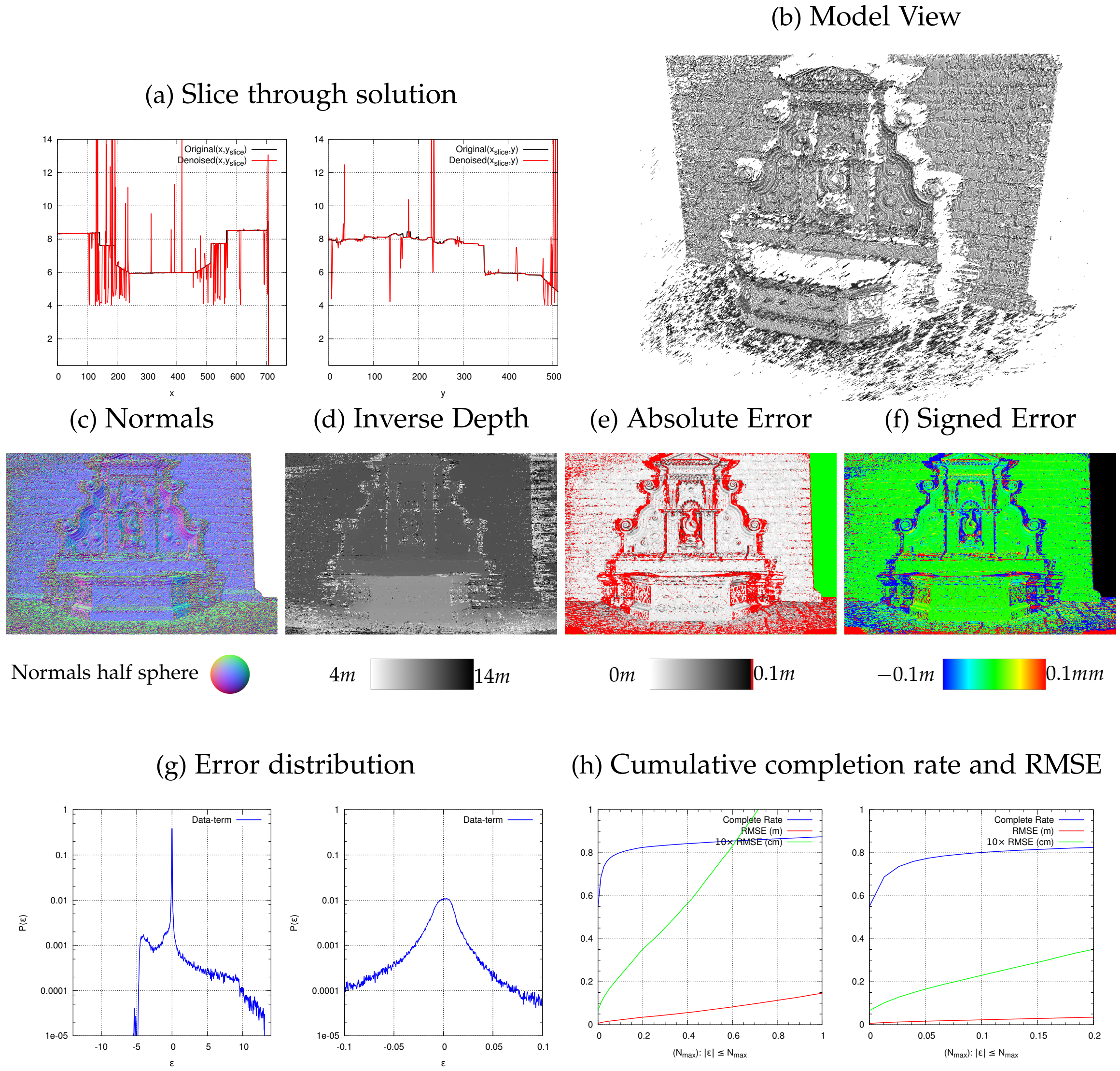
and the complete sequence $d_{\mathcal{K}}^{min}(x)$, we propose the following:

$$d^{min}(x) = \begin{cases} d_{\mathcal{K}}^{min}(x), & \text{if} \quad \frac{\min(c_{\mathcal{A}}^{min},\, c_{\mathcal{B}}^{min})}{c_{\mathcal{K}}^{min}} < r \\ d_{\mathcal{A}}^{min}(x), & \text{if} \quad c_{\mathcal{A}}^{min} < c_{\mathcal{B}}^{min} \\ d_{\mathcal{B}}^{min}(x), & \text{otherwise} \end{cases} \tag{4.19}$$

Here $c_K^{min} = \rho_{\mathcal{K}}(x, d_{\mathcal{K}}^{min}(x))$ and likewise for c_A^{min} and c_B^{min}. The function returns $d_{\mathcal{K}}^{min}(x)$ for $r = 0$, and the half sequence minimum $d_{\mathcal{H}}^{min}(x)$ for $r = 1$. A trade-off between the two is obtained by choosing $0 < r < 1$, enabling all frames to be used unless the cost obtained by all frames is greater than that obtained in either frame, determined by the cost ratio.

Figure (4.2) qualitatively illustrates the improvement in using a value for r which balances the best half sequence data term with the complete data term. In each case, settings are fixed to use five images from the sub-sampled fountain-P11 dataset from **Strecha et al.** (2008), Figure (4.1), using the highlighted frame as reference for the depth estimation. The neighbouring ± 2 frames (shown) are used in the data term only depth map estimation, using the interpolated minimum cost value for a 3×3 pixel mean subtracted patch data term from Equation (4.9) with $\sigma = 0.3$. The data term is discretised into 256 inverse depth values with the minimum and maximum depth value being set from the ground truth model bounding box available from **Strecha et al.** (2008).

We analyse the effectiveness of the consistency check further in Figures (4.3) and (4.4) using an increased image and solution resolution of 768×512 pixels. In this case, and further *quantitative* analysis with this dataset throughout this chapter, we use the higher resolution images since it represents a resolution which is still near real-time capable on the current top generation commodity GPGPU (requiring an increase of 2× more computation time over the lower resolution input), but importantly which suffers less distortion in the sub-sampled solution and ground truth geometry, enabling a more detailed analysis of the



(a) Slice through solution

(b) Model View

(c) Normals

(d) Inverse Depth

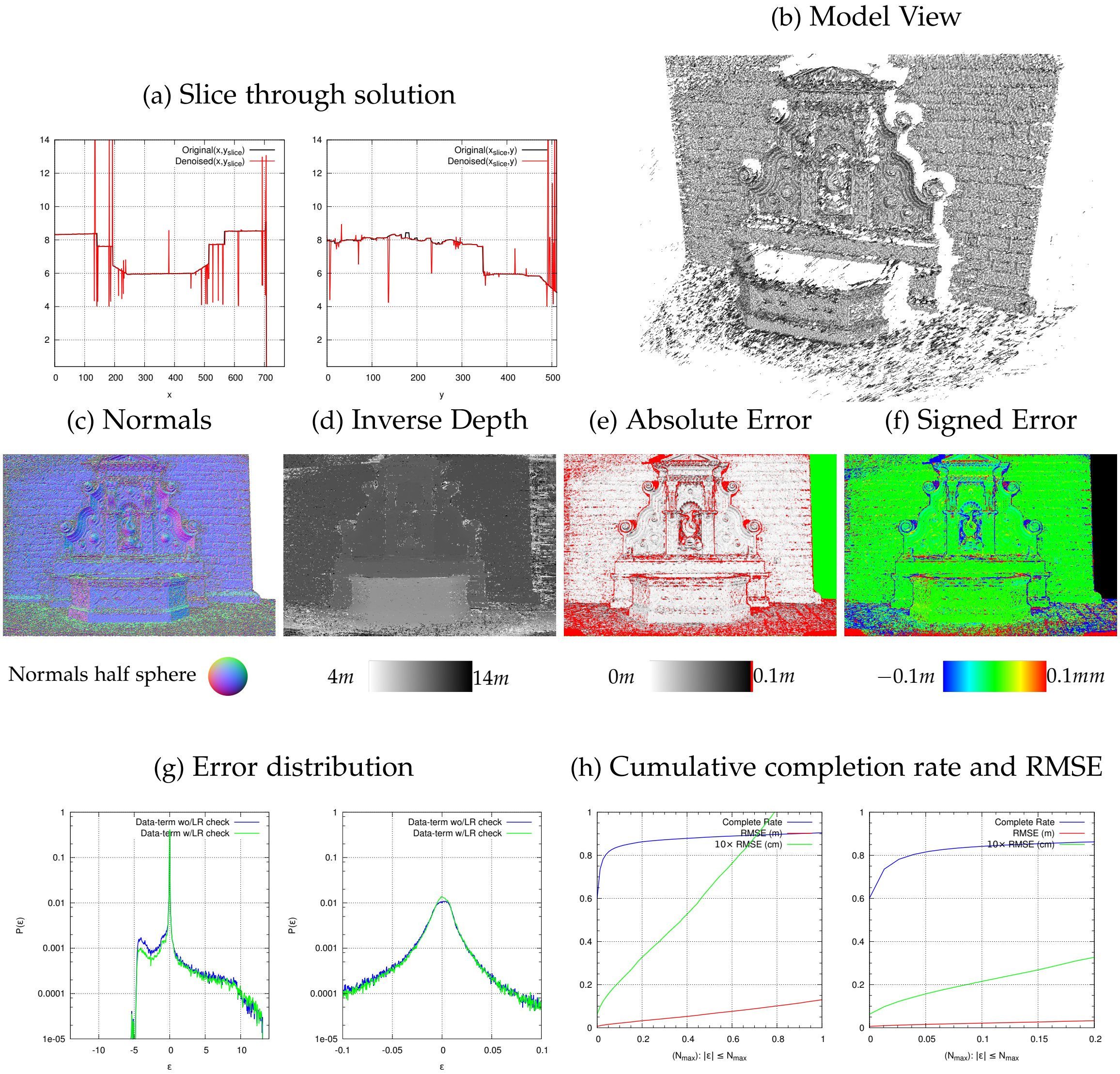
(e) Absolute Error

(f) Signed Error

(g) Error distribution

(h) Cumulative completion rate and RMSE

Figure 4.3: Data-term only without right-left consistency check. The error image (e) uses a grey scale to encoded absolute error to the ground truth depth at each pixel up to $0.1m$, is red for solution points with $> 0.1m$ absolute error. Green pixels encode that have no ground truth depth. The signed error is rendered in (f) with saturation at $\pm 0.1m$ error. In (h) we compute the RMSE and image completion (solution fill) ratio given a specified thresholding, N_{max}, on the absolute error, $|\epsilon|$, in the depth map.



(b) Model View

(a) Slice through solution

(c) Normals

(d) Inverse Depth

(e) Absolute Error

(f) Signed Error

(g) Error distribution

(h) Cumulative completion rate and RMSE

Figure 4.4: Data-term only with left-right consistency check. We plot the error histogram (g) for the consistency checked depth map together with the error distribution from the unchecked depth map. It is clear that left-right consistency check reduces errors near occlusion boundaries.

higher spatial frequency surface regions that are lost at a lower resolution. We compare the pixel-wise minimum depth map with and without the left-right consistency check on against the ground truth depth map. In Figure (4.4) We note a clear decrease in gross outliers at depth discontinuities with the left-right consistency check on.

4.5 Depth Map Denoising with Convex Optimisation

4.5.1 First order smoothness denoising

Given the local stereo data term minimum d, we proceed to compute a denoised depth map u, using a variational image denoising approach as introduced in Section (3.4.2). The denoised solution is obtained by minimising a global energy with a simple pixel wise error data term and solution smoothness based regularisation term:

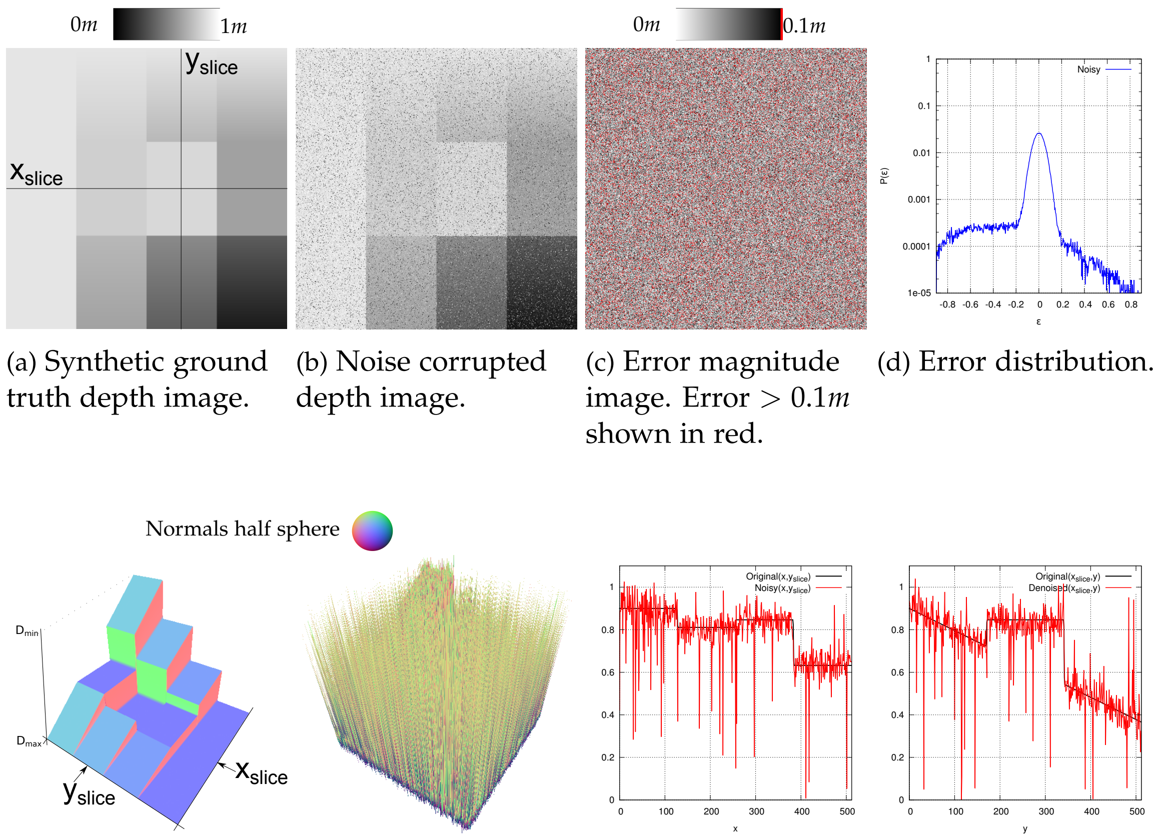
$$\min_u \left\{ \int_\Omega \psi_\mathcal{D}(u - d)\ dx + \lambda \int_\Omega \psi_\mathcal{R}(\mathrm{A}(u))\ dx \right\} . \tag{4.20}$$

Here the data term error measures the difference between the desired solution u and the depth map measurement d obtained from the local stereo method. We now detail a number of variational models that provide powerful denoising capabilities. For each model introduced we will illustrate its performance on a noise corrupted synthetic depth map dataset shown in Figure (4.5), enabling a comparison of each model solution against the ground truth.

In Chapter (1), we saw that within a depth map denoising setting the ℓ_1 norm is the closest convex model to the non-convex regularisation term that would minimise the energy associated with underlying gradient statistics of natural depth images. Using a quadratic penalisation on the data term $\psi_\mathcal{D}(s) = \frac{1}{2}s^2$ and Total Variation (TV) regularisation of the solution $\psi_\mathcal{R}(A(u)) = |\nabla u|_1$, **Rudin, Osher, and Fatemi** (1992) introduced the *ROF* image denoising model, demonstrated in Figure (4.7) (see Section 3.4.2 for more details on this model):

$$\min_u \left\{ \frac{1}{2} \int_\Omega (u - d)^2\ dx + \lambda \int_\Omega |\nabla u|\ dx \right\} . \tag{4.21}$$

TV of the function u uses the ℓ_1 norm penalisation. In contrast to a quadratic cost this allows discontinuities in the solution to form, since for any combination of increasing values between two function points including a complete homogeneous region followed a jump to the second value, TV measures the same cost (**Pock**, 2008). Minimisation of Equation (4.21) therefore results in solutions which are broken into piece-wise constant regions that minimise the joint data-term and regularisation energy shown.



(a) Synthetic ground truth depth image.

(b) Noise corrupted depth image.

(c) Error magnitude image. Error $> 0.1m$ shown in red.

(d) Error distribution.

(e) Ground truth surface normals.

(f) Noisy surface normals.

(g) Slice x.

(h) Slice y.

Figure 4.5: Ground truth synthetic depth image with noisy corrupted version. (a) Ground truth synthetic depth image, showing markings for the $x_{slice} = 256$ and $y_{slice} = 384$ slice positions. (b) Noise corrupted version of the ground truth image, using a mixture of Gaussian additive noise with $\sigma = 0.2$, with a uniformly random sampling of 8% of the pixels further corrupted with a uniform random value limited to be within the valid depth range, simulating outliers in a depth map data term. (c). Resulting error image, all values over $|\epsilon| = 0.1$ are coloured red, otherwise error magnitude is encoded from smallest to largest using white to black values. (d) Error distribution between noisy and ground truth synthetic images. Note we will show error histograms with a $\log_{10}$ scale on the y axis. We note that the the error resulting from the uniform pixel corruption is not symmetric reflecting the non uniform sampling of depth values in the synthetic image. (e,f) 3D normal surface rendering of the ground truth and noisy depth images. While the valid depth range is constrained by projection to be within $0m$ and $1m$ the ground truth has (D_{min}=0.1m, D_{max}=0.9m). (g,h) Slice through x and y dimensions of the image overlaying the noisy depth map with the ground truth value. We provide a demonstration of the denoising capabilities of the models introduced in this section in Figures (4.8-4.10), and also provide for comparison the result of simple Gaussian convolution in Figure (4.6).

If the probability distribution of the data term likelihood is not Gaussian, but instead a broader tailed distribution, we can replace the quadratic data term penalisation with the robust ℓ_1 norm. The TV$-\ell_1$ denoising model is:

$$\min_u \left\{ \int_\Omega |u-d| \, dx + \lambda \int_\Omega |\nabla u| \, dx \right\} . \tag{4.22}$$

One of the many interesting properties of this model, shown in Figure (4.8), is its ability to preserve contrast in the depth image. Increasing values of λ lead to removal of increasingly larger isolated structures in the image, while preserving the absolute values of the remaining image. This is in stark contrast to the ROF or full quadratic model which, though discontinuity preserving, results in a flattening of image values as λ increases.

A generalisation of the quadratic cost and ℓ_1 norm based models is obtained using the Huber norm, Equation (3.47), on both the regularisation and data term energy:

$$\min_u \left\{ \int_\Omega |u-d|_\delta \, dx + \lambda \int_\Omega |\nabla u|_\gamma \, dx \right\} . \tag{4.23}$$

When used in depth map denoising, on the data term the Huber norm better models depth measurement error obtained using the local stereo methods as a corrupted Gaussian. More striking is the effect of the small quadratic region in the Huber function when used in the regularisation cost, resulting in removal of the severe staircase effect that results when using the TV regularisation illustrated in Figure (4.9).

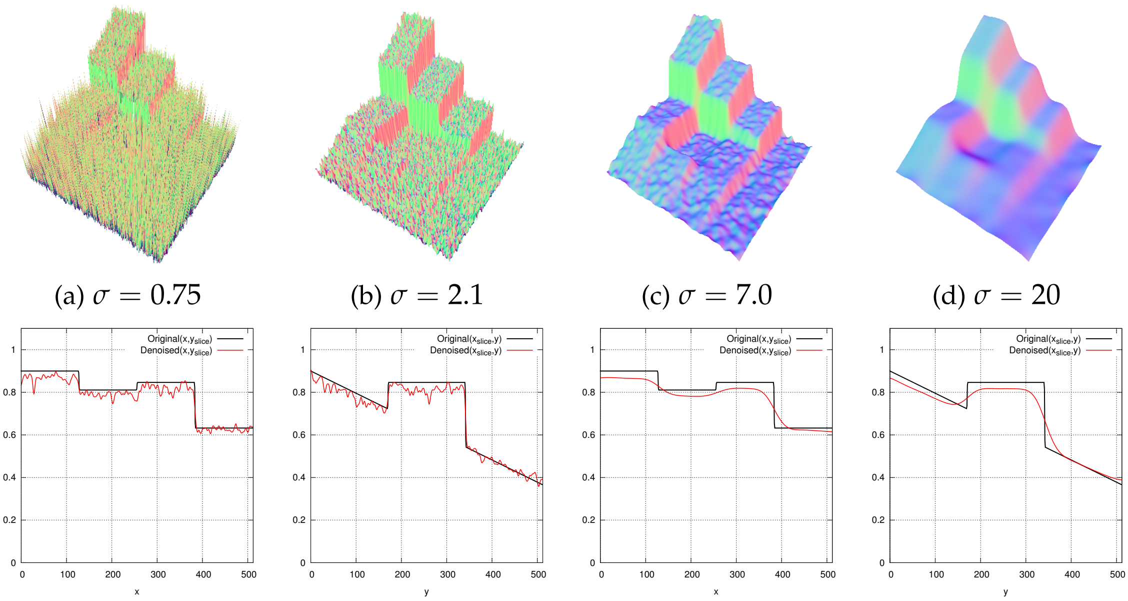
4.5.2 Total Generalised Variation

As introduced by **Bredies et al.** (2010), total generalised variation (TGV) regularisation, enables piecewise polynomial function reconstruction of any degree in contrast to the piecewise constant function reconstruction possible with the TV regularisation.

Pock et al. (2011) made effective use of the TGV to create a state of the art multiple depth map denoising, or depth map fusion, algorithm that we will return to later in this section. They demonstrate the power of the second order variant of the Total-generalised variation regularisation in conjunction with a denoising data term using a Huber cost function. The single depth input version of their TGV$_\alpha^2$-Huber denoising model is:

$$\min_u \left\{ \int_\Omega |u-d|_\delta \, dx + \alpha_1 \int_\Omega |\nabla u - v| \, dx + \alpha_0 \int_\Omega |\mathcal{E} v| \, dx \right\} \tag{4.24}$$

where $\alpha_1 = 2\alpha_0$ acts on the ℓ_1 norm of the symmetrised gradient operator $\mathcal{E}$ over a second variable v. The remarkable result of the second order TGV model is the ability to auto-



(a) $\sigma = 0.75$

(b) $\sigma = 2.1$

(c) $\sigma = 7.0$

(d) $\sigma = 20$

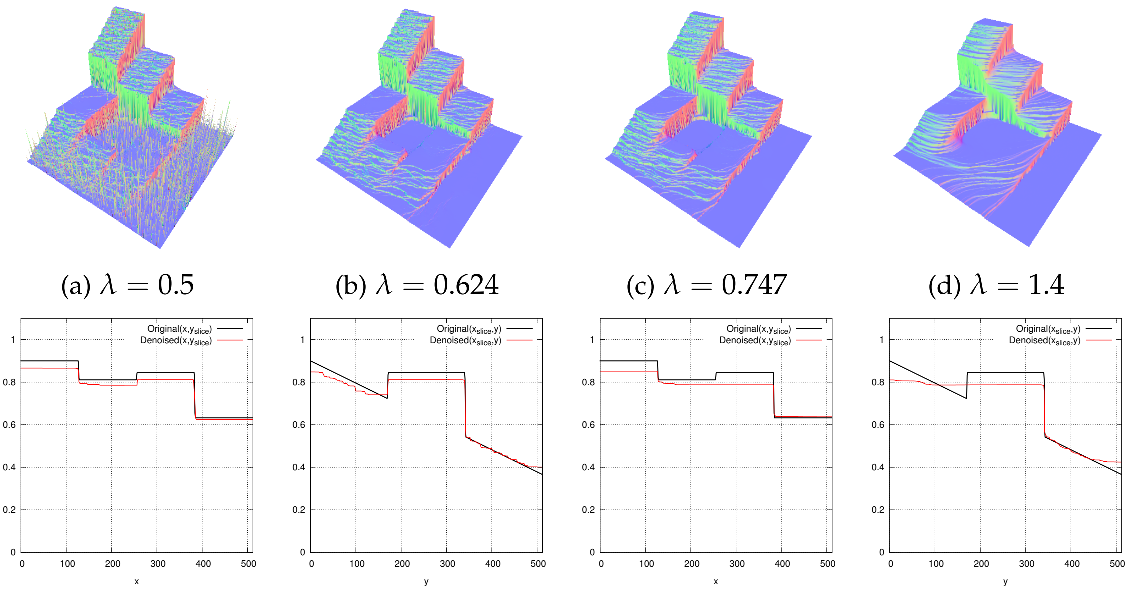
(e) Slice through x, $\sigma = 2.1$

(f) Slice through y, $\sigma = 2.1$

(g) Slice through x, $\sigma = 20$

(h) Slice through y, $\sigma = 20$

Figure 4.6: Gaussian Convolution



(a) $\lambda = 0.5$

(b) $\lambda = 0.624$

(c) $\lambda = 0.747$

(d) $\lambda = 1.4$

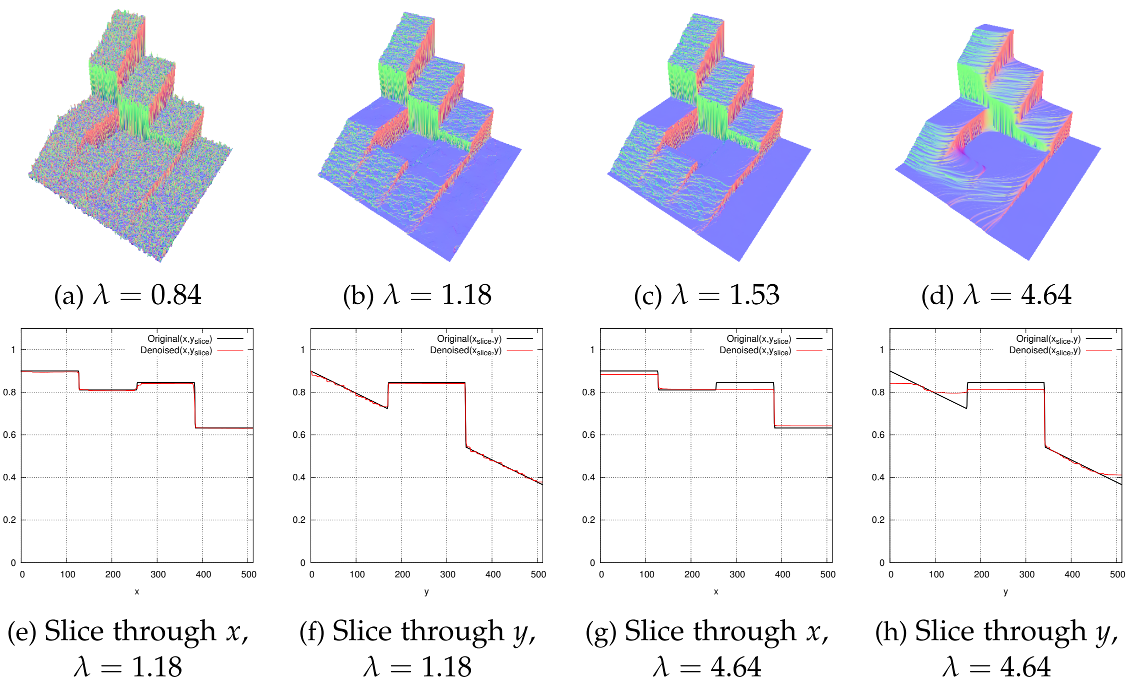
(e) Slice through x, $\lambda = 0.624$

(f) Slice through y, $\lambda = 0.624$

(g) Slice through x, $\lambda = 1.4$

(h) Slice through y, $\lambda = 1.4$

Figure 4.7: TV-ℓ_2^2



(a) $\lambda = 0.84$

(b) $\lambda = 1.18$

(c) $\lambda = 1.53$

(d) $\lambda = 4.64$

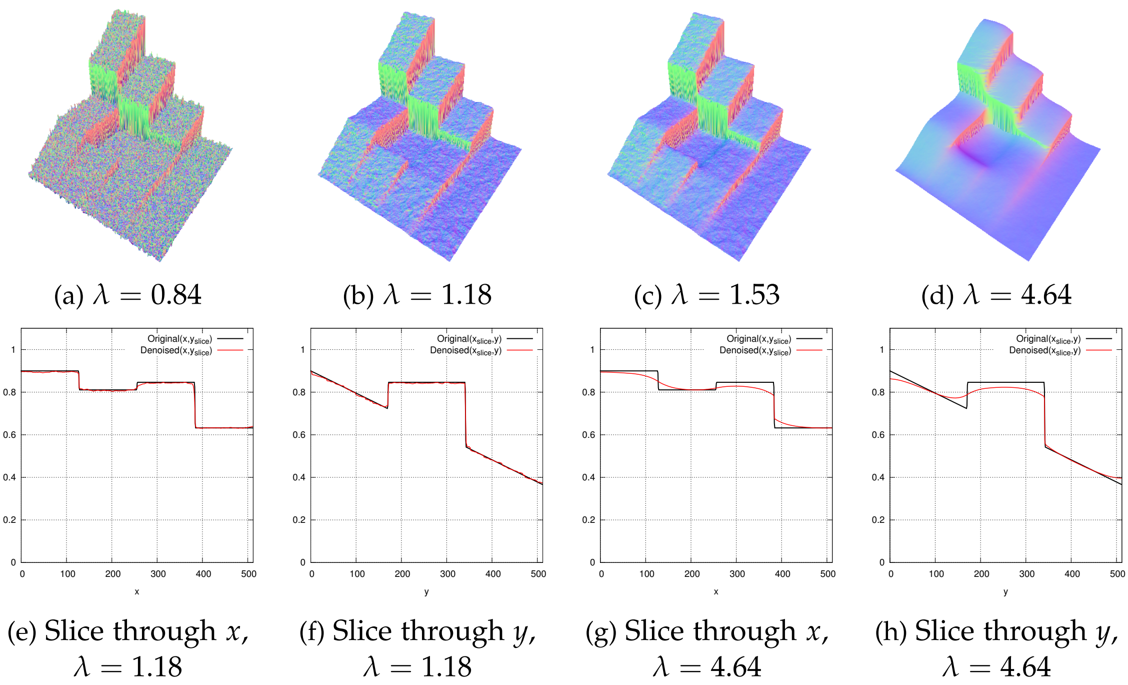
(e) Slice through x, $\lambda = 1.18$

(f) Slice through y, $\lambda = 1.18$

(g) Slice through x, $\lambda = 4.64$

(h) Slice through y, $\lambda = 4.64$

Figure 4.8: TV-ℓ_1



(a) $\lambda = 0.84$

(b) $\lambda = 1.18$

(c) $\lambda = 1.53$

(d) $\lambda = 4.64$

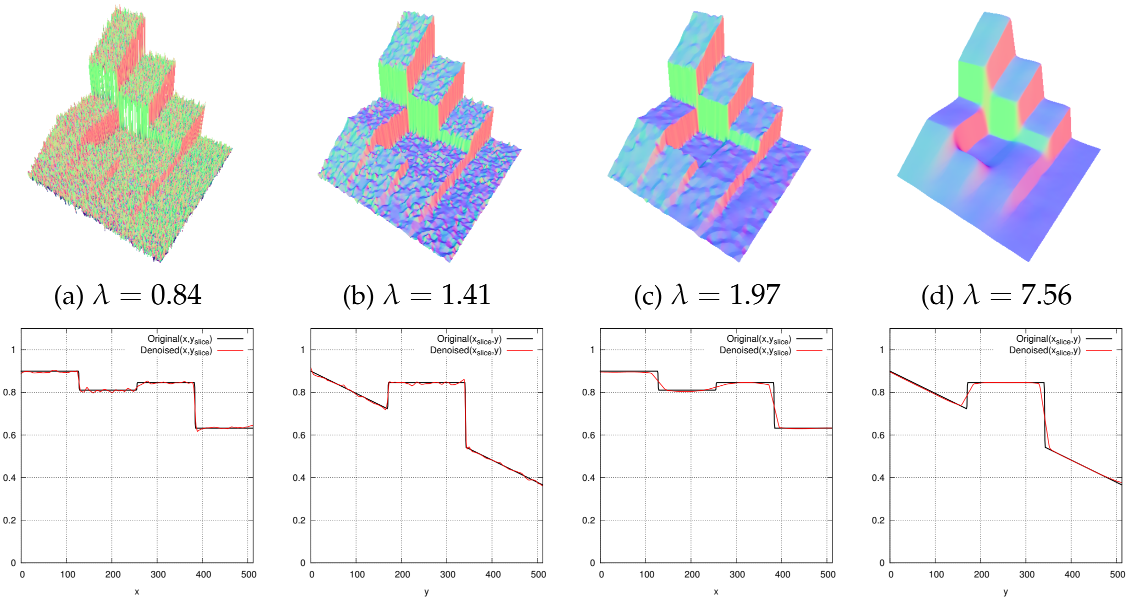
(e) Slice through x, $\lambda = 1.18$

(f) Slice through y, $\lambda = 1.18$

(g) Slice through x, $\lambda = 4.64$

(h) Slice through y, $\lambda = 4.64$

Figure 4.9: Huber-ℓ_1, with the Huber parameter $\gamma = 0.00147$



(a) $\lambda = 0.84$ (b) $\lambda = 1.41$ (c) $\lambda = 1.97$ (d) $\lambda = 7.56$

(e) Slice through x, $\lambda = 1.41$ (f) Slice through y, $\lambda = 1.41$ (g) Slice through x, $\lambda = 7.56$ (h) Slice through y, $\lambda = 7.56$

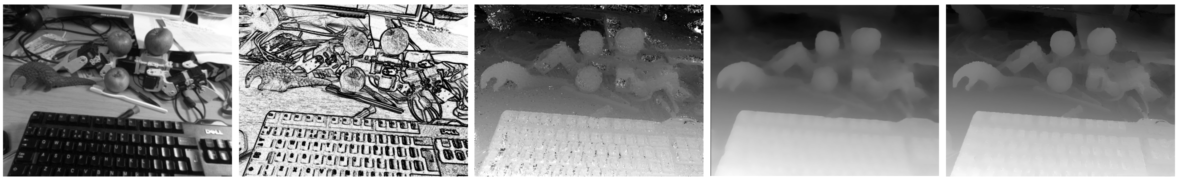
Figure 4.10: $\mathrm{TGV}_\alpha^2 - \ell_1$

matically balance the first and second derivatives on a per point basis. This is in contrast to global optimisation models which explicitly weight the varying degrees of smoothness manually on a whole image basis. We note that it is trivial to append the image driven weighting to the first order smoothness error term but in practice we have found that the TGV_α^2 regularisation provides superior performance without it, the model is demonstrated in Figure (4.10).

4.5.3 Inhomogeneous isotropic diffusion

As noted in Section (4.3.1), since we are denoising a depth map computed in a reference frame with an associated image $\mathcal{I}_r$, it is reasonable to assume that some image boundaries might correspond to depth discontinuities. By increasing the regularisation strength in regions which are expected to be smooth and reducing the strength at possible boundaries, known as image driven regularisation, we can effectively produce discontinuity preservation in the solution with similar performance to a non-convex smoothness term.

Discussed in Section (2.3.2) the majority of such image driven regularisation approaches used in both optical flow and variational depth estimation formulations used anisoptropic diffusion ((**Nagel and Enkelmann**, 1986)) where the regularisation is directionally weighted using a function of the image gradient. **Bresson et al.** (2005) demonstrated the effectiveness of a simple scalar weighted-TV regularisation, which has been used to great



(a) Example reference image, $\mathcal{I}_r$

(b) Weighting function, g.

(c) Noisy data term, d

(d) Denoised solution u using $|\nabla|_\gamma$.

(e) Denoised solution u using $g|\nabla|_\gamma$.

Figure 4.11: Example depth map denoising using the gHuber-ℓ_1 denoising model. Weighting function (b) computed for a reference image (a), here $\alpha = 100, \beta = 1.6$. The weighting function reduces the regularisation energy at strong image boundaries. The result of denoising the multi-view data term minimum (noisy depth map) from (c) shown without weighting of the regularisation term (d). Keeping optimisation and model parameters constant, but using the weighted regularisation results in more accurate reconstruction at depth discontinuities (e).

effect in high accuracy variational optical flow (**Werlberger**, 2012). Adding a per point weighting into the regularisation term of the previous model produces the weighted-Huber denoising model:

$$\min_u \left\{ \int_\Omega |u - d|_\delta \, dx + \lambda \int_\Omega g |\nabla u|_\gamma \, dx \right\} , \tag{4.25}$$

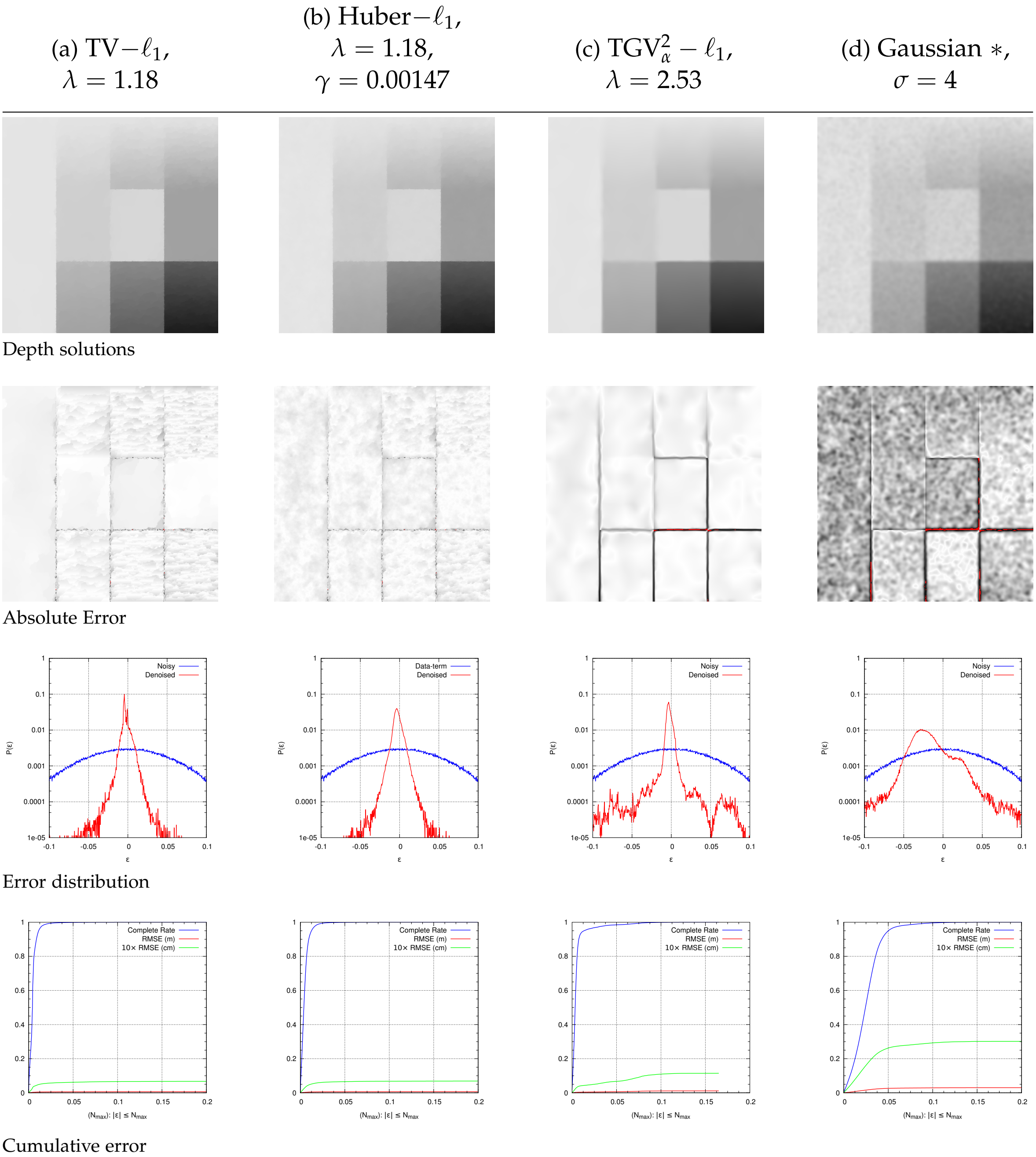
where the isotropic weighting function $g(x)$ is computed at each pixel by:

$$g(x) \equiv \exp^{-\alpha |\nabla \mathcal{I}_r(x)|^\beta} . \tag{4.26}$$

Such *inhomogeneous isotropic diffusion* enables stronger regularisation of the depth map, which typically would result in the removal of larger image structures. Since closed homogeneous regions within g will have corresponding boundaries with small g values the regularisation energy is reduced on the boundary allowing discontinuities to form with reduced cost to the global energy, illustrated in Figure (4.11).

4.5.4 Primal-Dual Formulations for Denoising

The above primal formulations of the denoising models contain Huber penalisation based cost functions or ℓ_1 norms which are not continuously differentiable. We will now provide the discrete primal-dual formulation of the models following the approach developed in **Chambolle and Pock** (2011). The primal-dual formulations enable trivially parallelisable gradient descent optimisation schemes to be used to minimise the global energies.



(a) TV$-\ell_1$, $\lambda = 1.18$

(b) Huber$-\ell_1$, $\lambda = 1.18$, $\gamma = 0.00147$

(c) $\text{TGV}^2_\alpha - \ell_1$, $\lambda = 2.53$

(d) Gaussian $*$, $\sigma = 4$

Depth solutions

Absolute Error

Error distribution

Cumulative error

Figure 4.12: Synthetic experiment solutions and errors. Each column shows a resulting solution and related error for the denoising technique specified at the top of the column. Rows from top to bottom show the solution depth map; image difference image; signed error ϵ histogram for the range $\pm 0.1m$ where the solution error distribution (red) is plotted together with the original noisy input error distribution. In the final row we generate the RMSE and image completion (fill) plots obtained for the solution pixels that have absolute error, $|\epsilon|$, within the specified magnitude: N_{max}. We compute the ratio of pixels remaining in the solution with $|\epsilon|$ below N_{max}, together with the solution RMSE. In this experiment we use the model parameters optimised to obtain the minimum RMSE for $N_{max} = 0.1$ at which point all models compared reach a completion rate of over 99%.

Primal-Dual *Weighted-Huber* Denoising

We point out that several basic but interesting variations of the denoising model are generalised by Equation (4.25). In particular we abbreviate the weighted TV-ℓ_1 case using inhomogeneous isotropic weighting and setting $(\delta, \gamma = 0)$ as gTV-ℓ_1. We also label the model instance gHuber-ℓ_1 when using the weighted Huber penalisation where $\gamma > 0$.

Following the approach outlined in Section (3.4), we transform each term in Equation (4.25) using the Legendre-Fenchel transform we arrive at a saddle-point problem:

$$\begin{aligned} \min_{d} \max_{p,r} \quad & \left\{ \langle r, u-d \rangle + \langle p, \nabla d \rangle - \tfrac{\delta}{2}\|p\|_2^2 - \tfrac{\gamma}{2}\|r\|_2^2 \right\} \\ \text{subject to} \quad & \|r\|_\infty \leq 1 \text{ , } \|p_{i,j}\| \leq g_{i,j}\lambda \text{ ,} \end{aligned} \tag{4.27}$$

We obtain the solution u by performing a gradient ascent on the dual variables and a gradient descent on the primal variables. The gradient-ascent update for the dual terms q, p is:

$$p_{i,j}^{n+1} = \Pi_{g_{i,j}} \left(\frac{p_{i,j}^n + \sigma \nabla u_{i,j}^n}{1 + \sigma\delta} \right) \text{ ,} \tag{4.28}$$

$$r_{i,j}^{n+1} = \Pi_{\lambda} \left(\frac{r_{i,j}^n + \sigma(u_{i,j}^n - d_{i,j})}{1 + \sigma\gamma} \right) \tag{4.29}$$

where the constraints on the dual variables are enforced by the projection operation given in Section (3.4). We note that the g-weighting term is assembled into the dual model by scaling the radius of the variable projection.

Fixing the dual variable, we then update the primal variable u^n using gradient descent:

$$u_{i,j}^{n+1} = u_{i,j}^n + \tau(\nabla \cdot p_{i,j}^{n+1} - \lambda r_{i,j}^{n+1})) \tag{4.30}$$

Primal-Dual TGV$_\alpha^2$-Huber Denoising

The primal-dual model for the TGV$_\alpha^2$-Huber denoising is similarly obtained, using the Legendre-Fenchel transform on all terms in Equation (4.24):

$$\begin{aligned} \min_{u,v} \max_{p,q,r} \quad & \Big\{ \langle \nabla u - v, p \rangle + \langle \mathcal{E} v, q \rangle + \langle u - d, r \rangle \\ & - \left(\delta_{\{|p| \leq \alpha_1\}} + \delta_{\{|q| \leq \alpha_0\}} + \delta_{\{|r| \leq 1\}} + \tfrac{\delta}{2}\|r\|_2^2 \right) \Big\} \\ \text{subject to} \quad & \|p\|_\infty \leq \alpha_1 \text{ , } \|q\|_\infty \leq \alpha_0 \text{ , } \|r\|_\infty \leq 1 \text{ .} \end{aligned} \tag{4.31}$$

Solution proceeds iteratively, first fixing the primal variables, gradient ascent on the dual variables is performed:

$$(p_{i,j})^{n+1} = \Pi_{\alpha_1}\Big((p_{i,j})^n + \sigma(\nabla(u^h_{i,j})^n - (v^h_{i,j})^n)\Big) \tag{4.32}$$

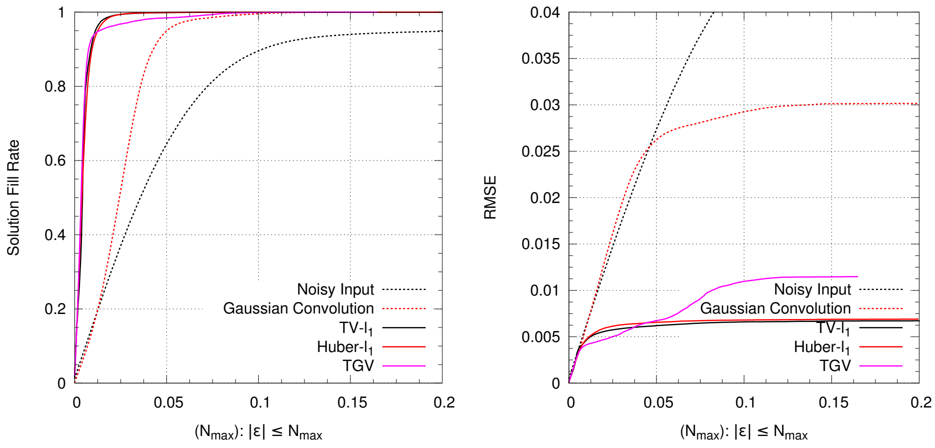
$$(q^h_{i,j})^{n+1} = \Pi_{\alpha_0}\Big((q^h_{i,j})^n + \sigma(\mathcal{E}(v^h_{i,j})^n)\Big) \tag{4.33}$$

$$(r^h_{i,j})^{n+1} = \Pi_1\left(\frac{(r^h_{i,j})^n + \sigma((u_{i,j})^n - (d^h_{i,j}))}{1+\sigma\delta}\right) . \tag{4.34}$$

Fixing the Dual variables, gradient descent on the primal variables is given by:

$$(u^h_{i,j})^{n+1} = \Big((u^h_{i,j})^n + \tau(\nabla\cdot(p_{i,j})^{n+1} - (r^h_{i,j})^{n+1})\Big) \tag{4.35}$$

$$(v^h_{i,j})^{n+1} = \Big((v^h_{i,j})^n + \tau(\mathcal{E}^T(q^h_{i,j})^{n+1} + (p^h_{i,j})^{n+1})\Big) . \tag{4.36}$$



(a) Solution fill rates.

(b) RMSE.

Figure 4.13: Performance analysis of depth map denoising algorithms on the synthetic dataset shown in Figure (4.5). We generate the RMSE and image completion (fill) plots obtained for the solution pixels that have absolute error to ground truth within the specified magnitude: N_{max}. The fill rate is computed as the ratio of pixels in the solution with $|\epsilon|$ below N_{max} over the number of pixels in the reference image observed by at one other view. The RMSE error is also computed using the average over all estimated pixels within the absolute error bound. In all experiments shown we use the model parameters optimised to obtain the minimum RMSE for $N_{max} = 0.1$. At this bound all denoised solutions reach a fill rate of over 99%.

4.5.5 Synhetic Depth Map Denoising Comparison

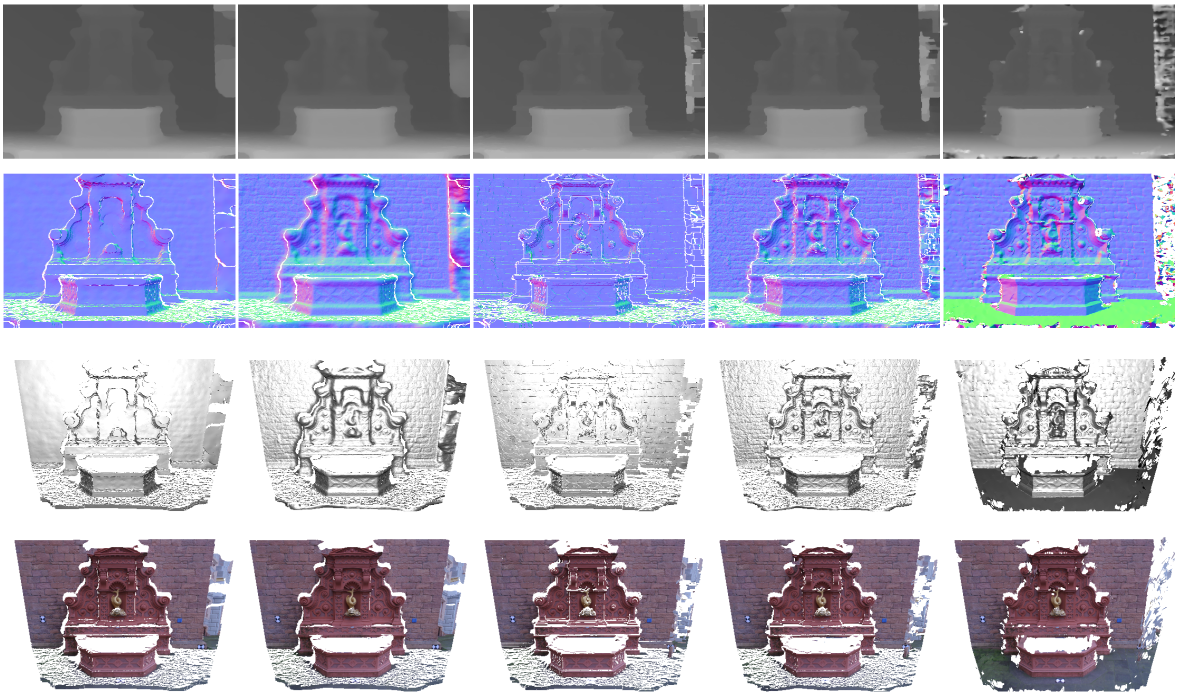
In Figure (4.12), and summarised in Figure (4.13) we compare the denoising techniques. Solutions are obtained by running optimisation for each model until a solution change of less than $1e-3m$. We set for each model the parameters that minimise the RMSE for a set threshold of the solution at $N_{max} = 0.1m$ absolute error. At this setting all model solutions (excluding Gaussian convolution) reach at least a 99% solution fill rate. It is notable that all solutions attain a modal error that is non zero; this does not occur when the depth map is corruptive with either additive Gaussian noise or the salt and pepper noise alone.

The characteristic piecewise constant bias of the TV solution seen in column (4.12)(a) results in multiple high modes in the error distribution induced by errors at the slopping points in the ground truth image. However, while the Huber penalty smooths the solution, it does so at the cost increased error for the fronto parallel regions. The TGV solution results in reduced error over all surfaces but suffers from increased errors at the depth discontinuities, this can be seen clearly in the cumulative error plots where TGV produces the lowest RMSE and up to $N_{max} < 0.0075m$ shown in Figure (4.13)(b).

4.5.6 MVS Depth Map Denoising Comparison

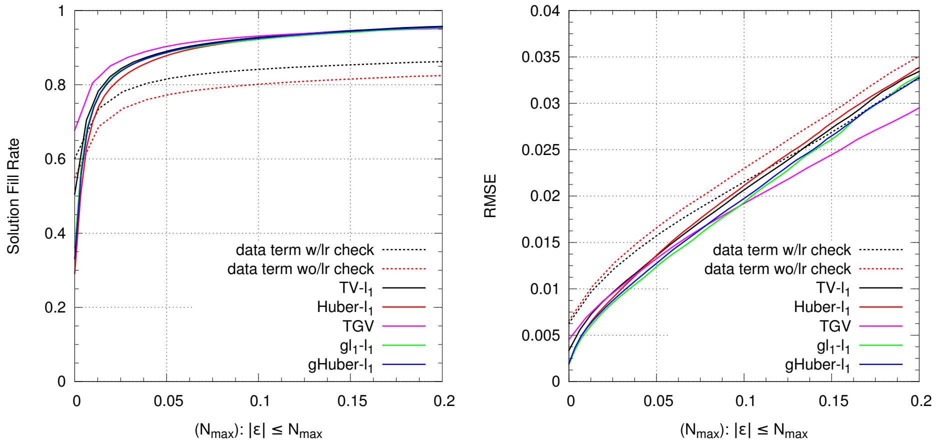
To analyse the relative performance of the depth map denoising techniques introduced in this section we utilise the ground truth data depth map from the fountain-P11 multiple view stereo sequence by **Strecha et al.** (2008), previously shown in Figure (4.1), which provides calibrated and rectified image input. First, to give a sense of the solution quality and importantly the space of typical solutions for the above models we compare the models qualitatively in Figure (4.14). We first sub sample the image data from 3072×2048 to 480×320 which coincides with attaining real-time (20-30fps) performance using our GPGPU implementation taking into account computation of the depth map data term and running the convex optimisation routines to convergence when the solution changes between iterations is less than $1e-3$. Since the optimisation problem is convex, the solution is not changed by the initialising solution which we set as the data term. This simple analysis is useful for accessing potential performance of the models in a live dense SLAM system. We provide as input to all depth map denoising models the same input, shown in Figure (4.2b) with details on the settings provided at the end of Subsection (4.4.3).

Using depth map reference frame 5 as the ground truth from the fountain-P11 sequence we perform a quantitative comparison of the techniques, detailing comparative solution slices, absolute and signed error images, error distributions and cumulative error and solution fill rate plots detailed in Figures (4.16-4.20) and summarised in Figure (4.15). All of the solutions clearly denoise the noisy depth map data term reducing the RMSE, drastically culling the broad tail errors present in the input depth maps. As seen in the synthetic de-noising examples the Huber based penalty terms rectify the constant region bias that results in reduced reconstruction fidelity for non fronto parallel surfaces, although under



(a) TV-ℓ_1 (b) Huber-ℓ_1 (c) gTV-ℓ_1 (d) gHuber-ℓ_1 (e) $\mathrm{TGV}_\alpha^2 - \ell_1$

Figure 4.14: A summary qualitative comparison of single image, convex depth map denoising models, highlighting the prominent differences produced by each model. All models use the five view occlusion robust depth map computed from the data-term minimum, illustrated in Figure (4.2b). Model parameters are not tuned for performance against the ground truth depth map, but are instead set based on achieving the best performance within the complete dense visual SLAM system, described in Chapter (7). For each model result we show (top to bottom rows) the denoised depth map, the normal map rendering in the image plane and Phong shaded mesh rendering shown tilted away from the image plane, and finally the textured mesh. Mesh vertices are culled using a visibility threshold to illustrate discontinuities in the depth map (the threshold is constant across results). The distinguishing differences between each model are clearly shown: without any image driven regularisation (a) and (b) demonstrate extremes of the Huber Model, from the ℓ_1 based TV regularisation in (a) illustrating the piece-wise constant solution bias, while (b) shows the smoothing effect of the small quadratic component in the Huber penalisation term. Image driven regularisation results in improvements for both regularisation settings, increasing detail preservation while still suppressing noise in the data term. Finally, in comparison with the first order smoothness terms used in models (a-d), the power of TGV_α^2 regularisation is clearly demonstrated, showing smooth planar reconstruction of the ground.

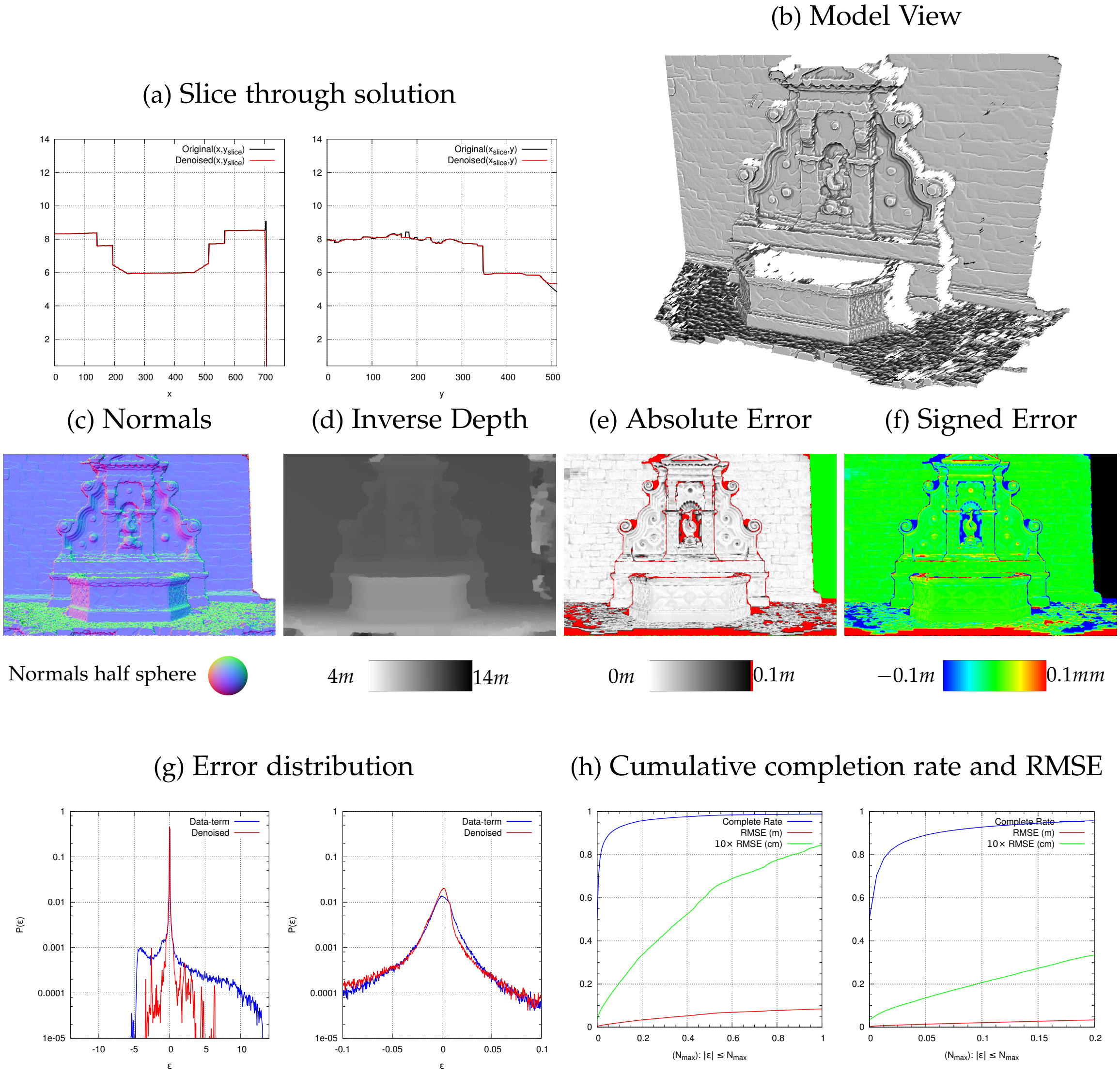


(a) Solution fill rates

(b) RMSE

Figure 4.15: Performance analysis of depth map denoising algorithms for reference depth map 5 from the fountain-P11 dataset. We generate the RMSE and image completion (fill) plots obtained for the solution pixels that have absolute error to ground truth within the specified magnitude: N_{max}. The fill rate is computed as the ratio of pixels in the solution with $|\epsilon|$ below N_{max} over the number of pixels in the reference image observed by at one other view. The RMSE error is also computed using the average over all estimated pixels within the absolute error bound. In all experiments shown we use the model parameters optimised to obtain the minimum RMSE for $N_{max} = 0.1$. At this bound all denoised solutions reach a fill rate of over 90%.

the RMSE metric, although the results are perceptually different the resulting change in reconstruction error is near negligible on this dataset. We also find that while the TGV solution produces the most complete solutions at low error thresholds as seen in (4.15), that it generally produces less accurate reconstructions in practice with errors resolving at depth discontinuities. This is possibly due to the higher order smoothness term trading off immediate jumps at such boundaries possible with TV against 2^{nd} order smoothness that can further reduce error in larger planar regions but that has lower bandwidth reconstruction capabilities due to the increased derivative filter size.



(a) Slice through solution

(b) Model View

(c) Normals

(d) Inverse Depth

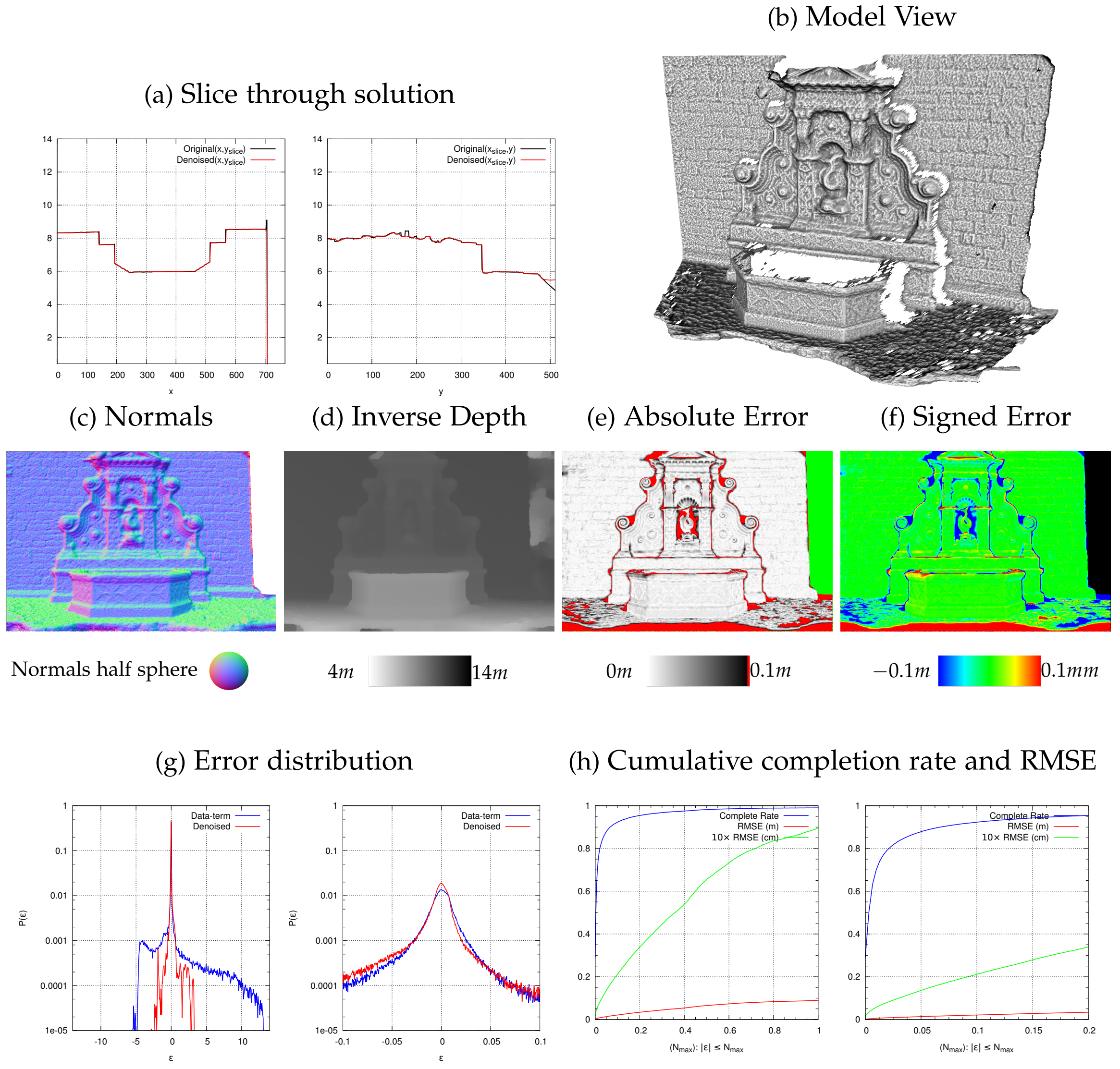
(e) Absolute Error

(f) Signed Error

(g) Error distribution

(h) Cumulative completion rate and RMSE

Figure 4.16: TV-ℓ_1 Solution, $\lambda = 2$. The error image (e) uses a grey scale to encoded absolute error to the ground truth depth at each pixel up to $0.1m$, is red for solution points with $> 0.1m$ absolute error. Green pixels encode that have no ground truth depth. The signed error is rendered in (f) with saturation at $\pm 0.1m$ error. We plot the error distribution (g) for the solution depth map (red) together with the distribution of the depth map dataterm used (blue). In (h) we compute the RMSE and image completion (solution fill) ratio given a specified thresholding, N_{max}, on the absolute error, $|\epsilon|$, in the depth map.



(a) Slice through solution

(b) Model View

(c) Normals

(d) Inverse Depth

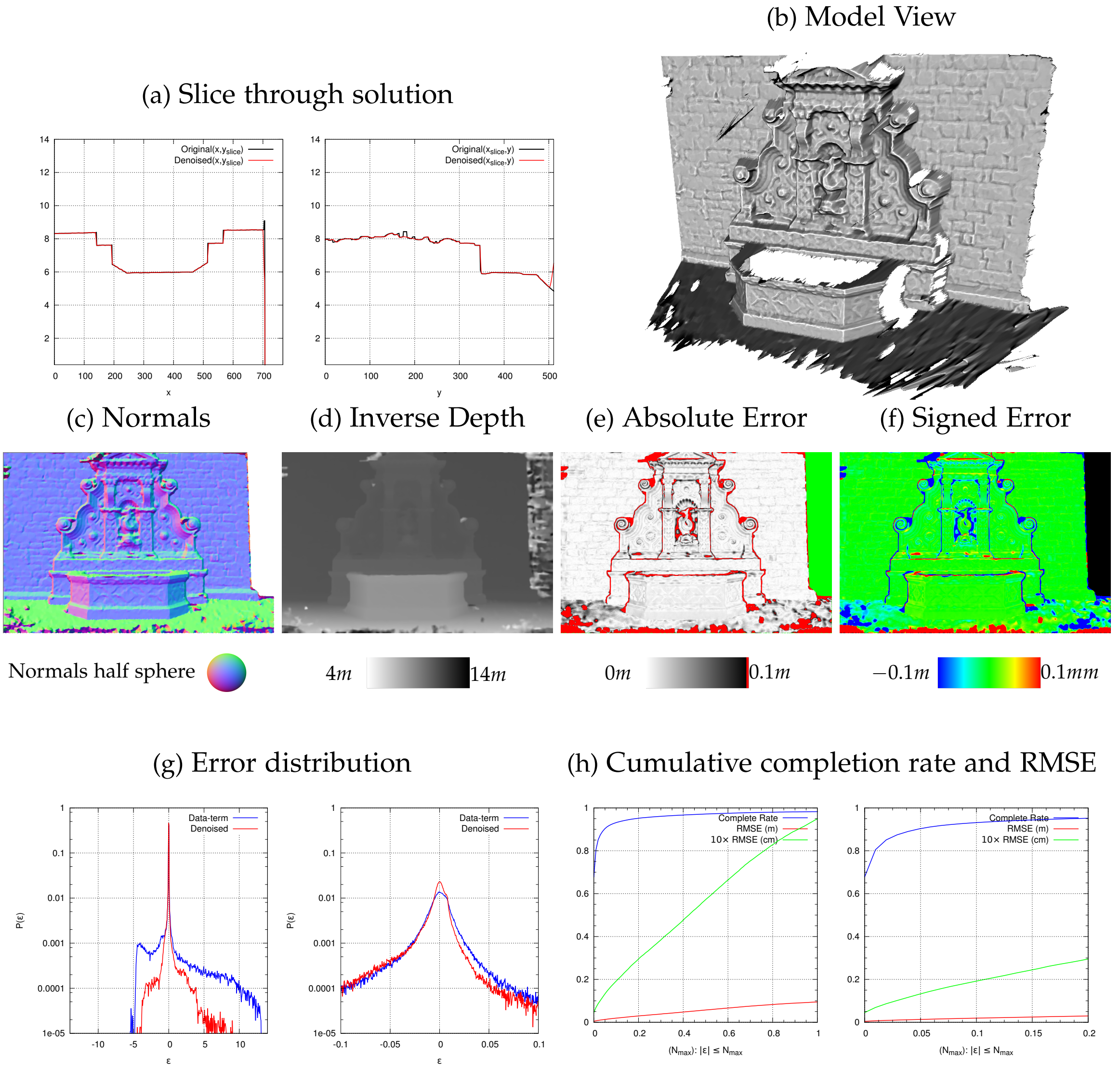
(e) Absolute Error

(f) Signed Error

(g) Error distribution

(h) Cumulative completion rate and RMSE

Figure 4.17: Huber-ℓ_1 Solution, $\lambda = 2$, Huber $\gamma = 0.00159$.



(a) Slice through solution

(b) Model View

(c) Normals

(d) Inverse Depth

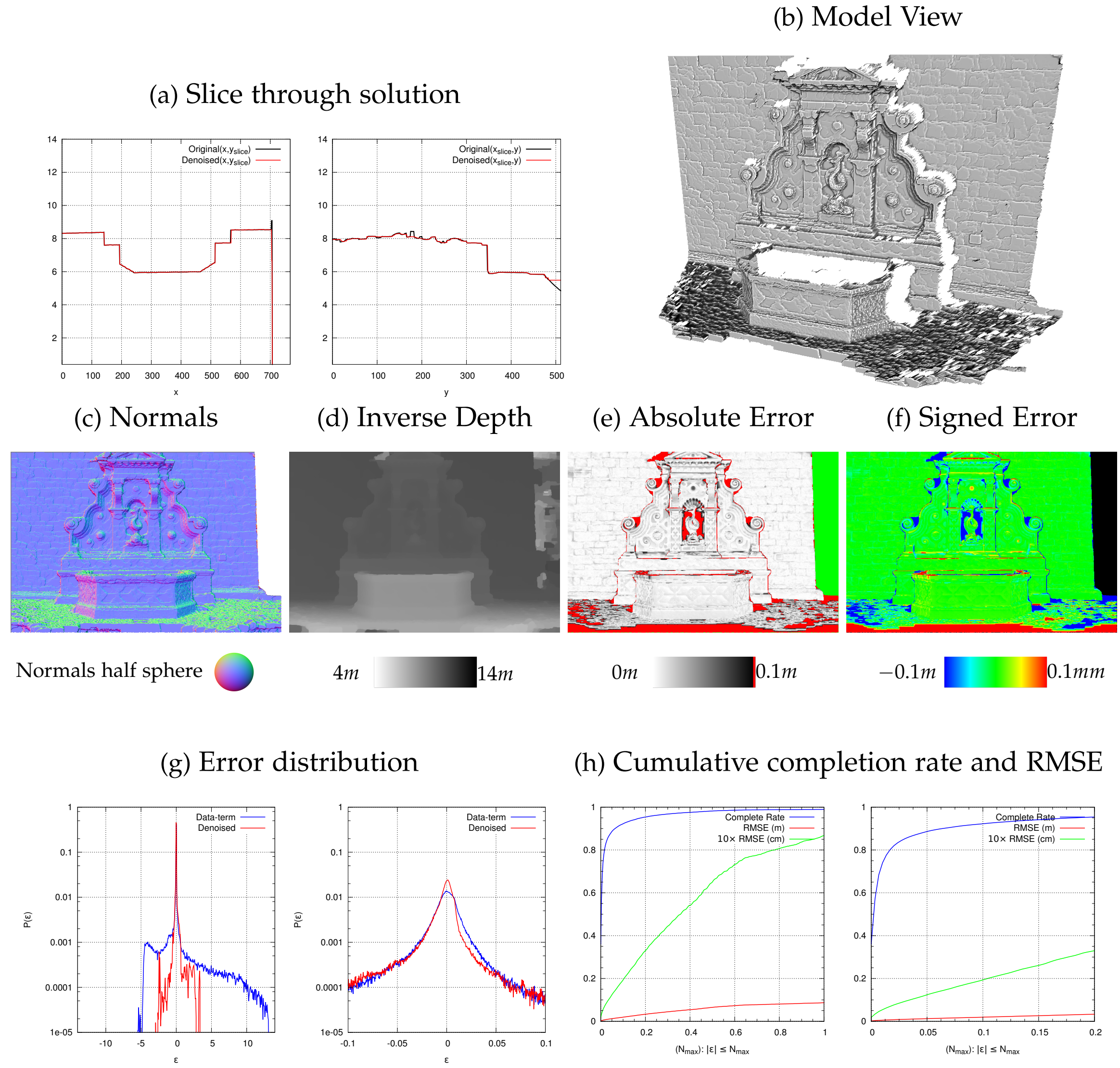
(e) Absolute Error

(f) Signed Error

(g) Error distribution

(h) Cumulative completion rate and RMSE

Figure 4.18: TGV$^2_\alpha - \ell_1$ Solution, $\lambda = 1.667$.



(b) Model View

(a) Slice through solution

(c) Normals

(d) Inverse Depth

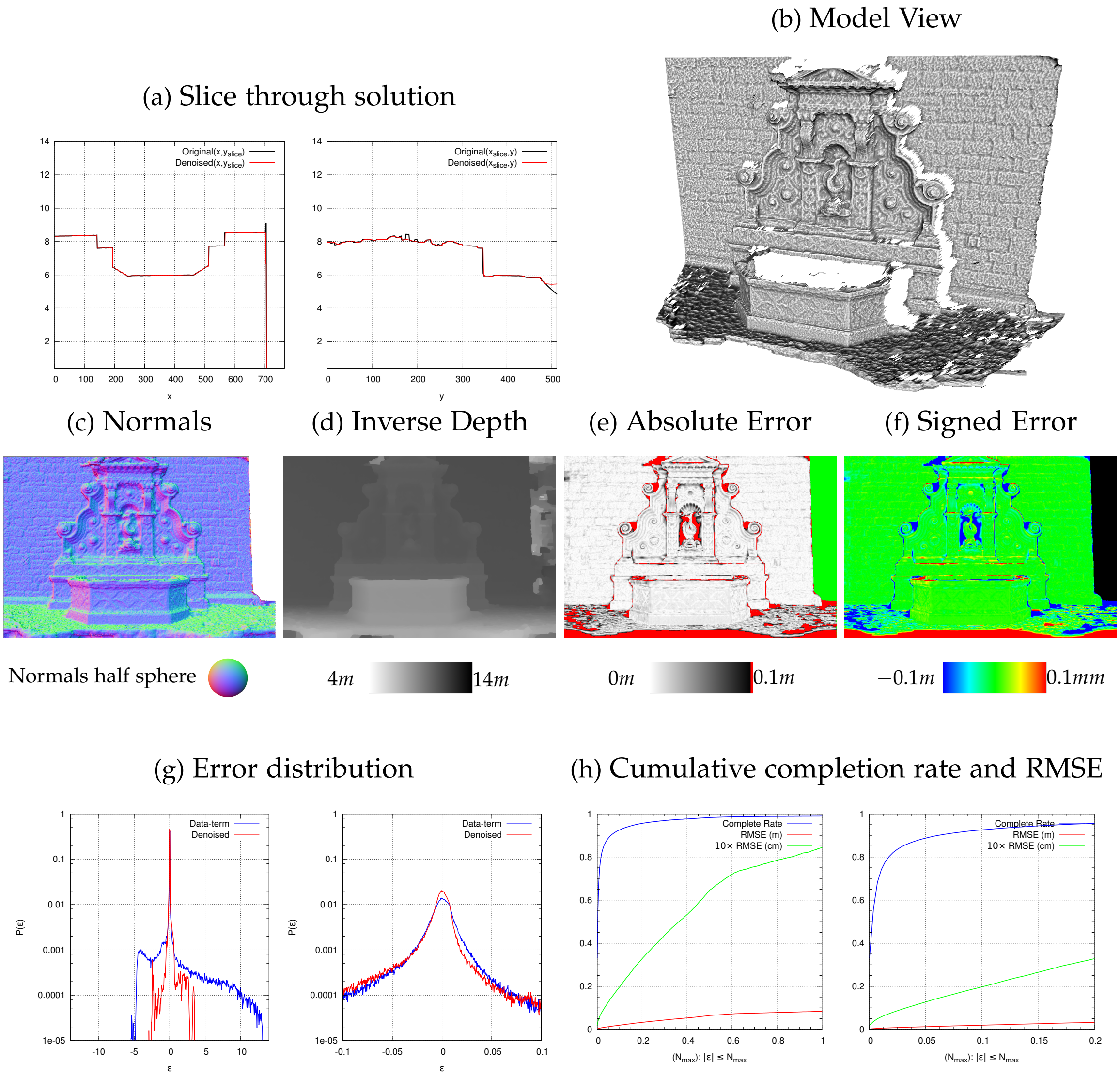
(e) Absolute Error

(f) Signed Error

(g) Error distribution

(h) Cumulative completion rate and RMSE

Figure 4.19: gTV-ℓ_1 Solution, $\lambda = 2$ with image driven regularisation $\alpha = 10$, $\beta = 1$.



(a) Slice through solution

(b) Model View

(c) Normals

(d) Inverse Depth

(e) Absolute Error

(f) Signed Error

(g) Error distribution

(h) Cumulative completion rate and RMSE

Figure 4.20: gHuber-ℓ_1 Solution, $\lambda = 2$, Huber $\gamma = 0.00159$, and image driven regularisation $\alpha = 10$, $\beta = 1$.

4.5.7 $2.5D$ Depth Map Fusion

In this subsection we look at the $2.5D$ depth map fusion approach, used previously by **Pock et al.** (2011) for robust estimation of a single depth map from multiple noisy overlapping depth maps. Given the supporting images captured under *perspective projected*, we must decide on a scheme to select subsets of images for computing the multiple depth map measurements, which must be in a single frame of reference. While it is possible to transform a depth map from one reference frame into another, the simplest mechanism is to use a common reference frame r, and then compute depth maps from subsets of neighbouring frames into the common reference frame, computing N depth map observations using pairs $(r, k \in 1..N)$:

$$d_k^{min}(x) = \underset{d \in \mathcal{M}}{\operatorname{argmin}}\, \rho(x, k, d) \ . \tag{4.37}$$

An obvious advantage over denoising the single summed data term minimum is that occlusions can now be correctly treated as outliers over all pairwise observations using a robust norm. The denoising energy extended to multiple depth map observations is:

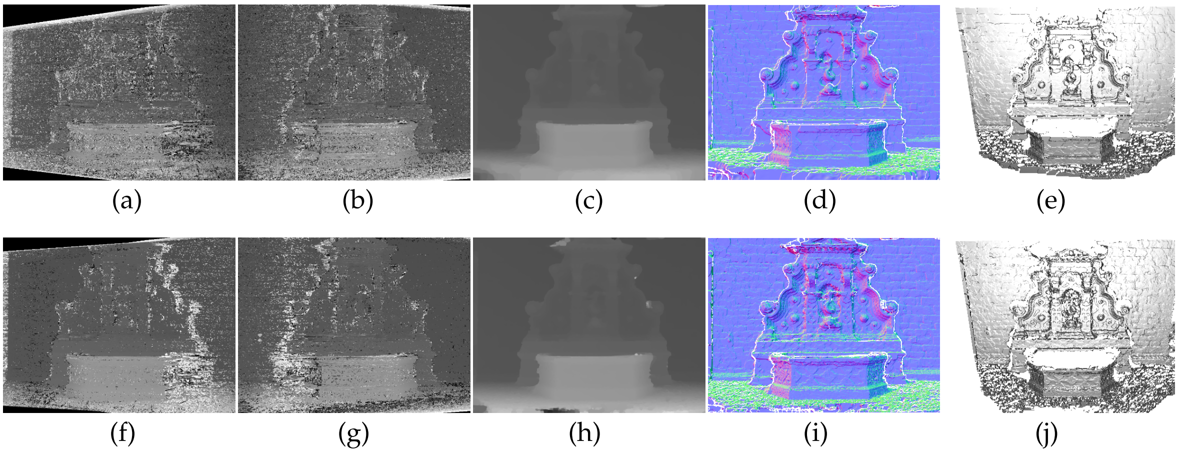
$$\min_u \left\{ \int_\Omega \sum_{k \in \mathcal{I}_r} \psi_{\mathcal{D}}(u - d_k) + \int_\Omega \lambda \psi_{\mathcal{R}}(\nabla u) \right\} \ , \tag{4.38}$$

where the summation over data-terms discards invalid pixels in d_k where no valid depth can be estimated. This occurs when all projections from the discretised depth along the reference ray projects outside of the image bounds for frame k.

The primal-dual formulation for each of the multiple depth map denoising schemes is obtained by introducing a dual variable for each of the depth map error terms. The primal-dual model for the multiple image extension to the weighted-Huber denoising model in Equation (4.25) is:

$$\begin{array}{ll} \min\limits_u \max\limits_{p,r} & \left\{ \lambda \sum_{k=1}^n \langle r_k, u - d_k \rangle + \langle p, \nabla u \rangle - \frac{\epsilon}{2}\|p\|^2 - \frac{\gamma}{2}\|r\|^2 \right\} \\ \text{subject to} & \|r\|_\infty \leq \lambda \ , \|p\|_\infty \leq g \ . \end{array} \tag{4.39}$$

Solution proceeds by an update on the dual variable p as given in (4.28) and by a gradient


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 4.21: 2.5D depth map fusion using the multi-image gTV-ℓ_1 model. Figures (a-e) illustrate multiple-image denoising using the pairwise data term depth maps. Pair-wise depth maps are computed into the common reference frame (without occlusion handling). Two of the four depth maps are shown in (a) and (b), while the solution depth map is shown in (c) with the normal map (d) and mesh rendering (e). Figures (f-j) demonstrate the multi-image denoising using two depth map inputs from the left and right image sequences. The data term minimum depth maps computed into the common reference frame are shown in (a) and (b), with denoised result shown in (c), we illustrate surface smoothness and discontinuities through the normal map (d) and the rendered surface mesh (e).

ascent update on each dual variable r_i associated with a valid depth pixel:

$$r_k^{n+1} = \begin{cases} \Pi_\lambda\left(\frac{r_i^n+\sigma(u^n-d_k)}{1+\sigma\gamma}\right) & \text{if } k \in \mathcal{I}_r \\ r_k^n & \text{otherwise} \end{cases} \tag{4.40}$$

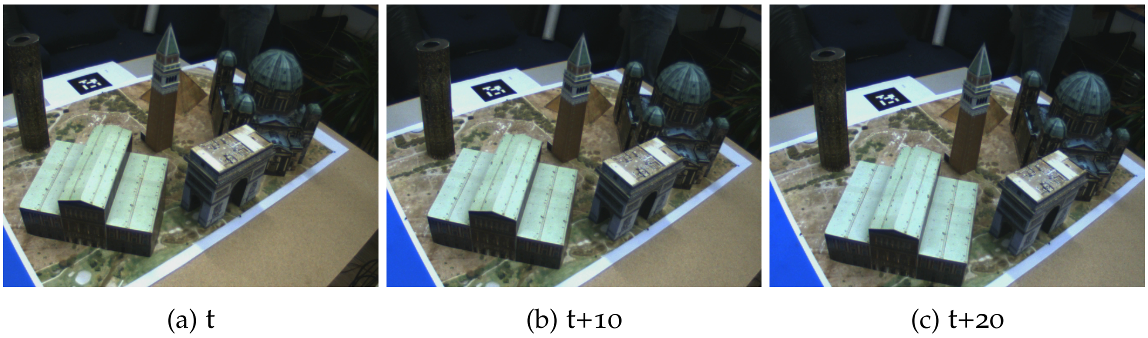
$$p^{n+1} = \Pi_g\left(\frac{p^n+\nabla u^n}{1+\sigma\epsilon}\right) \tag{4.41}$$

Fixing the dual variables, dualisation of the sum over of the data term norms leads to a summation over the dual variables in a primal gradient descent update,

$$u^{n+1} = u^n + \tau\Big(\nabla \cdot p^{n+1} - \sum_k r_k^{n+1}\Big) \,. \tag{4.42}$$

Derivation of the multiple image TGV$^2_\alpha$-Huber denoising model model originally developed by **Pock et al.** (2011) is obtained in a similar manner extending Equation (4.31).

In Figure (4.21) we qualitatively compare multiple depth map denoising using the N pair dataterms in Equation (4.37), with the result from fusing the two depth maps computed from the dataterm minimum of the left-right half sequences described in Subsection (4.4.3). This enables us to assess the capability of the multiple image denoising approach to implicitly cope with occlusions in comparison with the explicit selection of the per-pixel min-



(a) t

(b) t+10

(c) t+20

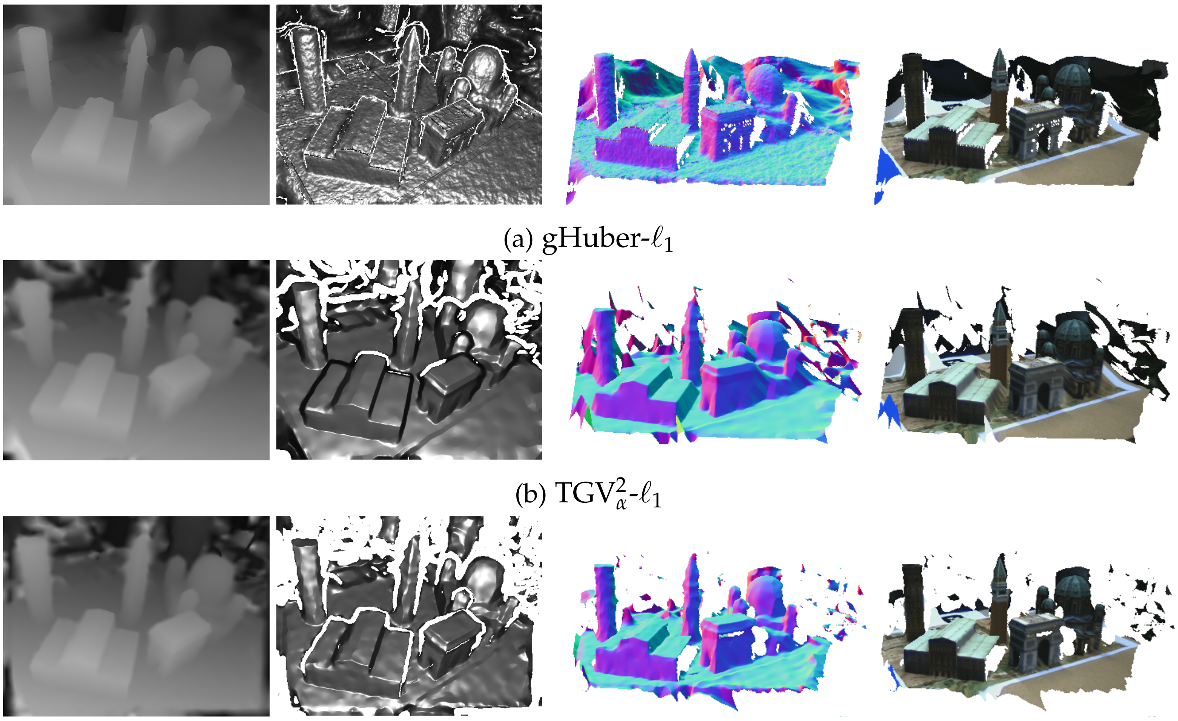
Figure 4.22: Frames from approximately 7 seconds into a video dataset of the Graz City of Sights model from **Gruber et al.** (2010). The images are captured 10 frames apart demonstrating the typical density of the video dataset in comparison to the high quality multiple view datasets.

imum in Equation (4.19).

The example clearly demonstrates the trade-off that exists with the $2.5D$ fusion approach: using more images in a single depth estimate leads to improved signal to noise ratio but accumulates errors at occlusion boundaries as the baseline of images increases. Conversely multiple depth maps from fewer frames per estimate leads to better handling of occlusion through outlier modelling but reduces the signal to noise ratio of depths for non-occluded regions.

4.5.8 Depth Map Denoising with Video Input

We now take the opportunity to briefly introduce the video rate multiple view stereo data set that will be used in a number of evaluations of the full dense reconstruction pipeline developed in Chapter (7). A key difference in the data obtained for depth estimation from a real-time moving video camera in comparison to highly utilised statically captured multiple view stereo datasets used in state of the art research (**Scharstein and Szeliski**, 2001; **Seitz et al.**, 2006; **Strecha et al.**, 2008) is the comparative density in the image data that video provides. The previous fountain-P11 data set from **Strecha et al.** (2008) provides a total of eleven very high resolution images calibrated to a level of accuracy which is not likely to be achieved in a real-time visual SLAM scenario, especially when using the lower resolution imagery required to make live computation feasible. Furthermore, we used only five down-sampled input images from the data set in total, using ± 2 neighbours of the reference frame, since increasing the the set to ± 5 tot include the total set resulted in severe degradation in the depth maps produced. The reason for this is the wide baseline nature of the sparse view set, which includes a large camera rotation component relative to our chosen reference frame, breaking the assumptions needed for the simple fronto-parallel,



(a) gHuber-ℓ_1

(b) TGV_α^2-ℓ_1

(c) TGV_α^2 2.5D fusion

Figure 4.23: A comparison of single and multiple image, convex depth map denoising models computed using the City of Sights video data and PTAM camera pose estimation. Setting the depth map reference frame to the image shown in Figure (4.22b), we use a total of 20 neighbouring frames in the multiple view stereo data term. Model parameters for the single depth image denoising models (a) and (b) remain as used in the comparison illustrated in Figure (4.14). In (c) we compute the data term minimum from the left-right half sequences for use in 2.5D fusion. Each row shows (left to right) the solution depth map, Phong shaded rendering of the resulting mesh with visibility culled vertices, and a normal map and textured mapped rendering of the resulting mesh from an alternate view.

fixed window, data term to work effectively.

In contrast to the high resolution dataset, In Figure (4.22) we show three frames from a video capture [1] of the City of Sights model developed by **Graber et al.** (2011) for evaluation of real-time augmented reality computer vision applications. Video capture was performed at $30Hz$ using a point grey flea2 at a resolution of 640×480 pixels using an 80 degree field-of-view fixed focus lens. The image data from this moving camera captured suffers from motion blur and decreased signal to noise ratio in comparison with the previous dataset. Furthermore no ground truth trajectory is provided for the camera poses.

The video browses over the City model for approximately 2.5 minutes making various

[1] We thank Gottfried Graber and Thomas Pock for providing the video of the model produced by Lukas Gruber.

loopy trajectories around and over the model. As illustrated by the frames shown in Figure (4.22), the trajectory is relatively smooth over 10 to 20 second periods providing dense, highly overlapping views of the scene. In Section (7.5) we will return to detail the experimental evaluation of real-time or live dense reconstruction systems that exploit this density in the video input. In Figure (4.23) we demonstrate three of the depth map denoising models using the video dataset. We use the real-time feature based parallel tracking and mapping system from **Klein and Murray** (2008) to estimate the relative poses of the frames, and also to provide initial structure in the scene with which to alter the minimum and maximum depth values using in the inverse depth parametrisation.

While the piecewise affine reconstruction capabilities of the TGV based model are again demonstrated in Figure (4.23b) in comparison to the gHuber-ℓ_1 model, we note that this more sophisticated model requires up to $10\times$ more iterations than the first order models to converge to useful state. This raises an important issues that arises in attempting to evaluate components of a live dense reconstruction system: given a fixed window of available processing time, a trade-off exists between attempting to compute the highest quality solution over a subset of frames and computing a lower quality solution over a larger set of frames. We will return to this in Chapter (7).

5

Convex Optimisation Based Multi-view Stereo Depth Estimation

Contents

The depth map denoising approaches described in Chapter (4) can utilise any local stereo data term, enabling gains from state-of-the-art local stereo estimation methods to be trivially incorporated. Furthermore, the denoising problem is fully convex, providing an assurance that regardless of the initial solution, the primal-dual optimisation approach will converge towards the global solution for increasing iterations. There is however *no guarantee* that the per point minimum obtained from the local stereo method is correct within some inlier noise model. If a solution exists within another local minimum or not as a local minimum at all, the best that can be achieved is for the value to be treated as an outlier. Regions with weak data terms effectively have the solution filled in by the regularisation term, which can lead to gross inaccuracies. The strength of the global optimisation approaches lies in the local data term cost being placed in the context of a globally smooth solution, enabling weak data terms to provide information where the per pixel minimum would yield an erroneous estimate.

In this chapter we develop two different multiple view stereo approaches that make full use of the data cost volume within a global optimisation framework. We first turn to variational stereo optimisation approaches which linearise the resulting non-convex functional using the full data term, previously introduced in Section (2.3.2). In section (5.1) we review the stereo techniques using modern primal-dual based convex optimisation that have developed from the earlier variational stereo methods.

In section (5.2) we introduce the specific convex optimisation models that we investigate, making use of either a single pixel or patch based data term within the image weighted gTV-ℓ_1 or Huber penalty based models or using the TGV_α^2-ℓ_1 model.

In section (5.3) we investigate an alternative global optimisation approach that replaces the linearisation of the data term with an exact search over a discretisation of the solution variable. In combination with a regularisation term, the global energy is solved using a form of annealing in which each iteration of the optimisation results in a reduction in feasible solutions for the data term.

In section (5.4) we discuss the relative merits of the convex optimisation models that we have investigated, looking at their applicability to real-time operation in a full dense reconstruction pipeline. On that basis we outline of the stereo method we will use in our dense reconstruction pipeline: combining both depth map denoising and multiple view stereo formulations. While we perform no quantitative experiments in this section, focussing instead on large scale differences in convergence and gross solution errors, we will go on to evaluate the method within a full dense reconstruction context in Chapter (7).

5.1 Modern Primal-Dual Approaches

All of the variational stereo approaches outlined in Section (2.3.2) make use of primal formulations of the continuous energy functional. The numerous models made use of robust penalties in the data term to cope with outliers in the data, together with regularisation that preserve discontinuities in the solution. However, the resulting PDE from the Euler-Lagrange equations often required further approximation to remove singularities arising at critical points in the gradient, discussed in Section (3.4).

Zach et al. (2007a); **Pock** (2008) introduced a formulation of the related optical flow energy functional, making use of a primal-dual representation of the energy which was then used with great success in a series of real-time capable optical flow implementations (see Section (3.4) for an introduction to primal-dual formulations).

Stuehmer et al. (2010) made use of the primal-dual formulation introduced in the optic flow formulation by **Zach et al.** (2007a) of the multiple-view depth map estimation problem using both an ℓ_1 penalty on a single pixel linearised brightness constancy data term and a TV regularisation of the solution. Their formulation makes use of the ability to solve a new energy functional that couples the two ℓ_1 terms together through quadratic term. By introducing the quadratic coupling term the optimisation can then proceed by alternation. The two view coupled energy is:

$$E_\theta(D) = \int_\Omega \left\{ |\nabla D|_1 + \frac{1}{2\theta}(D - v)^2 + \lambda|\epsilon(v)|_1 \right\} dx \,, \tag{5.1}$$

where θ is fixed to a small constant and the original TV-$L1$ energy is recovered as $\theta \to 0$. The first coupled term $|\nabla D|_1 + \frac{1}{2\theta}(D - v)^2$ is exactly the *ROF* model which can be solved for a fixed value v using a gradient ascent on the convex conjugate based dual formulation (Given in section 3.4.5). Then, fixing D the second coupled term $\frac{1}{2\theta}(D - v)^2 + \lambda|\epsilon(v)|_1$ presents a trivial point-wise optimisation problem that can be solved exactly for the current linearisation point of the data term ϵ.

Stuehmer et al. (2010) provide a solution to the multiple view data term version:

$$E_\theta(D) = \int_\Omega \left\{ |\nabla D|_1 + \frac{1}{2\theta}(D - v)^2 + \sum_{i \in \mathbf{I}} \lambda|\epsilon_i(v)|_1 \right\} dx \,. \tag{5.2}$$

Here the summation in the data term is for each available image $i \in \mathbf{I}$ where the current estimate of depth has a valid projection inside of the image boundary. Noting that the derivative of each ℓ_1 penalty in the data term is a sign function, their multiple data term solution involves searching directly over the set of critical points in $\frac{1}{2\theta}(D - v)^2 + \sum_{i \in \mathbf{I}} \lambda|\epsilon_i(v)|_1$ where the derivative is not defined. A minimum energy must exist at one of those points otherwise within a region which has an analytically defined solution based on the sign of each of the terms in the summation. The iterative alternating solution is embedded into a coarse to fine framework and implemented on a commodity GPU architecture enabling near real-time performance using up to 5 input views per depth map at an image resolution of 480×360.

Ranftl et al. (2012) recently presented a variational two-view stereo system employing a robust ℓ_1 penalty over a census descriptor based stereo error term, together with the second order variant of TGV regularisation developed by **Bredies et al.** (2010). They further extend the regularisation to use include the image driven anisotropic diffusion scheme from **Nagel and Enkelmann** (1986). The depth map solution is computed using the first order primal-dual algorithm developed by **Chambolle and Pock** (2011). They further use a preconditioning of the saddle-point formulation to increase the speed of convergence, (**Pock**

and Chambolle, 2011). They demonstrate their system in an automotive setting showing increased robustness to large lighting variation gained from using the census descriptor, in combination with the high quality sub-pixel solution common to variational formulations further demonstrating the improved smoothness from TGV_α^2 for slanted surfaces common in man made environments.

5.2 Models using Convex Optimisation

We now investigate primal-dual formulations for multiple view stereo using the local stereo data term and regularisation terms demonstrated previously in a depth map denoising setting in Chapter (4). Here we make use of a first order linearisation of the data term given an initial estimate of the solution, resulting in a fully convex optimisation step that is solved using the first order gradient descent technique. First we state the linearisation of the single and patch data terms and then specify the multiple view stereo models. As in the depth map denoising models we look to obtained a solution u, which can encode either a depth map or inverse depth map.

5.2.1 Computing the Linearised Error

The linearised version of the single pixel multi-view stereo error function is obtained by using a first order Taylor series expansion of $\mathcal{I}_k(\mathbf{w}(x,k,d))$ around a point $d_0(x)$:

$$\mathcal{I}_k(\mathbf{w}(x,k,d)) \approx \mathcal{I}_k(\mathbf{w}(x,k,d_0)) + (d - d_0)\nabla_d \mathcal{I}_k(\mathbf{w}(x,k,d_0)\ . \tag{5.3}$$

The linearised error function $\tilde{\rho}(x,k,d)$ is therefore:

$$\tilde{\rho}(x,k,d) = \mathcal{I}_r(x) - \mathcal{I}_k(\mathbf{w}(x,k,d_0)) - (d - d_0)\nabla_d \mathcal{I}_k(\mathbf{w}(x,kd_0))\ , \tag{5.4}$$

where the gradient, $\nabla_d \mathcal{I}_k(\mathbf{w}(x,k,d_0)) \triangleq \frac{\partial \mathcal{I}_l(\mathbf{w}(x,d))}{\partial d}\big|_{d=d_0}$ is evaluated via the chain rule as:

$$\frac{\partial \mathcal{I}_k(\mathbf{w}(x,k,d))}{\partial d} = \frac{\partial \mathcal{I}_k(\mathbf{w}(x,k,d))}{\partial \mathbf{w}(x,k,d)} \cdot \frac{\partial \mathbf{w}(x,k,d)}{\partial K T_{kr} K^{-1} \dot{x} \xi(x)} \cdot \frac{\partial K T_{kr} K^{-1} \dot{x} \xi(x)}{\partial \xi(x)} \cdot \frac{\partial \xi(x)}{\partial d}, \tag{5.5}$$

and $\frac{\partial \xi(x)}{\partial d}$ is the derivative of the inverse depth function, or if a solution in uniform depth is estimated $\frac{\partial \xi(x)}{\partial d} = 1$.

Given an initial solution point the linearised error term can therefore be computed and used in any of the convex global optimisation models previously discussed. As we will see below, in the multiple image setting optimisation will proceed similarly to the 2.5D fusion or multiple image denoising schemes described in Chapter (4), replacing each summand in the denoising error term $u - d_k^{min}$, with the linearised error function $\tilde{\rho}_k$.

Linearising Patch Data terms

Use of the mean subtracted patch error in Equation (4.9) is more involved. The term sums up pixels errors within a patch of pixels which are first penalised under either an ℓ_1 or *Huber* penalty, yielding a non-linear derivative undefined at 0. If we use the convex conjugate of the norm within each patch the result is an explosion of dual variables required: one for each pixel within each patch of each image $\mathcal{I}_k$. Therefore, we will instead make use of the primal form of the patch data term, and when using the ℓ_1 norm resort to the ϵ regularised version in Equation (3.52).

Since each pixel within the patch is associated with a different solution value, we take into account the local geometry of the patch by computing the linearisation on a warped version of $\mathcal{I}_k$, computed using the current depth map estimate. The approximated partial derivative of the patch error can then be computed around the depth estimate of the current solution point from the central pixel of the patch. Using penalisation function $\psi_D(s^2)$ the linearised patch data term is:

$$\psi_P(\tilde{\rho}_P(x,k,d)) = \sum_{y \in n(x)} \left. \frac{\partial \psi_D(\rho_P(x+y,k,d))}{\partial \rho_P(x+y,k,d)} \right|_{d_0} \left. \frac{\partial \rho_P(x+y,k,d)}{\partial d} \right|_{d_0} , \tag{5.6}$$

$$\begin{aligned} \frac{\partial \rho_P(x+y,k,d)}{\partial d}|_{d_0} = \Big(& \mathcal{I}_k(\mathbf{w}(x+y,k,d_0)) - \mu_k(d_0) \\ & + (d-d_0)(\nabla_d \mathcal{I}_k(\mathbf{w}(x+y,k,d) - \mu_{grad}) \\ & - \mathcal{I}_r(x+y) + \mu_r(x) \Big) . \end{aligned} \tag{5.7}$$

Given the initial solution point d_0, we define the warped image $\mathcal{I}_k^w = \mathcal{I}_k(\mathbf{w}(x+y,k,d_0))$ we compute:

$$\mu_k(d) = \mu_k(d_0) + (d-d_0)\mu_{grad}(d_0) , \tag{5.8}$$

where $\mu_k(d_0)$ is $\mathcal{N}_{\sigma_P^2} * \mathcal{I}_k^w$. Since convolution is distributive and associative, the blurred warped image derivatives $\mu_{grad}(d_0) = \boldsymbol{\nabla}_x \left(\mathcal{N}_{\sigma_P^2} * \mathcal{I}_k^w \right)$ can be computed either by the gradient of μ_k or by Gaussian convolution of $\boldsymbol{\nabla}_x \mathcal{I}_k^w$.

5.2.2 Weighted Huber-ℓ_1 Stereo

Using the Huber penalty over first order solution gradients $\|\nabla u\|_\gamma$ weighted using the inhomogeneous isotropic diffusion from Equation (4.26), together with an ℓ_1 penalised data term, $|\tilde{\rho}_k(x, u(x))|$ for each point in the reference frame $x \in \Omega$ and summed over multiple views $k \in \mathcal{K}$, we obtain the weighted Huber-ℓ_1 model:

$$\min_u \int_\Omega \left\{ \sum_{k=1}^{N} \lambda \, |\tilde{\rho}_k(x, u(x))| + g||\nabla u||_\epsilon \right\} \mathrm{d}x \, . \tag{5.9}$$

5.2.3 TGV-Huber Stereo

Incorporating the second order **TGV** regularisation with the sum of the linearised data terms each under a Huber penalty, we arrive at the multiple view stereo version of the TGV-fusion algorithm introduced by **Pock et al.** (2011):

$$\min_{u,v} \left\{ \int_\Omega \alpha_0 \|\nabla u - v\|_1 + \alpha_1 \|\mathcal{E} v\|_1 + \sum_{k=1}^{N} |\tilde{\rho}_k(x, u(x))|_\epsilon \right\} \mathrm{d}x \tag{5.10}$$

5.2.4 Primal-Dual solutions

As outlined in Section (5.1), **Stuehmer et al.** (2010) presented a novel solution to the TV-ℓ_1 form of the multiple view stereo depth estimation using a uniform depth formulation. They utilise duality on the smoothness term and solve the minimisation arising from the sum of ℓ_1 norms on the linearised data term using a generalised thresholding approach which requires sorting of the critical points of the solution at each iteration; a computationally expensive operation for implementation on commodity GPGPU hardware, which they achieve.

Primal-Dual Weighted Huber-ℓ_1 Stereo

If instead we use the Legendre-Fenchel transform on the sum of ℓ_1 norms used in the data term we can avoid the computational burden of sorting the critical points of the solution for each frame used in the data term. The full primal-dual stereo model is given by:

$$\begin{aligned} \min_u \max_{p,r} \quad & \left\{ \textstyle\sum_{x\in\Omega} \sum_{k=1}^{N} \langle r_k, \tilde{\rho}(x, k, u(x)) \rangle + \langle p, \nabla u \rangle - \frac{\epsilon}{2} \|p\|^2 - \delta_{|p|\leq 1} \right\} \\ \text{subject to} \quad & \|r_k\|_\infty \leq \lambda \, , \|p\|_\infty \leq g \, . \end{aligned} \tag{5.11}$$

Solution of the primal-dual model begins each iteration by computing the new linearisation of the data term in Equation (5.4), around the current solution point u^n. Dualisation of the data term norm results in a trivial summing up over each re-projected dual data variable multiplied by the corresponding data term gradient, at the cost of keeping N dual variables in memory. This is identical to multiple image denoising solution, alternating between gradient ascent to solve the dual variables r_k, and p given in Equation (4.40), and the primal variable gradient descent update given in Equation (4.42).

Primal-Dual TGV$^2_\alpha$-Huber Stereo

The primal-dual formulation for the TGV$^2_\alpha$-Huber Stereo model is likewise obtained by applying the Legendre-Fenchel transform on all terms in Equation Equation (5.10):

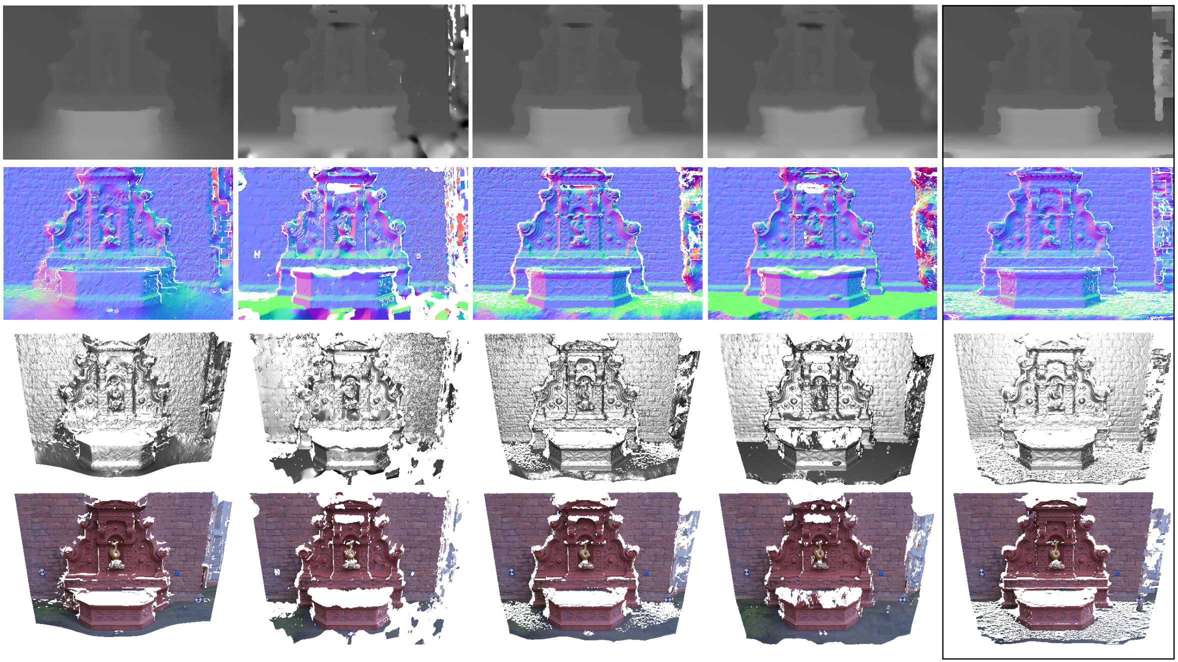
$$\begin{aligned}\min_{u,v}\max_{p,q,r}\Bigg\{\sum_{x\in\Omega}\sum_{k=1}^{N}\langle\tilde{\rho}(x,k,u(x)),r_k\rangle-\langle\nabla u-v,p\rangle+\langle\mathcal{E}v,q\rangle+\\ \left(\delta_{|p|\le\alpha_1}+\delta_{|q|\le\alpha_0}+\delta_{|q|\le 1}+\frac{\epsilon}{2}r^2\right)\Bigg\} \qquad (5.12)\\ \text{subject to}\|p\|_\infty\le\alpha_1\ ,\|q\|_\infty\le\alpha_0\ ,\|r\|_\infty\le 1\ .\end{aligned}$$

The primal-dual model differs from the 2.5D TGV^2_α depth map fusion model developed in **Pock et al.** (2011) by the use of a linearised data term data term $\tilde{\rho}(x,k,d)$. Optimisation again proceeds by alternation of the gradient ascent of the dual variables p, q, r_k with re-projection onto the convex-sets, followed by gradient descent on the primal variable u.

5.2.5 Primal-Dual Multi-view Depth Map Model Comparison

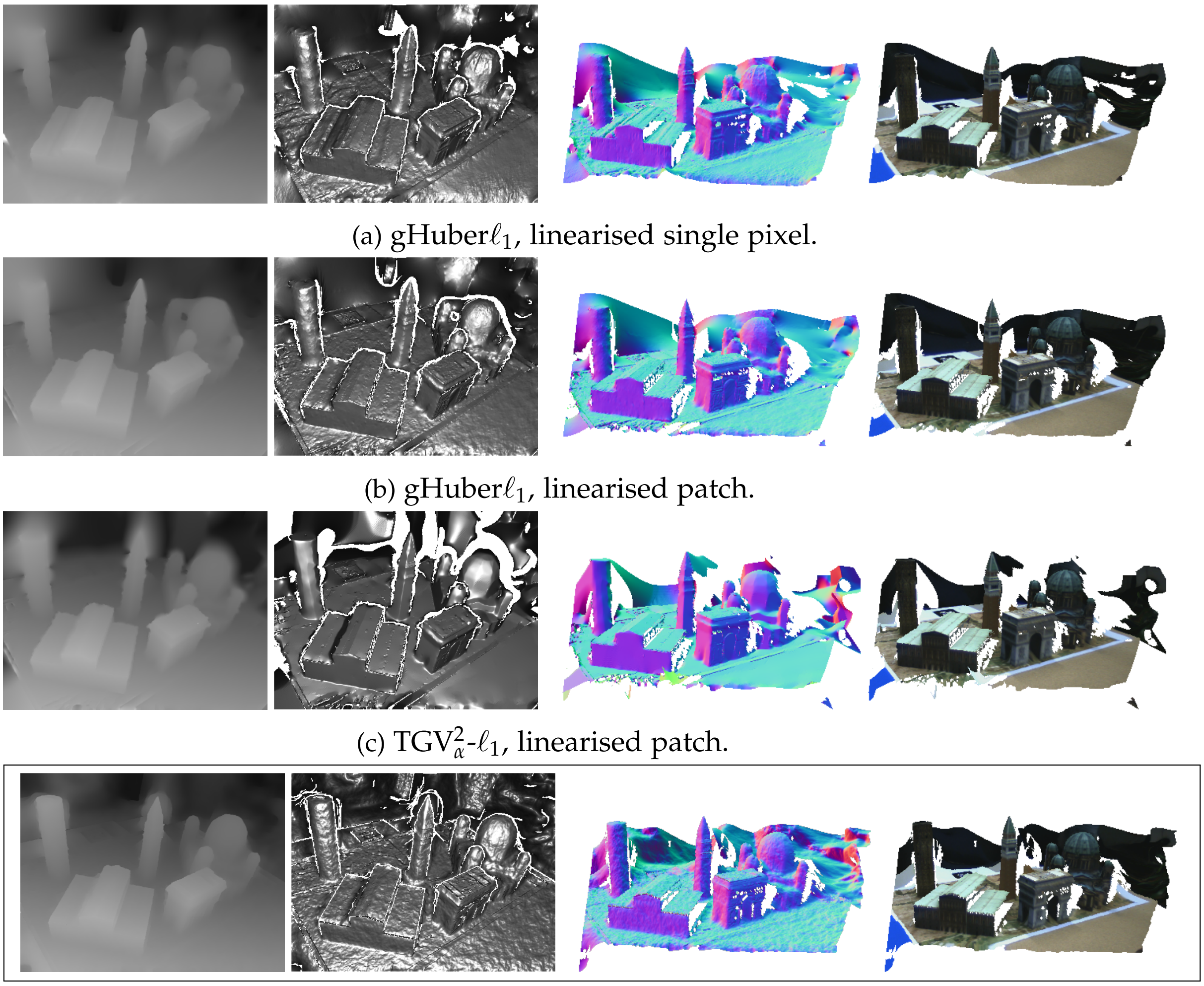
Solution Initialisation: Iterative solution of variational optimisation methods using a linearised data term are typically embedded in a coarse to fine scheme briefly discussed in Section (2.3.2). Instead, we find that initialisation from the full resolution solution from the weighted Huber-ℓ_1 depth map denoising model in Equation (4.23), consistently and significantly outperformed our coarse to fine implementation in both computation time and effectiveness in preventing the optimisation becoming trapped in a local minimum.

In Figures (5.1) and (5.2) we show the solutions resulting from optimisation using the models presented above, for the sub-sampled fountain-P11 data previously described in Figure (4.1), and the City of Sights video frames illustrated in Figure (4.22). Models use either the linearised single pixel error from Equation (5.4), or the mean subtracted patch error term from (5.6), in all cases using the ℓ_1 based penalty. When using the patch based data term, we use the same 3×3 pixel patch as used in the depth map denoising comparison in Subsection (4.5.6).



(a) gHuber-ℓ_1 SP (b) TGV$^2_\alpha$-ℓ_1 SP (c) gHuber-ℓ_1 Patch (d) TGV$^2_\alpha$-ℓ_1 Patch (e) gHuber-ℓ_1 DMD

Figure 5.1: A comparison of convex multiple view stereo depth map models (a-d) highlighting the prominent differences produced by each model on the five subsampled frames from the fountain-P11 dataset shown in Figure (4.1). Each model solution is initialised using the patch based gHuber-ℓ_1 depth map denoising model solution (e) developed in Section(4.5). For each model result we show (top to bottom rows) the denoised depth map, the normal map rendering in the image plane and Phong shaded mesh rendering shown tilted away from the image plane, and finally the textured mesh. Mesh vertices are culled using a visibility threshold to illustrate discontinuities in the depth map (the threshold is constant across results). Despite good initialisation, solutions using the single pixel (SP) linearised data term (a,b) fail to reconstruct the ground plane. This is rectified with the use of the mean subtracted patch based data term (c,d). The TGV$^2_\alpha$ regularisation (d) enables true affine reconstruction for the planar surfaces of the scene. Both patch based multiple view stereo models (c,d) show only a minor increase in surface detail quality in comparison to the initial depth map denoising solution (e).



(a) gHuberℓ_1, linearised single pixel.

(b) gHuberℓ_1, linearised patch.

(c) TGV$^2_\alpha$-ℓ_1, linearised patch.

(d) gHuber-ℓ_1, depth map denoising.

Figure 5.2: A comparison of convex multiple view stereo depth map models (a-c), computed using the City of Sights video data and PTAM camera pose estimation described in Section (4.5.6). We initialise each of the models using the patch based gHuber-ℓ_1 depth map denoising model (d). In (c) we compute the data term minimum from the left-right half sequences for use in 2.5D fusion. Each row shows (left to right) the solution depth map, Phong shaded rendering of the resulting mesh with visibility culled vertices, and a normal map and textured mapped rendering of the resulting mesh from an alternate view. We note that there is little overall improvement in the solution for the chosen frames using either the single pixel (a) or patch based (b) multiple view models in comparison with the initial solution (d), whereas there is a significant improvement for the linearised patch based TGV$^2_\alpha$-ℓ_1 model (c) in comparison to both of the depth map denoising solution using the second order regularisation shown in Figures (4.23b) and (4.23c) at occlusion boundaries near the central tower in the scene.

In the fixed N-view stereo setting, the input frames are a known constant translation and therefore the data-term quality, given a fixed range of depth estimation, is stable. in the multiple view stereo case using a single moving camera, there are instead several mechanisms that can be used to select which frames will be used with a given reference frame. In this chapter and the preceding chapter, we used the simplest mechanism where the temporally nearest $N/2$ frames are select around the reference frame from a video stream, which can lead to large variations in the relative translation of frames altering the quality of the data term, and requiring tuning of the convex model parameters. In Chapter (7) we will return to look at the stereo estimation pipeline in the setting of full dense reconstruction where we integrate hundreds of depth maps into a global surface model. There, we attempt to stabilise the data term quality by automatically selecting a suitable subset of frames from the input stream. We will also make use of the estimated depth map confidence from Equation (4.13) enabling the down-weighting of depth maps with potentially poorer quality data terms. In later chapters we will refer to the process of optimisation using the multiple view stereo model, initialised using the depth map denoising solution as multiple view stereo polishing.

A further point about dynamic selection of the input data set must be made in defence of the potential benefit of the multiple view models, since we find only a small improvement in the quality of solution on the video data set described in Figure (5.2). Specifically, in both evaluations in Figures (5.1) and (5.2) we have used the same input frames in the multiple view stereo model optimisation as when computing the initialising solution with the depth map denoising model. In Chapter (7) we demonstrate that the combination of the two models affords improvement in the solution by first using a small baseline set of frames for depth map denoising, followed by a wider base-line set of frames for the multiple view stereo polishing made possible by the good initialisation provided by the former method.

Practical Model with Primal Data term, Dual Regularisation

The number of variables required for the dual form of each data term can become prohibitive when a large number of images is used, we have therefore also investigated the use of the primal form of the linearised data term, which together with the convex-conjugate form of the regularisation results in a hybrid gHuber-$\ell_1 - \eta$ model, using an η-regularised ℓ_1 penalty:

$$\begin{array}{ll} \min\limits_{u} \max\limits_{p} & \left\{ \lambda \sum_{k=1}^{N} \sum_{x} \sqrt{\tilde{\rho}_k(x,u)^2 + \eta^2} + \langle p, g\nabla u \rangle - \frac{\epsilon}{2}\|p\|^2 - \delta_{\{|p| \leq 1\}} \right\} \\ \text{subject to} & \|p\|_\infty \leq g \; . \end{array} \tag{5.13}$$

Summing over the the η-regularised partial derivatives at the current solution point u^n gives the linearisation of the data term derivative:

$$\sum_{k=1}^{N} \frac{\partial \tilde{\rho}_k^n(x, u_0)}{\partial u} = \sum_{k=1}^{N} \frac{\tilde{\rho}_k(x, u)}{\sqrt{\tilde{\rho}_k(x, u)^2 + \eta^2}} \nabla \rho_k(x, u) \ . \tag{5.14}$$

This single pixel primal data term can be trivially replaced with the linearised form of the mean subtracted patch data term given in Equation (5.6). For either data term, the solution is given by a gradient ascent on the single dual variable for the regularisation term with projection back onto the convex set, followed by gradient descent on the primal variable.

While we find that the use of duality in the *regularisation term* results in benefits in speed of convergence, in practice we have found that using the η-regularised form of the ℓ_1 penalty in the *data term* does not significantly impact the quality or convergence speed of the depth map solution when using a large number (5 to 20) of small-baseline images. This is despite the fact that if the η constant dominates the regularised form of the cost, then as the error decreases, a slowing in the rate of convergence to the solution will occur when using a gradient descent method. While further investigation is required to resolve this, one possible explanation for the observed behaviour is that the operating range of the penalty over a stereo error function is in practice large relative to η, which is practically the case given the numerous sources of error present in the live stereo estimation setting from image noise to inaccuracies in camera calibration.

5.3 Global Cost Volume Optimisation

In this section we introduce of alternative global optimisation model for using many more frames within the data term than is feasible with the approaches described above using a linearised data term. Specifically, we look at the advantages that explicitly computing and storing the multiple view stereo cost volume $\mathbf{C}_r$ can bring, illustrated in Figure (5.3). A row $\mathbf{C}_r(x)$ in the cost volume stores the accumulated photo consistency error as a function of inverse depth d. Here, we compute an element in the cost volume using the simple single pixel brightness constancy based term, rather than using a patch-based normalised cost, or pre-processing the input data to increase illumination invariance over wide baselines.

Under the brightness constancy assumption, we hope for ρ to be smallest at the inverse depth corresponding to the true surface. As discussed in Section (4.2), this does not hold for images captured over a widening baseline or even for the same viewpoint when lighting changes. Using the single pixel data term we can however show the advantage of reconstruction from a large number (100s) of video frames taken from very small baseline over a short space of time.

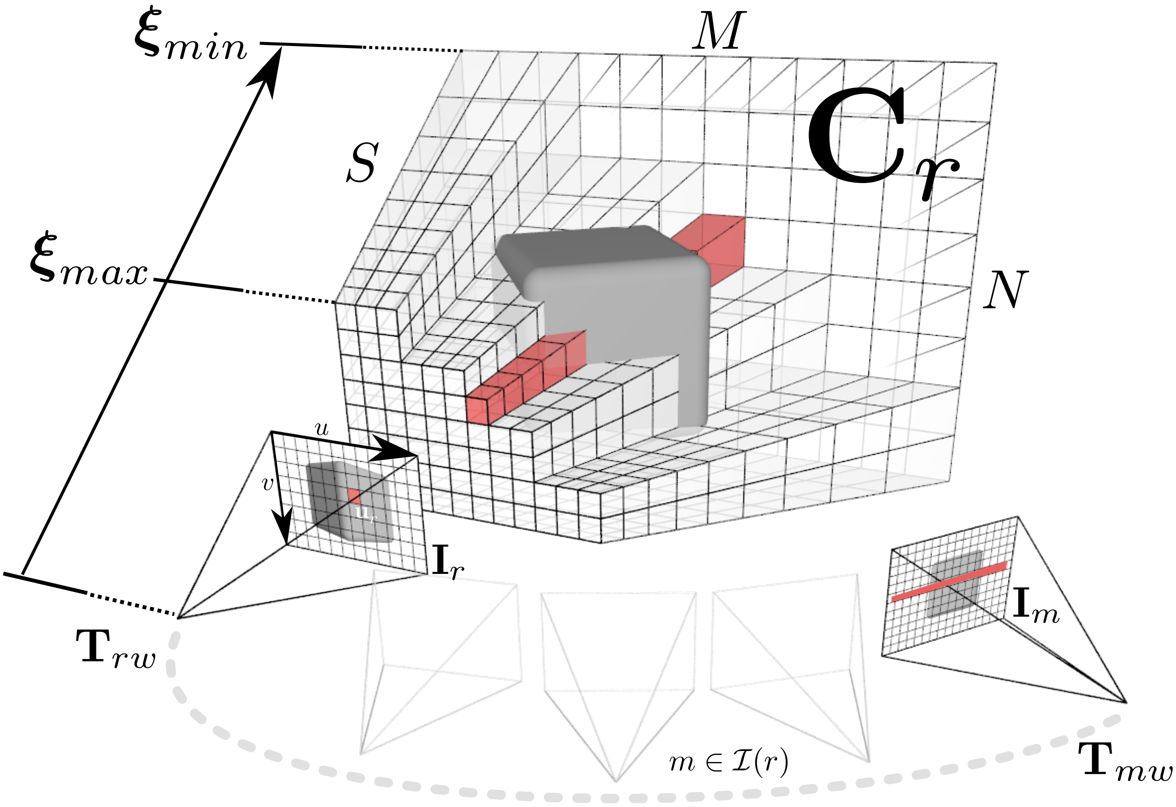
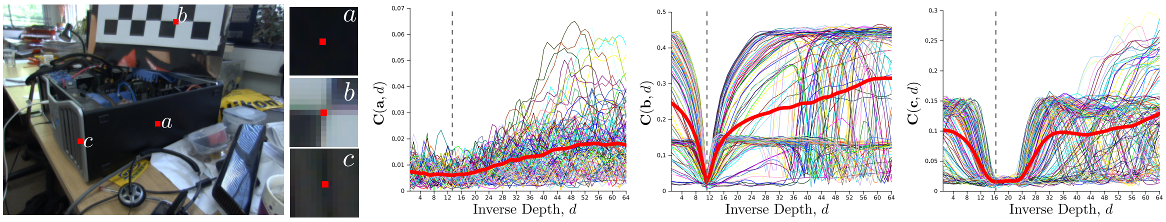


Figure 5.3: A keyframe r consists of a reference image $\mathbf{I}_r$ with pose $\mathtt{T}_{rw}$ and data cost volume $\mathbf{C}_r$. Each pixel of the reference frame $x_r \in \Omega$ has an associated row of entries $\mathbf{C}_r(x)$ (shown in red) that store the average photometric error or cost $\mathbf{C}_r(x, d)$ computed for each inverse depth $d \in \mathcal{D}$ in the inverse depth range $\mathcal{D} = [\xi_{min}, \xi_{max}]$. We use tens to hundreds of video frames indexed as $m \in \mathcal{I}(r)$, where $\mathcal{I}(r)$ is the set of frames nearby and overlapping r, to compute the values stored in the cost volume.



(a) (b) (c)

Figure 5.4: Plots for the single pixel photometric functions $\rho(x)$ and the resulting total data cost row $\mathbf{C}(x)$ are shown for three example pixels in the reference frame, chosen in regions of differing discernibility. Pixel (a) is in a textureless region and not well localisable; (b) is within a strongly textured region where a point feature might be detected; and (c) is in a region of linear repeating texture. While the individual costs exhibit many local minima, the total cost (thick red line) shows clear a clear minimum in all except for nearly homogeneous regions.

In Figure (5.4), we show plots for three reference pixels where the function ρ has been computed and averaged to form $\mathbf{C}(x)$. It is clear that while an individual data term ρ can have many minima, the total cost generally has very few and often a clear minimum. As shown in Figure (5.5), an inverse depth map can be extracted from the cost volume by computing $\text{argmin}_d \, \mathbf{C}(x, d)$ for each pixel x in the reference frame. It is clear that the estimates obtained in featureless regions are prone to false minima. As in the previous stereo estimation models, we therefore seek an inverse depth map ξ which minimises an energy functional that regularises the photometric data term cost with a smoothness term. The energy functional, combining the cost volume with the weighted Huber regularisation is:

$$\min_{\xi} \left\{ \int_{\Omega} \mathbf{C}(x, \xi(x)) \; dx + \lambda \int_{\Omega} g |\nabla \xi|_{\gamma} \; dx \right\} . \tag{5.15}$$

We previously demonstrated variational optimisation approaches to minimising the above functional that approximate the data term by linearising the data term, and solving the resulting approximation iteratively. Typically such schemes are embedded within a coarse to fine optimisation framework that can lead to loss of reconstruction detail and do not guarantee avoidance of local minima. Also, when the linearisation is performed directly in image space as in **Stuehmer et al.** (2010) and throughout Section (5.2), all images used must be kept in working memory. Moreover, all images must be recalled within each iteration of the optimisation for computing the new linearised data term. This leads to optimisation times which scale linearly in the number of images used. In the single image depth map denoising approaches from Section (4.5), we took advantage of extracting the per pixel minimum of the cost volume which was then denoised. Since aggregation into the cost volume is independent of the global depth map denoising optimisation, many

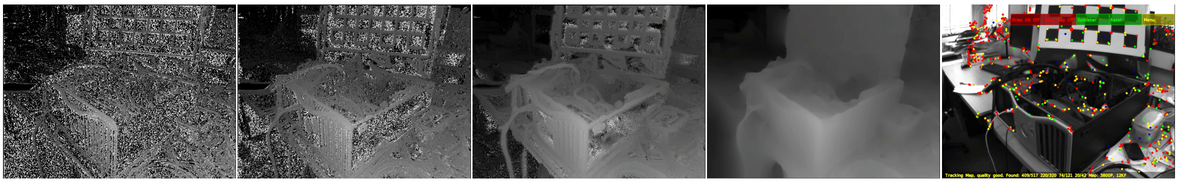


Figure 5.5: Incremental cost volume construction; we show the current inverse depth map extracted as the current minimum cost for each pixel row $d_u^{min} = \text{argmin}_d\, \mathbf{C}(u, d)$ as 2, 10 and 30 overlapping images are used in the data term (a-c). Also shown is the regularised solution that we solve to provide each keyframe inverse depth map (d). In comparison to the nearly 300×10^3 points estimated in our keyframe, we show the ≈ 1000 point features comprising the current scene structure estimate in the same frame for localisation in PTAM (**Klein and Murray** (2007)) (e). In Chapter (8) we demonstrate the use of the dense reconstruction to perform dense tracking, increasing tracking robustness during rapid camera motion.

more images can be used within the depth map estimation.

In this section, we extend an alternative solution first proposed by **Steinbrucker et al.** (2009) for estimation of large displacement optic flow. Here, we approximate the original global energy functional by coupling the data and regularisation terms through an auxiliary variable $\boldsymbol{\alpha} : \Omega \to \mathbb{R}$:

$$\mathbf{E}_{\xi,\boldsymbol{\alpha}} = \int_{\Omega} \left\{ g(x) \|\nabla \xi(x)\|_{\epsilon} + \frac{1}{2\theta} \left(\xi(x) - \boldsymbol{\alpha}(x)\right)^2 \quad + \lambda \mathbf{C}\left(x, \boldsymbol{\alpha}(x)\right) \right\} \mathrm{d}x \,. \tag{5.16}$$

The coupling term $\mathbf{Q}(x) = \frac{1}{2\theta}(\xi(x) - \boldsymbol{\alpha}(x))^2$ serves to drive the original and auxiliary variables together, enforcing $\xi = \boldsymbol{\alpha}$ as $\theta \to 0$, resulting in the original Energy in Equation 5.15. As a function of ξ, the convex sum $g(x)\|\nabla \xi(x)\|_{\epsilon} + \mathbf{Q}(x)$ is a small modification of the TV-quadratic ROF image denoising model given in Equation (4.21). Crucially, although still non-convex in the auxiliary variable $\boldsymbol{\alpha}$, each $\mathbf{Q}(x) + \lambda \mathbf{C}\left(x, \boldsymbol{\alpha}(x)\right)$ is now trivially point-wise optimisable and can be solved using an exhaustive search over a finite range of discretely sampled inverse depth values. Importantly, the discrete cost volume $\mathbf{C}$ can be computed by keeping the average cost up to date as each overlapping frame $\mathcal{I}_{k \in K}$ arrives removing the need to store images or poses and enabling constant time optimisation for any number of overlapping images. Such an approach potentially combines both the computational efficiency of the depth map denoising schemes with the advantages of full multiple view stereo optimisation schemes that can make use of weaker data terms.

We now detail our iterative minimisation solution for Equation (5.16). We transform the gHuber regularisation term using the Legendre-Fenchel transform into the primal-dual form, and make use of the stacked vector forms of the continuous variables with $\mathbf{d}$ for ξ and $\mathbf{a}$ for $\boldsymbol{\alpha}$. The resulting resulting saddle-point problem in primal variable $\mathbf{d}$ and dual variable $\mathbf{q}$ is coupled with the data term giving the sum of convex and non-convex functions:

$$\mathbf{E}\left(\mathbf{d},\mathbf{a},\mathbf{q}\right) = \left\{ \langle \nabla \mathbf{d}, \mathbf{q} \rangle + \ \ \tfrac{1}{2\theta}\|\mathbf{d}-\mathbf{a}\|_2^2 + \ \ \lambda \mathbf{C}\left(\mathbf{a}\right) - \delta_q(\mathbf{q}) - \frac{\epsilon}{2}\|\mathbf{q}\|_2^2 \right\} . \tag{5.17}$$

For a fixed value $\mathbf{d}$ we obtain the solution for each $a_x = \mathbf{a}(x) \in \mathcal{M}$ in the remaining non-convex function using a point-wise search to solve:

$$\underset{a_x \in \mathbf{D}}{\operatorname{argmin}} \ \ \mathbf{E}^{\text{aux}}(x, d_x, a_x) \ , \tag{5.18}$$

$$\mathbf{E}^{\text{aux}}(x, d_x, a_x) = \ \ \tfrac{1}{2\theta}\left(d_x - a_x\right)^2 + \lambda \mathbf{C}\left(x, a_x\right) \ . \tag{5.19}$$

The complete optimisation starting at iteration $n = 0$ begins by setting dual variable $\mathbf{q}^0 = 0$ and initialising each element of the primal variable with the data cost minimum, $d_x^0 = a_x^0 = \operatorname{argmin}_{a_x \in \mathcal{M}} \mathbf{C}(x, a_x)$, we then perform the following fixed point iterations in alternation.

1. Fixing the current value of $\mathbf{a}^n$ we perform a semi-implicit gradient ascent on the dual variable:

$$\mathbf{q}^{n+1} = \Pi_{\mathbf{g}}\left((\mathbf{q}^n + \sigma_{\mathbf{q}} \nabla \mathbf{d}^n)/(1+\sigma_{\mathbf{q}}\epsilon)\right), \tag{5.20}$$

   and gradient descent on the primal variable:

$$\mathbf{d}^{n+1} = (\mathbf{d}^n + \sigma_{\mathbf{d}}(GA^\top \mathbf{q}^{n+1} + \frac{1}{\theta^n}\mathbf{a}^n))/(1+\frac{\sigma_{\mathbf{d}}}{\theta^n}) \tag{5.21}$$

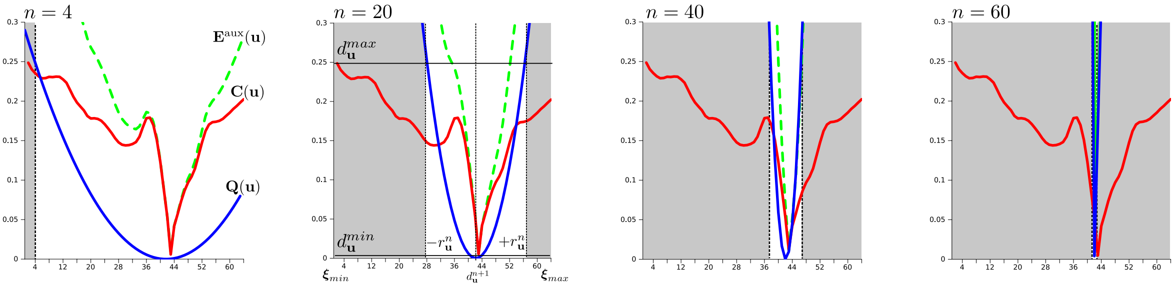
2. Fixing $\mathbf{d}^{n+1}$, we then perform a point-wise exhaustive search for each a_x^{n+1} in the discretised inverse depth space $\mathcal{M}$ solving the minimisation in Equation (5.18).

3. If $\theta^n > \theta_{end}$ update $\theta^{n+1} = \theta^n\left(1 - \beta n\right)$, $n \leftarrow n + 1$ and `goto` (1), otherwise end.

Accelerating the Non-Convex Solution

The exhaustive search over the the sample space $\mathcal{M}$ to solve Equation (5.18) ensures global optimality of the iteration (within the sampling limit). We now demonstrate in Figure (5.6) that there exists a deterministically decreasing feasible region within which the global minimum of Equation (5.19) must exist, considerably reducing the number of samples that need to be tested.

For a pixel x, the known data cost minimum and maximum are $C_x^{max} = \mathbf{C}(d_x^{max})$ and $C_x^{min} = \mathbf{C}(d_x^{min})$. These are trivial to maintain when building the cost volume. As both



(a) n=4 (b) n=20 (c) n=40 (d) n=60

Figure 5.6: Accelerated exhaustive search as iterations progress (a-d): at each pixel we wish to minimise the total energy $\mathbf{E}^{aux}(x)$ (green), which is the sum of the fixed data energy $\mathbf{C}(x)$ (red) and the current convex coupling between primal and auxiliary variables $\mathbf{Q}(x)$ (blue). This latter term is a parabola which gets narrower as optimisation progresses, setting a bound on the region within which a minimum of $\mathbf{E}^{aux}(x)$ can possibly lie and allowing the search region (unshaded) to get smaller and smaller.

terms in Equation(5.19) are positive, we know that the minimum value of any cost volume row is just C_x^{min}. This occurs if the quadratic component is zero when $a_x^{n+1} = d_x^{n+1} = d_x^{min}$. In any case, if we set $a_x^{n+1} = d_x^{n+1}$ then we cannot exceed C_x^{max}, resulting in the energy bound:

$$C_x^{min} + \frac{1}{2\theta^n}\left(a_x^{n+1} - d_x^{n+1}\right)^2 \leq C_x^{max}\ . \tag{5.22}$$

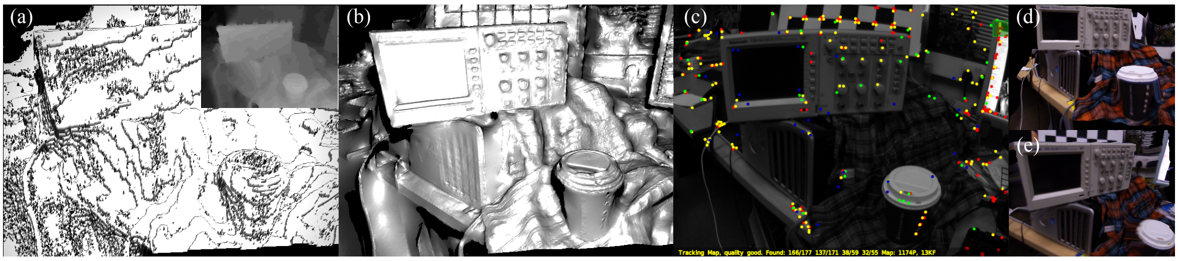
Rearranging for a_u^{n+1} we find a feasible region either side of the current fixed point d_u^{n+1} within which the solution of the optimisation must exist:

$$\begin{aligned} a_x^{n+1} \in & \quad \left[d_x^{n+1} - r_x^{n+1}, d_x^{n+1} + r_x^{n+1}\right] && (5.23) \\ r_x^{n+1} = & \quad 2\theta^n\lambda\left(C_x^{max} - C_x^{min}\right) && (5.24) \end{aligned}$$

As shown in Figure (5.6), the search region size drastically decreases after only a small number of iterations, reducing the number of sample points that need to be tested in the cost volume to ensure the optimality of Equation (5.18).

More sophisticated schemes could be utilised to further decrease the number of points visited in the cost volume optimisation. A simple extension to the above acceleration scheme would be to make use of the fact that the global minimum is also a local minimum. Feasible solutions are therefore zero crossings in the derivative of $\mathbf{E}^{\text{aux}}$ which are sparse.

Such a pre-processing scheme, where data term local minima are used, has previously been demonstrated in the related image-based rendering scheme of **Fitzgibbon et al.** (2005), who compute a novel view synthesis given several calibrated input views by minimising the sum of the photo-consistency data-term error with an image patch prior. They also demonstrate a form of alternating optimisation switching between selection of a preprocessed set of


(a)
(b)
(c)
(d)
(e)
Tracking Map, quality good. Found: 166/177 137/171 38/59 32/55 Map: 1174P, 13KF

(a) (b) (c) (d)

Figure 5.7: Example inverse depth map reconstructions obtained from DTAM using a single low sample cost volume with $S = 32$. (a) Regularised solution obtained *without* the sub-sample refinement is shown as a 3D mesh model with Phong shading (inverse depth map solution shown in inset). (b) Regularised solution *with* sub-sample refinement using the same cost volume also shown as a 3D mesh model. (c) The video frame as used in PTAM, with the point model projections of features found in the current frame and used in tracking. (d,e) Novel wide baseline texture mapped views of the reconstructed scene used for tracking in DTAM.

data-term local minima and the regularisation term energy minimisation.

Further work by **Taylor and Bhusnurmath** (2008) makes use of data-term convexification pre-processing step. They combine a first and second order smoothness prior on the depth map solution with a piecewise linear convex approximation to the stereo data-term error function, forming its lower convex-hull. The resulting convex data term reduces to a set of piecewise linear constraints that together with the smoothness terms results in a linear program which can be optimised using the interior point log barrier method (**Boyd and Vandenberghe**, 2004). In both cases the operations for both specific minimum selection (i.e. sorting) or data-term convexification requires extra computation over the approach described in this subsection. Further work is therefore required to understand the benefits of more sophisticated data-term approximations in relation to the trade-off between data-term approximation error and computational efficiency on modern GPGPU hardware.

Increasing Solution Accuracy

To obtain sub-pixel optical flow accuracy, **Steinbrucker et al.** (2009) increased the sampling density of the cost function. Likewise, it would be possible to increase the density of inverse depth samples $\mathcal{M}$ to increase surface reconstruction accuracy, however this is prohibitively expensive both in the memory requirements for the increased volume resolution, and also for the increased computational time for the per-point search.

Fortunately, as can be seen in Figure (5.6) the sampled point-wise energy $\mathbf{Q}(x)$ is typically well modelled near the discrete minimum with a parabola centred at the true minimum. We can therefore achieve sub-sample accuracy by performing a single Newton step, previously described in Equation (4.14) for data term interpolation, using numerical derivatives of

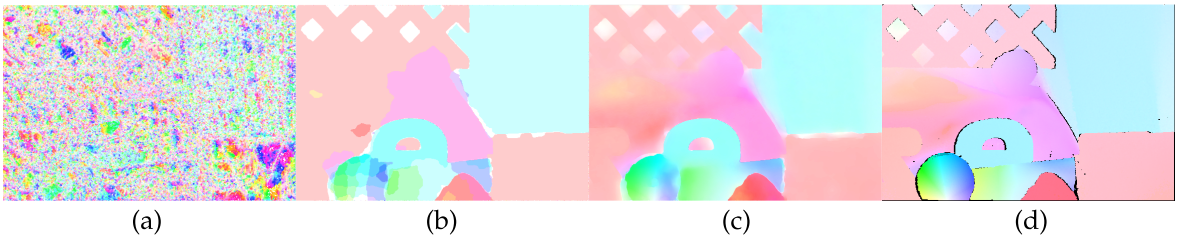


Figure 5.8: Coloured flow fields for the *rubberwhale* two view optical flow data set: Single pixel data term minimum (a); regularised exact search on integer flow vectors (b); regularised exact search with embedded interpolation using Equation (5.25), demonstrating sub-pixel flow solution without the additional computational cost in explicit data term oversampling (c); ground truth flow field (d), ground truth flow field and colouring scheme from (**Baker et al.**, 2011).

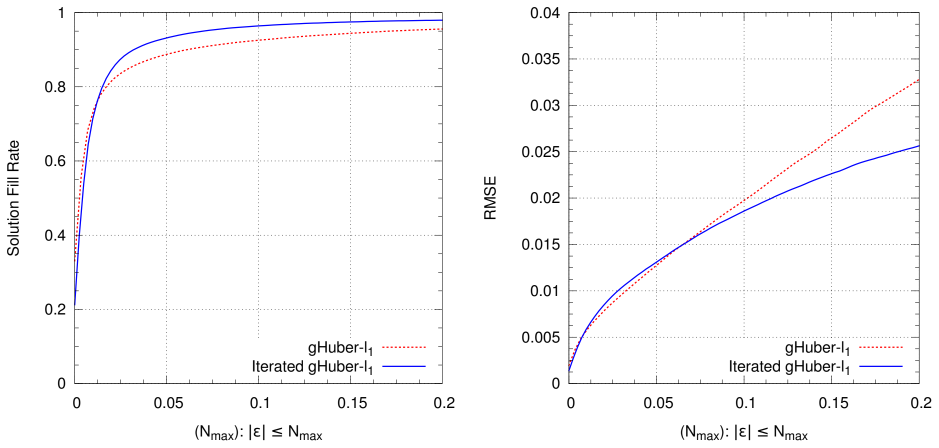
$\mathbf{Q}(x)$ around the current discrete minimum a_x^{n+1}:

$$\hat{a}_x^{n+1} = a_x^{n+1} - \frac{\boldsymbol{\nabla}\mathbf{E}^{\text{aux}}(u, d_x^{n+1}, a_x^{n+1})}{\boldsymbol{\nabla}^2\mathbf{E}^{\text{aux}}(u, d_x^{n+1}, a_x^{n+1})} \,. \tag{5.25}$$

We embed this refinement step into the iterative optimisation scheme by replacing the the discrete a_x^{n+1} with the sub-sample accurate version. It is not possible to perform this refinement post-optimisation, as at that point the quadratic coupling energy is large (due to a very small θ), and so the fitted parabola is a spike situated at the minimum. Post processing based interpolation of the pure data term around the discrete (global) minimum as performed in Equation (4.14) also results in reduced performance since the data term without regularisation is noisy. As demonstrated in Figure (5.7) embedding the refinement step inside each iteration results in vastly increased reconstruction quality, and enables detailed reconstructions even for low sample rates, e.g. $|\mathcal{M}| \leq 64$. We have further applied this optimisation technique to optical flow estimation where the cost volume is defined over a discretisation of pixel translations instead inverse depth. We obtain sub-pixel flows using the above interpolated scheme at no extra memory or computational cost in comparison to the over sampling approach used by **Steinbrucker et al.** (2009), an example optical flow result is illustrated in Figure (5.8).

Setting Parameter Values and Post Processing

Gradient ascent/descent time-steps $\sigma_{\mathbf{q}}, \sigma_{\mathbf{d}}$ are set optimally for the update scheme provided as detailed in **Chambolle and Pock** (2011). Various values of β can be used to drive θ towards 0 as iterations increase while ensuring $\theta^{n+1} < \theta^n\,(1 - \beta n)$. Larger values result in lower quality reconstructions, while smaller values of β with increased iterations result in higher quality. In Chapter (9) we utilise the optimisation method detailed in this section in a full dense visual SLAM system, demonstrating incremental real-time scene reconstruction



(a) Solution fill rates

(b) RMSE

Figure 5.9: Performance analysis of depth map denoising algorithms for reference depth map 5 from the fountain-P11 dataset comparing the gHuber-ℓ_1 depth map denoising approach from Section (4.5) with the iterated denoising model introduced in Section (5.3.1). As described in previous evaluations, we generate the RMSE and image completion (fill) plots obtained for the solution pixels that have absolute error to ground truth within the specified magnitude: N_{max}. It is clear from both the increased cumulative fill rate and reduced RMSE that the iterated model results in a higher quality depth map compared to the original depth map denoising solution.

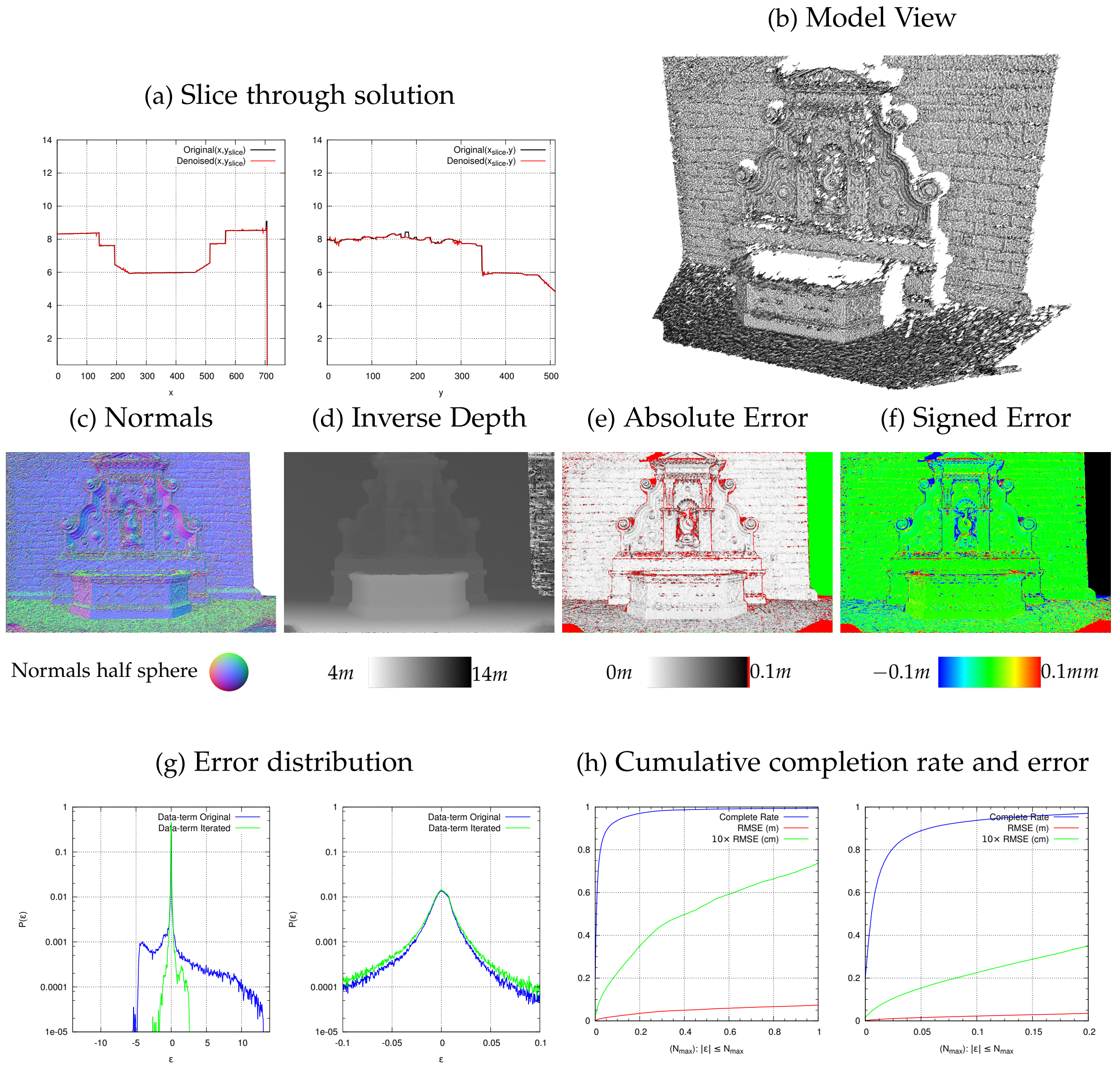
results. In our experiments we set $\beta = 0.001$ while $\theta^n \geq 0.001$ else $\beta = 0.0001$ resulting in a faster initial convergence. We use $\theta^0 = 0.2$ and $\theta_{end} = 1.0e - 4$.

Also in the real-time setting, estimating multiple depth maps using the technique, we note that λ should reflect the data term quality and is therefore set dynamically to $1/(1 + 0.5\bar{d})$, where $\bar{d}$ is the minimum scene depth. This dynamically altered data term weighting sensibly increases regularisation power for more distant scene reconstructions that, assuming similar camera motions for both closer and further scenes, will have a poorer quality data term.

Finally, we note that optimisation iterations can be interleaved with updating of the cost volume, enabling the surface to be made available (though in a non fully converged state) for use in applications in a just in time manner. We demonstrate all of these elements in the dense tracking and mapping (DTAM) system in Chapter (9).

5.3.1 Iterated global cost volume optimisation

To investigate basic capabilities of the non-convex model further, we evaluate a modified and to some extent simplified version of it using the fountain-P11 dataset, used throughout


(a) Slice through solution

(b) Model View

(c) Normals

(d) Inverse Depth

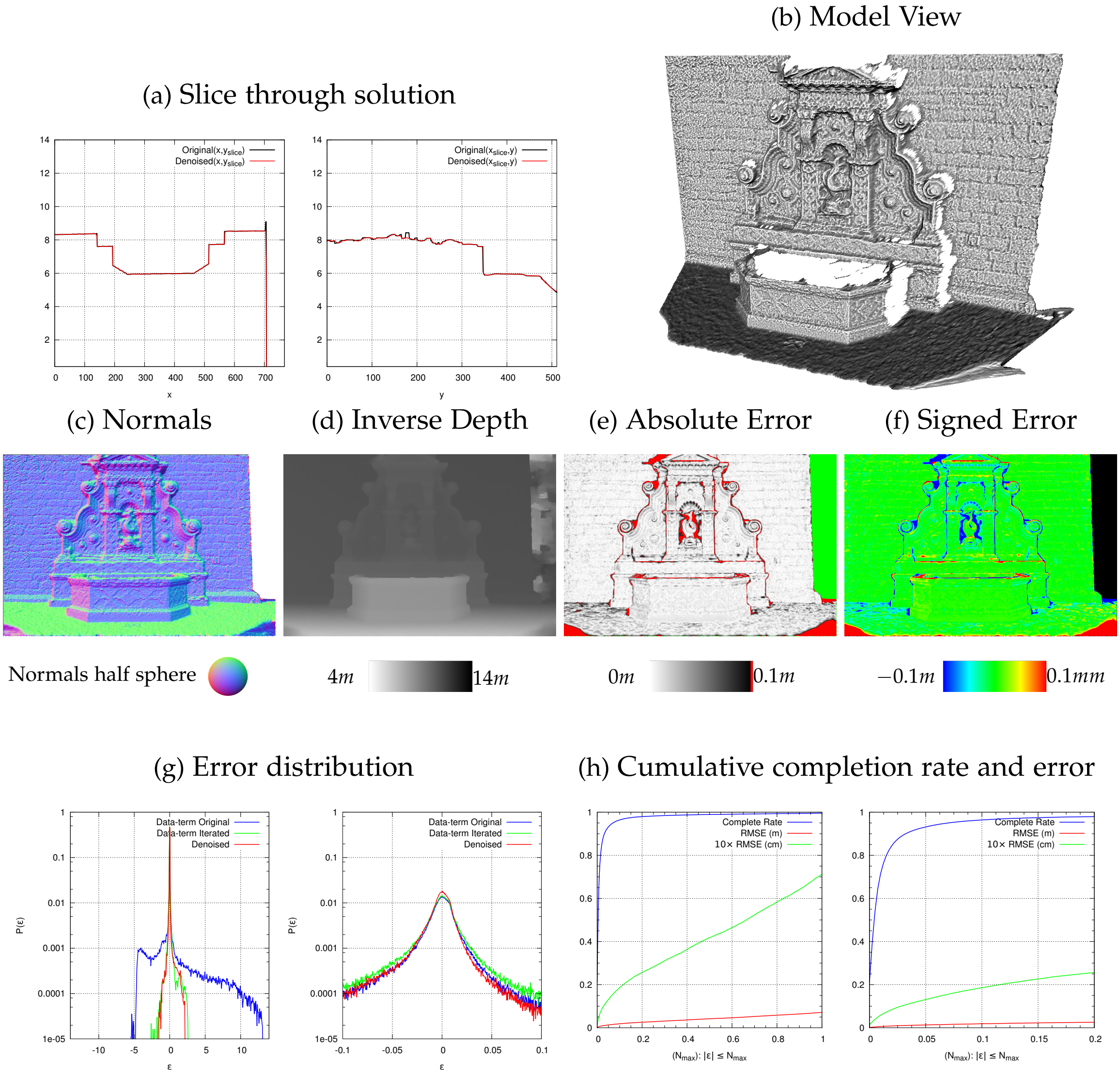
(e) Absolute Error

(f) Signed Error

(g) Error distribution

(h) Cumulative completion rate and error

Figure 5.10: Data-term only at solution to full gHuber regularisation with iterated non-convex data term search. $\lambda = 2$, Huber $\gamma = 0.00159$, image driven regulation $\alpha = 10$, $\beta = 1$. We plot the final (constrained search) depth map data error distribution with the error distribution obtained with the global per pixel cost volume minimum depth value used for initialisation of the optimisation. The error image (e) uses a grey scale to encoded absolute error to the ground truth depth at each pixel up to $0.1m$, is red for solution points with $> 0.1m$ absolute error. Green encode pixels that have no ground truth depth. The signed error is rendered in (f) with saturation at $\pm 0.1m$ error.


(a) Slice through solution

(b) Model View

(c) Normals

(d) Inverse Depth

(e) Absolute Error

(f) Signed Error

(g) Error distribution

(h) Cumulative completion rate and error

Figure 5.11: gHuber regularised solution with iterated non-convex data term search, $\lambda = 2$, Huber $\gamma = 0.00159$, image driven regulation $\alpha = 10$, $\beta = 1$. We plot the solution error histogram against the final (constrained search) data-term minimum depth map and the initialising per pixel cost volume minimum depth.

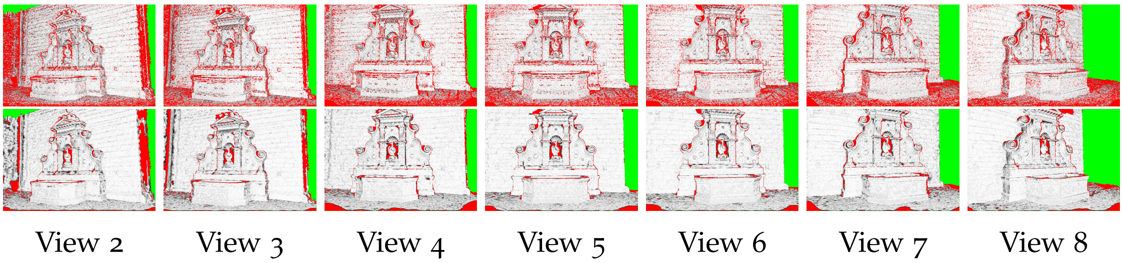
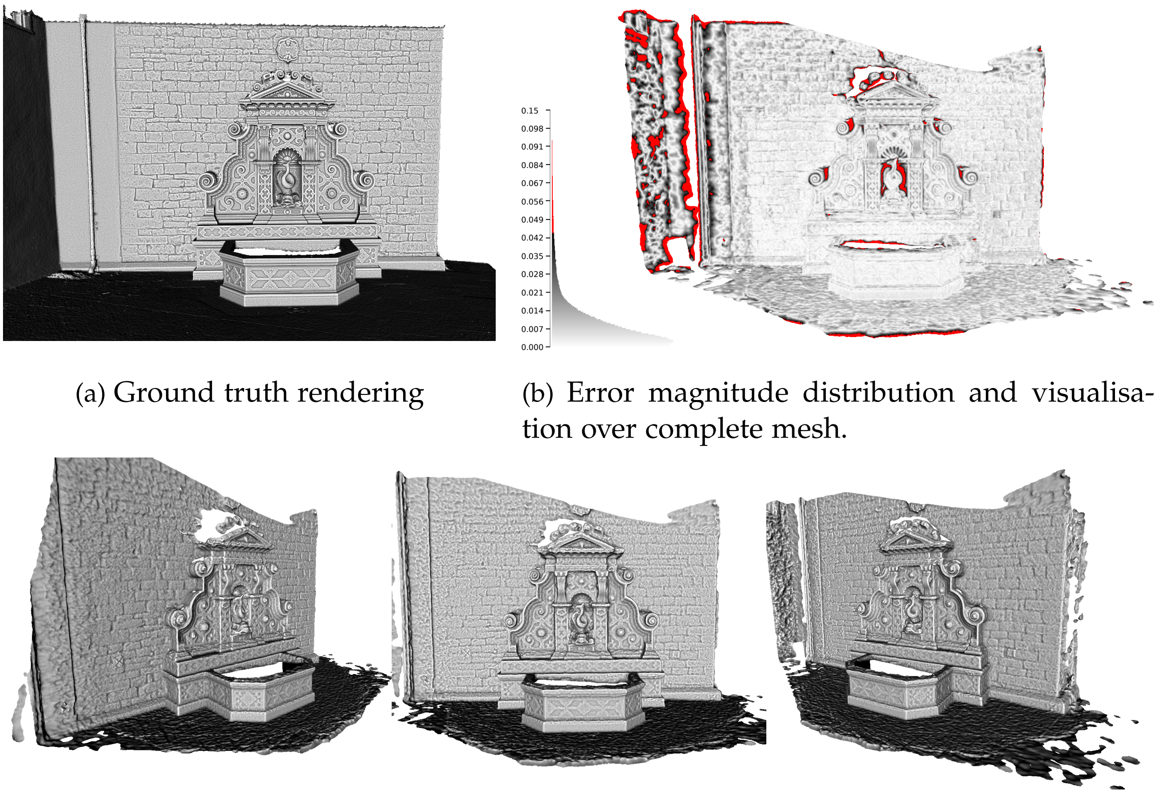


Figure 5.12: Computed absolute error images corresponding to reconstructions for fountain-P11 reference frames 2...8 showing the initialising depth map data terms (top row) and solutions (bottom row). The results are shown for the gHuber regularised solution with iterated non-convex data term search.



(a) Ground truth rendering

(b) Error magnitude distribution and visualisation over complete mesh.

(c) Complete Model Reconstruction

Figure 5.13: Reconstruction of the complete fountain-P11 scene using 7 depth maps. Fusion of the individual depth maps is performed using Poisson surface reconstrution with default settings and maximum octree depth 10, followed by removal faces with connected edges greater than $0.2m$. We compute the absolute error over the entire model (b) against the ground truth rendered in (a), using the Hausdorff distance. The absolute error histogram is shown in meters, note that the error range is halved to $[0 - 0.05]m$ encoded from white to black grey scales to enable greater inspection of errors on the complete model; red pixels show error magnitude above $0.05m$.

the depth map denoising Section (4.5.6). Specifically we note that the model introduced in this section (5.3) interleaves a simple (constrained) per-pixel cost volume minimum search together with a variational denoising step. It is therefore of interest to compare the model using the same parameters for the photometric error as used in the pure denoising evaluation (4.5.6), which uses a simple 3×3 patch based photometric cost. To that end we use the same photometric cost function together with the $gHuber - \ell_1$ model evaluated in Figure (4.19) and trivially interleave estimation of the dataterm minimum constrained to be within ± 20 quantisation steps of the denoised depth map solution from the previous iteration (removing the need to continuously change the search range). The solution is seeded with the output from the gHuber-ℓ_1 model solution. We summarise analysis of the solution against the ground truth depth map and compare the result against the original depth map denoising result in Figure (5.9).

In Figure (5.10) we show that, at the point of solution, the raw data term exhibits greatly decreases noise in comparison to the initialising depth map taken as the per pixel data term minimum of the unconstrained cost volume. In particular a large reduction in outliers at depth discontinuities is obtained. In Figure (5.12) we show the final optimisation result. Most notably in comparison to the depth map denoising solution in Figure (4.19), the iterated search on the cost volume resolves errors in the ground plane, resulting in increased solution fill rate and reduced RMSE. In Figure (5.12) a complete set of depth maps is evaluated against the ground truth and the absolute error and outliers visualised. For each depth map, we use five images in total with two frames arranged either side of the reference frame into which the solution is computed. Finally, we evaluate the result of the complete set of depth maps when fused into a single surface model. In Figure (5.13) we triangulate each of the 7 depth maps, and remove vertices connected by edges longer than $0.2m$, this is a simple heuristic to remove grossly incorrect solution points. We then fuse all depth maps into a single model using the Poisson surface reconstruction technique (**Kazhdan et al.**, 2006) using default settings and with an octree depth of 10. We compute the Hausdorff distance between the reconstructed and ground truth models, and plot the resulting error distribution. We note that although qualitatively the reconstructions appear competitive with systems evaluated by **Strecha et al.** (2008), we can not draw a definitive quantitative comparison on this result since the original evaluation took into account the estimate of ground truth variance. We note that evaluation using the full resolution images would also improve the reconstruction results over the 768×512 resolution frames used here.

5.4 Real-Time Systems Discussion

Due to the value to applications of stereo estimation in a real-time setting, a large body of work continues to deal with engineering implementations of current stereo formulations and mechanisms to achieve higher performance. **Humenberger et al.** (2010) provided a thorough overview and comparison amongst a large selection of two view stereo algorithms implemented across commodity of the shelf platforms which include high performance CPUs, digital signal processors and more recently commodity parallel processing through the use of GPGPU. However,the majority of those systems rely on a rectified image pair and are therefore not there directly applicable to single moving camera multiple-view depth estimation, although such pair-wise rectified stereo estimation has previous been demonstrated in an off-line video based reconstruction system by **Pollefeys et al.** (1999).

In contrast to the large body of research in obtaining stereo in the static camera setting, there is relatively little in the way of real-time or live capable stereo estimation from a single moving camera. **Zach et al.** (2006) and **Gallup et al.** (2007) both demonstrate real-time capable GPU plansweep using a combination of NCC based patch data terms and occlusion handling using the best half sequence technique for use within full scale dense reconstruction pipelines where the stereo estimation is performed with a sliding window of images around each reference frame, however neither of these approaches enable the quality of depth map estimation possible in the global optimisation framework.

The primal-dual approach presented by **Stuehmer et al.** (2010) outlined in Section (5.1) provided the first real-time capable variational single camera depth map implementation. The continuous optimisation formulation leads to a sub-pixel accurate depth estimate which is important in a single camera setting where there is no intrinsic knowledge of a metric scale of the scene, making it hard to pre-set the discretisation on depth required for discrete label based approaches.

In this Chapter we developed several single camera depth map estimation methods, using primal-dual formulations to produce trivially optimisable convex model that are efficiently implemented on modern GPGPU hardware. In Section (5.2) we developed new models that use a patch based data term in contrast to the single pixel data term used by **Stuehmer et al.** (2010). We also demonstrated optimisation of the model using a full dualisation of the primal energy in contrast to the generalised shareholding scheme used in their system. Finally, we introduced the optimisation strategy for the model using a solution initialisation from the gHuber-ℓ_1 depth map denoising method detailed in (4) which we found in practice to provide superior performance to the coarse to fine framework.

Within the set of models investigated in Section (5.2) we again found that use of the total generalised variation regularisation framework resulted in the higher quality reconstruction of planar surfaces. However, as in the depth map denoising models investigated in Chapter (4), computation of the second order model required substantially more processing time despite initialisation of the solution close to the optimum. We are able to achieve a approximately $10 - 20Hz$ depth map processing rate using the gHuber-ℓ_1 model for the dataset examples demonstrated in this Chapter using commodity GPGPU hardware with 10 to 20 frames used in the data term. We therefore have selected the combination of the depth map denoising and multiple view patch based gHuber-ℓ_1 models for use in the incremental dense reconstruction framework developed in Chapter (7).

6

Surface Representation, Integration and Prediction

Contents

In this chapter we detail the mechanisms used to take a stream of depth map surface measurements computed using the multiple stereo methods in Chapters (4) and (5) or from another source such as a depth camera, and produce a consistent dense reconstruction that replaces the point-cloud model used in the feature-based sparse visual SLAM systems.

We are specifically interested in *incremental* dense reconstruction applicable to real-time operation, enabling continuous integration of frame-rate surface measurements. Moreover, the ability to efficiently obtain a rendering of the most up to date surface reconstruction also at frame rate, is central to our aim of closing the tracking and mapping loop in dense SLAM; using the full predictive quality of a dense surface model to increase the efficiency and accuracy of real-time dense reconstruction in Chapter (7) and performing featureless tracking of the current sensor frame against the dense surface model in Chapter (8).

To that end we begin this chapter in Section (6.1) with a description of the surface measurement we assume as input to the system, pointing out the useful structure available in a depth map that constrains the surface reconstruction problem. We provide an overview of the available surface representations, and outline the integration and surface prediction mechanisms associated with them

In Section (6.2) we then detail the chosen surface representation and integration mechanisms that satisfy the incremental integration and prediction requirements: volumetric signed distance functions.

The second half of the chapter is concerned with computing a *geometric and photometric* prediction of the scene. In Subsection (6.3) we describe simple modifications to two classic techniques used for extracting the surface geometry from the volumetric signed distance function. In Section (6.4) we then detail two techniques for representing and rendering the photometric appearance of the surface required for dense passive camera tracking in Chapter (8).

6.1 Surface Reconstruction Approaches

The ability to accurately and efficiently represent surfaces and reconstruct surfaces from noisy data is of great importance in the engineering sciences as a whole. Surface reconstruction methods can be broadly categorised on the basis of the underlying surface representation and particular assumptions about the structure of the available surface measurements. The most permissive type of reconstruction algorithm attempts to solve the least well defined reconstruction problem, assuming nothing more than an organised point cloud as input. The difficulty of the task is greatly increased if the point sampling is irregular as might be obtained from a sparse structure from motion pipeline.

6.1.1 Surface Measurements from Depth Maps

Fortunately our surface measurements have far more information and structure. Specifically, we can exploit the projective nature of the depth map measurement to perform a step discontinuity constrained triangulation of the measurements into a piecewise linear surface mesh, (**Hilton et al.**, 1996). At time k a raw depth map $\mathcal{D}_k$ provides a measurement $\mathcal{D}_k(u) \in \mathbb{R}$ at each valid image pixel in the image domain $u \in \Omega \subset \mathbb{R}^2$. We assume a calibrated camera with known intrinsic parameters K, such that under the assumption that the depth map is computed into a rectilinear frame (details on calibration are provided in Section (3.2), each pixel measurement can be back projected to the 3D point $p_k = \mathcal{D}_k(u)K^{-1}\dot{u}$ in the sensor frame of reference to form a vertex map v_k,

$$v_k(u) = \mathcal{D}_k(u)K^{-1}\dot{u} \ . \tag{6.1}$$

Since each depth map measurement represents the observed surface geometry projected into a regular grid, we can estimate the corresponding normal vector at each grid point $u = (x, y)^\top$, using a cross product between neighbouring map vertices in the depth map:

$$N_k(u) = \nu\big[(v_k(x+1, y) - v_k(x, y)) \times (v_k(x, y+1) - v_k(x, y))\big] \; , \tag{6.2}$$

$$\nu[\mathbf{x}] = \mathbf{x}/\|\mathbf{x}\|_2 \; , \tag{6.3}$$

where $\nu[\mathbf{x}]$ normalises a vector to unit magnitude. We also define a vertex validity mask: $V_k(u) \mapsto 1$ for each pixel where a depth measurement transforms to a valid vertex; otherwise if a depth measurement is missing $V_k(u) \mapsto 0$. If the neighbouring vertices required in Equation (6.2) are invalid we instead look for a vertex at the alternate neighbour, and invalidate the point measurement entirely if the normal can not be estimated. Given the surface vertex and normal estimates at each point we construct the mesh connecting each of valid vertex with the neighbouring vertices used in computing the normal estimate unless there is a surface discontinuity. By computing the angle formed between the pixel ray and the surface normal in the sensor frame of reference we estimate discontinuities where the surface is near perpendicular to the ray.

As we will see later in this section, the explicit grid representation of the depth map provides even more information; implicitly, there is a free space measurement between a vertex and the camera center. We now outline a number of surface reconstruction algorithms, highlighting the underlying surface representations used and their applicability to incremental surface reconstruction.

6.1.2 Explicit Surfaces

A large number of reconstruction methods assume that the representation of the scene can be captured by a specific shape model such as a human body or face, or specific architectural form, thereby turning the reconstruction problem into one of model fitting in a much lower dimensional space (**Szeliski**, 2010). In our more general scene reconstruction setting, the topology and scene type is not fixed before hand. Furthermore, within an incremental reconstruction setting, the topology may change during reconstruction as more data disambiguates a solution. We will therefore review those techniques which do not require domain specific knowledge, but should attempt to exploit the more general prior knowledge about surface continuity suitable for a variety of scenes.

Using Meshes Directly

Turk and Levoy (1994) presented a simple but efficient direct mesh method, stitching together multiple overlapping depth map meshes. They provide algorithms to remove or fuse redundant noisy vertices based on heuristic distance metrics to obtain a single explicit mesh representation of the scene. Such direct mesh operations can also take place using image space visibility constraints between depth maps (**Merrell et al.**, 2007). In such direct mesh zippering and fusion algorithms, the piecewise linear surface is captured by a triangle mesh and operations to change surface topology are handled explicitly.

Delaunay Triangulation of multiple overlapping meshes can also make use of explicit connectivity information. Forms of space carving utilise the visibility constraints available from the depth map structure to provide constraints on the reconstructed surface (**Labatut et al.**, 2007). Direct and explicit meshes have an advantage of being both efficient to store and render. Moreover, in contrast to many of the more sophisticated algorithms we outline next, explicit mesh representation enables the full resolution of the depth map measurement to be maintained in the global scene model. Hence, if a depth map is computed nearer to the surface the resulting mesh will trivially hold a higher resolution representation of the region in comparison with a co-observing depth map computed at a distance.

Oriented patches and Surfels

Surfaces can be represented by oriented patch samples (**Szeliski and Tonnesen**, 1992). By maintaining an unstructured set of surface elements representing the locations, orientations and scales of linear surface elements, the surface can be explicitly specified without the need to initialise or maintain the connectivity information specified in surface meshes. This provides great flexibility in representing complex shape of arbitrary topology and has given rise to a number of useful surface representations. These define the continuous surface via an interpolation, using local neighbourhoods of samples to constraint a local surface reconstruction, for example via a moving least squares approach (**Levin**, 1999; **Alexa et al.**, 2001).

The explicit representation of the surface element also enables direct correspondences to be computed within a dense reconstruction setting, allowing the traditional data-association based optimisation approaches to extend beyond simple point base scene representations (**Furukawa and Ponce**, 2007; **Habbecke and Kobbelt**, 2007; **Lhuillier and Quan**, 2005).

Oriented patches are also one of the dominant representations for real-time rendering of large scale or intricate scenes, offering simple ways to reduce the scene complexity. The popular primitive *splatting* framework efficiently renders such surfaces by drawing for each element a simple 2D primitive such as a disc, in the image, that grossly represents the area covered by the projected 3D primitive (**Rusinkiewicz and Levoy**, 2000; **Pfister et al.**, 2000; **Kobbelt and Botsch**, 2004).

Explicit parametric models

Large regions of urban and office scenes can be approximated using large planar facets (**Gallup et al.**, 2010a; **Sinha et al.**, 2009). Simple parametric forms can vastly reduce the complexity of the scene model while providing a strong prior on the scene reconstruction enabling filling of large regions with noisy or no depth measurement. Reconstruction using large scale simple primitives requires solving a joint segmentation and data association problem, associating measurements from each depth map measurement to a particular plane estimate. **Furukawa et al.** (2009) demonstrated the power of the stricter Manhattan world prior on the scene reconstruction where all surfaces are belong to one of six orthogonal orientations. This holds well in practice in indoor and outdoor setting of modern building reconstructions, but is a poor approximation in general scenes. More recently (**Flint et al.** (2011)) also utilised the Manhattan world assumption, but within an online visual SLAM system to enable maps consisting of semantically meaningful surfaces such as walls and floors in an office environment to be more efficiently constructed.

6.1.3 Implicit Surfaces

Mesh based surface representations suffer from complexity in topological changes that must be explicitly represented, while surfel representations remove any representation of specific surface connectivity. If, however, representation of surface continuity is required, surface reconstruction pipelines using oriented patches typically go on to use the surfels as input to complete surface reconstruction methods based on an implicit surface representation.

For a surface in n dimensional space, an implicit surface S_0 is defined through a scalar field $f : \mathbb{R}^n \mapsto \mathbb{R}$ as the $n-1$ dimensional manifold extracted as the t level set where $f(x) = t$: (**Osher and Fedkiw**, 2002):

$$S_0 \triangleq \{x \in \mathbb{R}^n | f(x) = t\} \ . \tag{6.4}$$

Hence, for zero level set $t = 0$. Implicit surfaces provide a mathematically elegant was to define and work with surfaces of arbitrary topology without explicitly representing surface connectivity.

Early reconstruction approaches computed the field by summing radial basis functions ϕ centred at each point c_i in the data set together with an offsetting function:

$$f(x) = \sum_{i=1}^{n} a_i \phi(x - c_i) + P(x) \ . \tag{6.5}$$

The resulting optimisation problem is then to find the co-efficients a_i of the basis function together with a linear offseting function, $P(x)$, over the space. This can be seen as the solution of a variational optimisation problem (**Turk and OBrien**, 1999; **Hoppe et al.**, 1992) or can be modelled using Gaussian Processes (**Williams and Fitzgibbon**, 2007).

Modern methods incorporate a number of extensions to the basic optimisation approach. **Ohtake et al.** (2003) presents a complete pipeline for scattered data interpolation where oriented point clouds are used as input which we outline here, we later use this method in an earlier variant of a live dense reconstruction system in Chapter (9). Their multi-scale framework defines the implicit surface as a hierarchy of compactly supported radial basis functions that interpolate locally fitted quadric surfaces within local neighbourhoods of points. The points themselves are quantised into nodes of an octree structure. Using unstructured point clouds, no other knowledge of how the point samples relate to the surface sampling is provided and trade-off persists between producing a surface interpolation that fills gaps of under-sampled regions and producing overly smooth reconstructions in regions that have high sampling density.

Given an oriented point set, a point set hierarchy is recursively constructed by clustering samples in an octree-based structure. Level $k = (1, 2, \ldots, M)$ of the hierarchy contains eight equal quadrants. The centroid of the samples inside each quadrant is computed together with an an averaged unit normal value. The hierarchy of centroids $\mathbf{p}_i^k \in P^k$ provides an efficient representation of the original samples, and allows a recursive function representation to be built. Given a base function:

$$f^0(\mathbf{x}) = -1 \; , \tag{6.6}$$

a recursive interpolating function is defined:

$$f^k(\mathbf{x}) = f^{k-1}(\mathbf{x}) + o^k(\mathbf{x}) \; , \tag{6.7}$$

where the offsetting function $o^k(\mathbf{x})$ is solved for each level:

$$o^k = \sum_{\mathbf{p}_i^k \in P^k} \phi_{\sigma^k}(g_i^k(\mathbf{x}) + \lambda_i^k)(\|\mathbf{x} - \mathbf{p}_i^k\|) \; . \tag{6.8}$$

For a given level k, the point hierarchy is approximated by the surface $f^k(\mathbf{x}) = 0$, which provides the basis for the solution at the next finest level of the hierarchy $k+1$. The function $g_i^k(\mathbf{x})$ performs local quadric fitting for P^k. An important aspect of the interpolating function is the strictly positive definite compactly supported basis function $\phi_{\sigma^k}(\mathbf{r})$ for distance $\mathbf{r}$ (**Wendland**, 1995), that has the property $\phi_{\sigma^k}(\mathbf{r}) = 0$ for $\mathbf{r} \geq \sigma$, leading to a sparse system of equations that is solved for coefficients λ_i^k. The support size σ^k is estimated using the density of the original sample set in quadrant k, which ensures that larger support is obtained in sparser areas of the point cloud leading to high quality hole filling capabilities.

Polygonisation of the zero level set, using the method of **Bloomenthal** (1994), is also aided by MSCSRBF since the majority of computation when evaluating the implicit surface is performed at previous levels of the function hierarchy.

Kazhdan et al. (2006) introduced the Poisson Surface Reconstruction (PSR) technique that has become the most widely used approach for surface reconstruction from an unstructured but oriented set of points. Given an oriented point cloud, the PSR method solves for an volumetric implicit surface where the indicator function is constrained to take on the gradient at the points defined by the input oriented point samples. The problem is posed as a sparse, well conditioned linear system of equations which enables various well engineered and out-of-core linear system solvers to be used to perform efficient surface reconstruction on very large scale data-sets, (**Bolitho et al.**, 2007). More recently commodity GPGPU hardware implementations of the technique have demonstrated near real-time surface reconstruction using PSR exploiting data parallel octrees (**Bolitho et al.**, 2009; **Zhou et al.**, 2011). While the approach is promising, the technique does not explicitly make use of the free-space information provided by projective depth map surface measurements. The state of the art data parallel octree approaches can perform real-time reconstruction and surface extraction for nearly 0.5 million points using modern commodity hardware, but if depth maps are generated at VGA resolution and a $30Hz$ frame-rate, a real-time surface reconstruction system must be able to cope with up to 10 million new points per *second*, unless some form of sub-sampling on the data is performed.

6.1.4 Volumetric approaches

Volumetric modelling methods enable representation of not only the surface manifold or volume occupancy, but can also represent and distinguish between free space and regions for which the occupancy is unknown. This distinction between free and unknown space naturally arises when the observations over the surface, for example in the form of a depth map, can provide free space information but can clearly not provide information for space beyond the measured surface.

Occupancy Grids

Introduced by **Moravec** (1988) and **Elfes and Matthies** (1987), the vast majority of occupancy grid mapping research has been performed in robot environment mapping and navigation, predominately in a $2D$ setting. Occupancy grids (in 2D) and volumes (in 3D), represent the environment using a discretisation of the working volume into evenly spaced cells. The value at each cells is used to indicate whether the state of the space it represents is occupied, free, or unknown. In a probabilistic form, by representing the map as a discrete grid of binary random variables $m(x) \in \mathbf{m}$, occupancy grid mapping can be formalised as a binary state estimation problem (**Thrun and Bücken**, 1996; **Thrun et al.**, 2005). Given each sensor pose T_i associated with a measurement over the map state space z_i, and assuming independence between the state of cell in the map representation, the problem of obtaining a consistent map can be factorised $p(\mathbf{m}|z_{1:t}, T_{1:t}) = \prod_x p(m(x)|z_{1:t}, T_{1:t})$, enabling a maximum likelihood estimate of the map to be obtained incrementally using an inverse depth measurement model that directly provides a probability density function over the occupancy of each cell: $p(m(x) = occupied|z_i, T_i)$.

In the $3D$ setting, volumetric storage requirements can become prohibitive for large scale mapping and reconstruction operations, and since a great amount of redundancy exists in large areas of free space, an octree representation can be utilised to great effect (**Szeliski**, 1993; **Wurm et al.**, 2010). Furthermore, while $3D$ occupancy grids can effortlessly represent any surface topology, a substantial saving in memory can be obtained if the topology of the scene can be adequately approximated as homeomorphic to the disc. In this case a height-field representation is suitable, representing the height of the occupying region in each cell with little extra memory requirement over a 2D occupancy map, **Gallup et al.** (2010b).

Surfaces are extracted from probabilistic occupancy grids as the set of modal points in the distribution over some given occupancy threshold, resulting in a ridge detection problem. This lack of explicitly defined interface is inconsequential to roboticists that have used the maps acquired predominantly to avoid interaction (collision) with surfaces in robot navigation. However, such difficulty in precisely defining the surface interface makes the method less useful when the primary goal is to capture and render the highest quality surface reconstruction possible.

Volumetric Signed Distance Functions

A distance function is a volumetrically defined implicit surface in the form of Equation (6.4): as:

$$d(x) = \min_{x \in \partial\Omega} (\psi(x - x_c)) \; , \tag{6.9}$$

for all points x in the volume, where $x_c \in \partial\Lambda$ defines the point set over the surface in the volume and $\psi(.)$ is a distance metric. The signed distance function S is then defined,

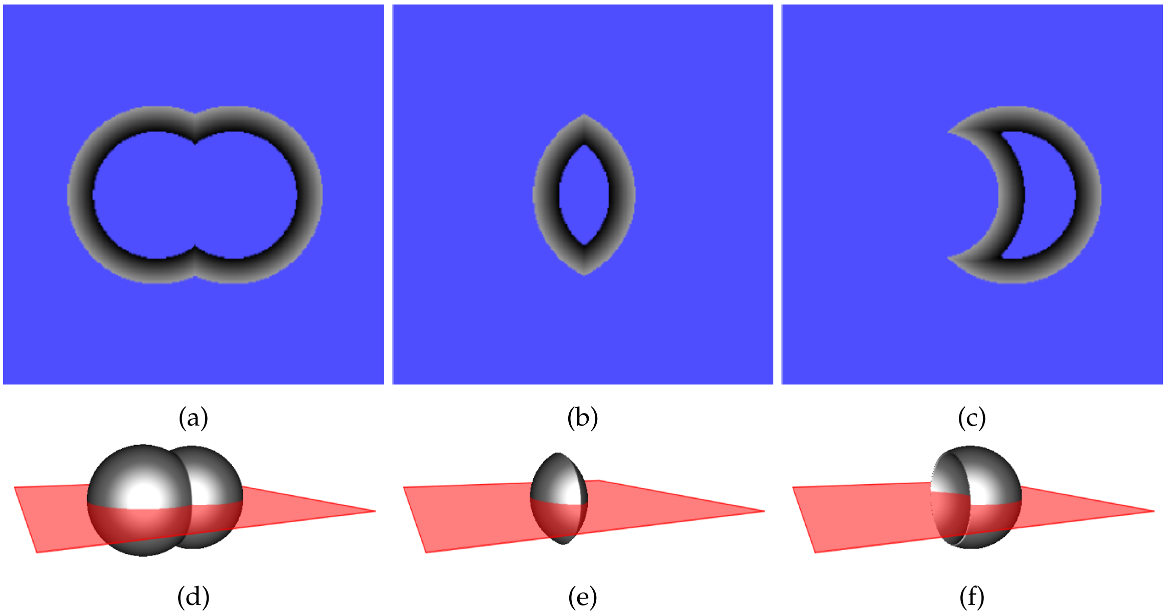


Figure 6.1: Demonstration of the union (a,d), intersection (b,e) and set subtraction (c,f) operations applied to two displaced spheres using the signed distance function form. In (a-d) a slice through the resulting implicit surfaces are shown for the surface functions (d-f). The SDF is shown truncated within a fixed magnitude to illustrate the nature by which a surface can be represented as a thin band around the surface interface, implicitly represented using the signed distanced. Importantly this truncated representation, when applied to depth measurements enables a simple approach to incremental surface reconstruction, described in this section.

(**Osher and Fedkiw**, 2002):

$$S(x) = \begin{cases} -d(x) & \text{iff } \ x \in \Lambda^- \\ 0 & \text{iff } \ x \in \partial\Lambda \\ d(x) & \text{iff } \ x \in \Lambda^+ \end{cases} . \tag{6.10}$$

Here the volumetric function is divided into the two regions which we will usefully think of here as defining the points within free space $x \in \Lambda^+$ and points in non-observable space , Λ^-, separated by the surface at the zero level set $S(x) = 0$. In Figure (6.1), this implicit surface representation for two spheres undergoing set operations in shown. Such set operations can be applied to the volume values directly, enabling solid modelling operations that would require intricate mesh operations.

Hilton et al. (1996) and **Curless and Levoy** (1996) independently introduced signed distance function (SDF) integration algorithms that have become the gold standard for surface reconstruction from *dense* 2.5D depth map measurements. The volumetric approach introduced by **Curless and Levoy** (1996) in particular has a number of properties that make it suitable for real-time *incremental* reconstruction.

In Chapter (1) we introduced the SDF representation and motivated volumetric truncated SDF reconstruction by formulating it as a simple multiple $3D$ image denoising problem. While the approach is intuitive and motivates later optimal SDF denoising strategies by **Zach et al.** (2007b); **Zach** (2008), it simply asserts that each depth map can be transformed into its truncated signed distance function form, turning the surface reconstruction problem into a volumetric denoising problem. Fortunately, **Curless** (1997) provided the insight needed to understand why the volumetric representation and integration technique work so well, deriving the weighted SDF integration approach as an optimal solution for reconstructing a surface manifold from multiple $2.5D$ surface measurements.

Curless and Levoy (1996) noted that in contrast to probabilistic occupancy grid surface extraction, the zero-level set of the SDF can be extracted in an efficient manner, since zero crossings in the SDF are well defined. Crucially, rendering a view of the currently reconstructed surface into a virtual camera is both simple and efficient on modern parallel hardware using implicit surface raycasting (**Parker et al.**, 1998). For each pixel in the virtual camera view the associated ray is traversed, with detection of the first zero crossing along the ray indicating the the visible surface element. We detail more efficient direct raycasting and marching cubes iso-surface extraction techniques for extracting the surface geometry in Section (6.3).

Gibson (1998) demonstrated that distance fields better represent high frequency surface transitions using lower volume resolutions than occupancy representations. This important feature stems from the implicit representation of the interface as a zero value or zero crossing in a smooth function that can be efficiently interpolated using few samples. This is in contrast to the step function that results from direct representation of the boundary using explicit occupancy. The ability to define a distance field through an interpolation of fewer samples results in a high level of redundancy in the regular grid representation. **Gibson et al.** (2000) replace regular sampling with an adaptive grid and demonstrate high levels of compression for the surface representation.

Representing and integrating surface measurements in the signed distance function form therefore provides a highly suitable framework for handling the two basic requirements of a live dense reconstruction frame: enabling incremental integration of the massive rates of surface measurements acquired, and enabling rapid rendering of the most up to date surface estimate to provide a surface prediction that we will rely on when we solve the problem of real-time camera tracking.

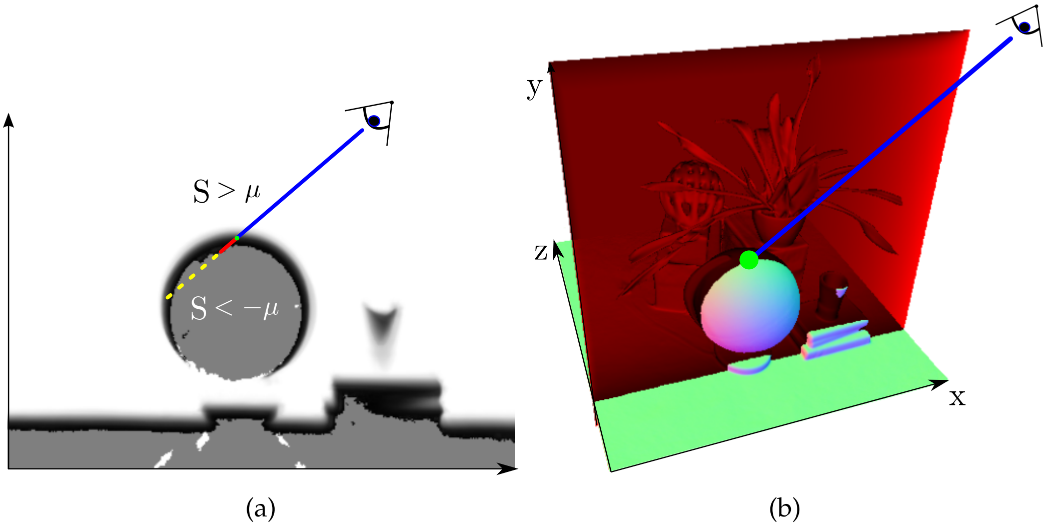


Figure 6.2: A slice through the truncated signed distance volume showing the truncated function $S > \mu$ (white), the smooth distance field around the surface interface $S = 0$ and voxels that have not yet had a valid measurement(grey) as detailed in Equation (6.15). A sample ray is drawn from the camera center intersecting the surface, illustrating the projective truncated SDF approximation with which the TSDF measurement is built. Following the ray traversal through the volume, we colour the ray blue indicating the positive region of the SDF; red throughout the negative SDF region and dashed yellow within the non-represented region. The representation for the projective TSDF illustrated is Figure (6.3a) together the associated weight function in that enables indication of regions with have no valid TSDF values shown in Figure (6.3a).

6.2 Volumetric Signed Distance Function Integration

We now detail the volumetric SDF integration algorithm that enables frame-rate depth map measurements to be integrated into a consistent surface representation. This ability enables us to replace the sparse point cloud scene representations used in feature based SLAM systems with a dense surface substrate that provides far richer surface predictions necessary for the dense tracking methods we develop in Chapter (8).

6.2.1 Mapping as Surface Reconstruction

Each consecutive depth frame with an associated camera pose is fused incrementally into one single 3D reconstruction using the volumetric truncated signed distance function (TSDF) **Curless and Levoy** (1996). In a true signed distance function, the value corresponds to the signed distance to the closest zero crossing (the surface interface), taking on positive and increasing values moving from the visible surface into free space, and negative and decreasing values on the non-visible side. The result of averaging the multiple SDF forms of the depth maps, aligned into a global frame, is a global surface fusion.

6.2.2 Truncation of the SDF

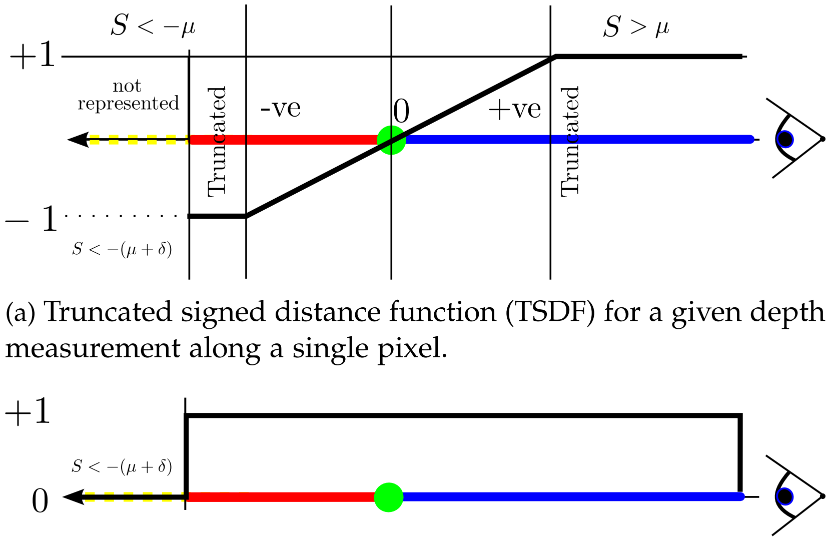
An example given in Figure (6.2) demonstrates how the TSDF allows us to represent arbitrary genus surfaces as zero crossings within the volume. We will denote the global TSDF that contains a fusion of the registered depth measurements from frames $1 \ldots k$ as $\mathbf{S}_k(p)$, where $p \in \mathbb{R}^3$ is a global frame point in the 3D volume to be reconstructed. A discretisation of the TSDF with a specified resolution is stored in global GPU memory where all processing will reside. From here on we assume a fixed bijective mapping between voxel/memory elements and the continuous TSDF representation and will refer only to the continuous TSDF $\mathbf{S}_k$. Two components are stored at each location of the TSDF: the current truncated signed distance value $S_k(p)$ and a weight $W_k(p)$:

$$\mathbf{S}_k(p) \mapsto [S_k(p), W_k(p)] \, . \tag{6.11}$$

We now discus each component of $\mathbf{S}_k(p$ further and provide their computation in Equations (6.12) and (6.16). A dense surface measurement (such as a raw depth map $\mathcal{D}_k$) provides two important constraints on the surface being reconstructed. First, assuming we can truncate the uncertainty of a depth measurement such that the true value lies within $\pm\mu$ of the measured value, then for a distance r from the camera center along each depth map ray for pixel $x \in \Omega$, $r < (\lambda \mathcal{D}_k(x) - \mu)$ is a measurement of free space (here $\lambda = \|\mathrm{K}^{-1}\dot{x}\|_2$ scales the measurement along the pixel ray). Second, we assume that no surface information is obtained in the reconstruction volume at $r > (\lambda \mathcal{D}_k(x) + \mu)$ along the camera ray. Therefore the SDF need only represent the region of uncertainty where the surface measurement exists $|r - \lambda \mathcal{D}_k(x)| \leq \mu$. A TSDF allows the asymmetry between free space, uncertain measurement and unknown areas to be represented. Points that are within visible space at distance greater than μ from the nearest surface interface are truncated to a maximum distance μ. Non-visible points farther than μ from the surface are not measured. Otherwise the SDF represents the distance to the nearest surface point.

Projective Approximation

Although efficient algorithms exist for computing the true discrete SDF for a given set of point measurements (complexity is linear in the the number of voxels), sophisticated implementations are required to achieve top performance on GPU hardware, without which real-time computation is not possible for a reasonable size volume. Instead, we use a projective truncated signed distance function that is readily computed and trivially parallelisable. For a raw depth map $\mathcal{D}_k$ with a known pose $\mathtt{T}_{g,k} \in SE_3$, its global frame projective



(a) Truncated signed distance function (TSDF) for a given depth measurement along a single pixel.

(b) Weighting function along the pixel ray.

Figure 6.3: Diagram of the truncated SDF functions critical regions used when computing the projective TSDF at each voxel in the volume (a). The computed TSDF approximation is integrated into the global TSDF using an iteratively computed weighted average using .(b) shows the basic weight function computed to representation regions which have a valid TSDF measurement (where $W_{\mathcal{D}_k} = 1$) or alternatively is not represented in the TSDF measurement ($W_{\mathcal{D}_k} = 0$)). We note that several factors are applied to the valid measurement weight, e.g. a per depth map pixel weight used to down-weight lower quality depth measurements.

TSDF $[S_{\mathcal{D}_k}, W_{\mathcal{D}_k}]$ at a point p in the global frame g is computed as:

$$S_{\mathcal{D}_k}(p) = \Psi\left(\mathcal{D}_k(x) - \lambda^{-1}\|(\mathbf{t}_{g,k} - p\|_2\right)\ , \tag{6.12}$$

$$\lambda = \|\mathrm{K}^{-1}\dot{x}\|_2\ , \tag{6.13}$$

$$x = \left\lfloor \pi\left(\mathrm{K}\mathrm{T}_{g,k}^{-1} p\right) \right\rfloor\ , \tag{6.14}$$

$$\Psi(\eta) = \begin{cases} \min\left(1, \frac{\eta}{\mu}\right) & \text{iff } \eta \geq -\mu \\ \mathit{null} & \mathit{otherwise} \end{cases} \tag{6.15}$$

We use a nearest neighbour lookup $\lfloor . \rfloor$ instead of interpolating the depth value, to prevent smearing of measurements at depth discontinuities. $\frac{1}{\lambda}$ converts the ray distance to p to a depth (we found no considerable difference in using SDF values computed using distances along the ray or along the optical axis). Ψ performs the SDF truncation. The truncation function is scaled to ensure that a surface measurement (zero crossing in the SDF) is represented by at least one non-truncated voxel value in the discretised volume either side of the surface. Also, the support is increased linearly with distance from the sensor center to correctly represent noisier measurements at large depths. Figure (6.3) illustrates the truncated signed distance function representation computed along a ray from each voxel to the camera center. A suitable measurement weight $W_{\mathcal{D}_k}(p)$ can be computed assuming the confidence of the depth estimate decreases with more obliquely viewed surfaces and with increased viewing distance using:

$$W_{\mathcal{D}_k}(p) \propto \frac{cos(\theta)}{\mathcal{D}_k(x)}\,, \tag{6.16}$$

where θ is the angle between the associated pixel ray direction and the surface normal measurement in the local frame. In Chapter (7) we will instead make use of the explicitly computed confidence obtained for the multiple view stereo depth maps from Chapter (5).

The projective TSDF measurement is only correct exactly at the surface $S_{\mathcal{D}_k}(p) = 0$ or if there is only a single point measurement in isolation. When a surface is present the closest point along a ray could be another surface point not on the ray, associated with the pixel in Equation (6.14). **Gibson et al.** (2000) showed that for points close to the surface, a correction can be applied by scaling the SDF by $cos(\theta)$. However, we have found that approximation within the truncation region for $100s$ or more fused TSDFs from multiple viewpoints (as performed here) converges towards an SDF with a pseudo-Euclidean metric that does not hinder mapping and tracking performance.

In Sections (6.1.4) we gave an overview of the volumetric depth map fusion developed by **Curless and Levoy** (1996), noting that an optimal manifold can be computed as the weighted average signed distance function, equivalent to the multiple input volumetric denoising solution obtained under an quadratic dataterm penalty, introduced in Section (1.4.3), which can be computed incrementally. Defined point-wise $\{p|S_k(p) \neq null\}$, the weighted average computation is:

$$S_k(p) = \frac{W_{k-1}(p)S_{k-1}(p) + W_{\mathcal{D}_k}(p)S_{\mathcal{D}_k}(p)}{W_{k-1}(p) + W_{\mathcal{D}_k}(p)} \tag{6.17}$$

$$W_k(p) = W_{k-1}(p) + W_{\mathcal{D}_k}(p) \tag{6.18}$$

No update on the global TSDF is performed for values resulting from unmeasurable regions specified in Equation (6.15) resulting in $W_k(p) = 0$. While $W_k(p)$ provides weighting of the TSDF proportional to the confidence of surface measurement, we have also found that in practice simply letting $W_{\mathcal{D}_k}(p) = 1$, resulting in a straightforward average, provides good results.

6.2.3 Moving Average Signed Distance Functions

By truncating the updated weight over some value W_η,

$$W_k(p) \leftarrow \min(W_{k-1}(p) + W_{\mathcal{D}_k}(p), W_\eta) \ , \tag{6.19}$$

a moving average surface reconstruction can be obtained enabling reconstruction in scenes with dynamic object motion.

6.2.4 Trivial Parallelism of SDF Fusion

Although a large number of voxels can be visited that will not project into the current image, the simplicity of the per voxel weighted average GPGPU kernel, means that operation time is memory bound, not computation bound, and with current GPU hardware over 65 gigavoxels/second ($\approx 2ms$ per full volume update for a 512^3 voxel reconstruction) can be updated. We use 16 bits per component in $\mathbf{S}(p)$, although experimentally we have verified that as few as 6 bits are required for the SDF value. We demonstrate the real-time performance of the technique as part of the complete KinectFusion dense SLAM system described in Chapter (9).

6.3 Predicting Geometric Measurements

With the most up-to-date reconstruction available with continuous surface fusion comes the ability to compute a dense surface prediction, by rendering the surface encoded in the zero level set $S_k = 0$ into a virtual camera. In the remaining sections of this chapter we describe mechanisms for computing a geometric and photometric prediction of the reconstructed scene.

6.3.1 Surface Prediction from Ray Casting the TSDF

As we have a dense surface reconstruction in the form of a global SDF, a per pixel raycast can be performed (**Parker et al.**, 1998). Each pixel's corresponding ray, $\mathtt{T}_{g,k}\mathtt{K}^{-1}\dot{x}$, is marched starting from the minimum depth for the pixel and stopping when a zero crossing ($+ve$ to $-ve$ for a visible surface) is found indicating the surface interface. Marching also stops if a $-ve$ to $+ve$ back face is found, or ultimately when exiting the working volume, both resulting in no surface measurement at the pixel x.

For points on or very close to the surface interface $S_k(p) = 0$ it is assumed that the gradient of the TSDF at p is orthogonal to the zero level set, and so the surface normal for the associated pixel x along which p was found can be computed directly from S_k using a

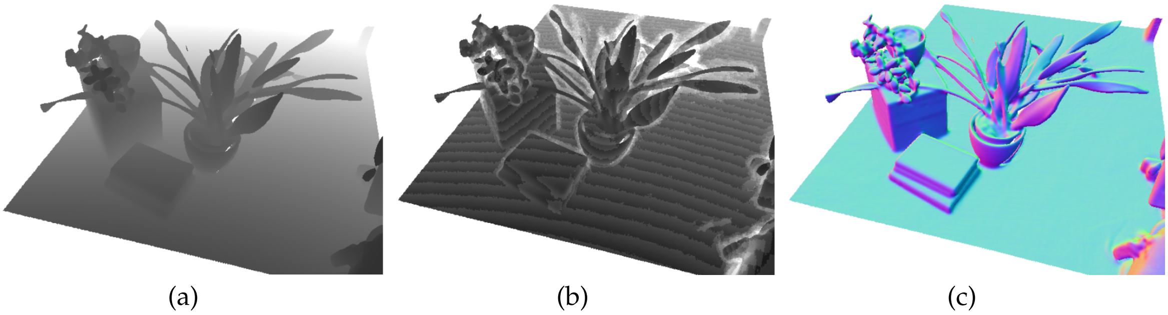


Figure 6.4: Demonstration of the space skipping ray casting. In (a) and (b) we render at each pixel the number of steps required in raycasting to obtain the surface intersection. In (a) for each pixel the ray is traversed in steps of at most one voxel (white equals 480 steps and black 60 steps). In (b) ray marching steps are drastically reduced by skipping empty space according to the minimum truncation μ (white equals 70 steps and black 10 steps, resulting in $\approx 7\times$ speedup). Step counts can be seen to increase around the surface interface in (b) where the signed distance function is *not* been truncated. (c) Normal map rendering of the resulting surface.

numerical derivative of the SDF:

$$\mathtt{R}_{g,k}\hat{N}_k(x) = \nu^*\left[\nabla S(p)\right], \tag{6.20}$$

$$\nabla S = \left[\frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}, \frac{\partial S}{\partial z}\right]^\top . \tag{6.21}$$

Here $\nu^*[.]$ scales elements of the gradient and normalises the vector to unit magnitude, ensuring correct isotropy for a given voxel resolution and reconstruction volume dimensions. Hence $\mathtt{R}_{g,k}\hat{N}_k(x)$ is the normal of the estimated surface in the global frame predicted into pixel x.

Since the reconstruction volume is fixed in resolution, the time for raycasting is bound by the maximum number of steps steps required to traverse a ray from the camera center to the volume boundary, or from the starting volume-ray intersection point when the camera center is outside of the reconstruction volume. Therefore, in contrast with explicit surface representation and rendering techniques, the maximum rendering time using raycasting is independent of the scene complexity.

Classically a min/max block acceleration structure **Parker et al.** (1998) can be used to speed up marching through empty space. However, in our scenario where surface rendering and surface fusion are interleaved at frame rate, the min/max macro block structure would also require continual updating. Instead we find that simple ray skipping provides a straightforward acceleration, exploiting the TSDF sparsity. In ray skipping we utilise the fact that near $S(p) = 0$ the fused volume holds a good approximation to the true signed distance

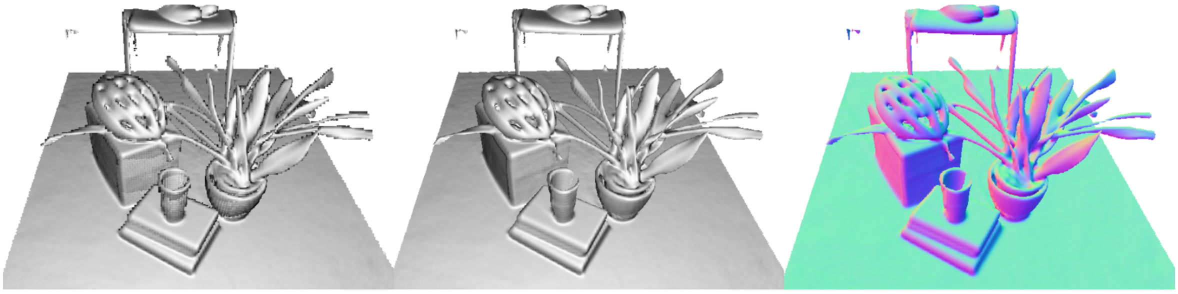


Figure 6.5: Rendering of a reconstructed scene with raycasting of the TSDF. In (b) the basic raycasting result is shown using trilinear interpolation of the TSDF function. In (b) we show the same ray-casting solution but using the analytical zero crossing intersection resulting in higher quality surface boundaries using Equation (6.22). The normal map rendering for the scene is shown in (c).

from p to the nearest surface interface. Using our known truncation distance we can march along the ray in steps with size $< \mu$ while values of $S(p)$ have $+ve$ truncated values, as we can assume a step μ must pass through at least one *non-truncated* $+ve$ value before stepping over the surface zero crossing. The speed-up obtained is demonstrated in Figure (6.4) by measuring the number of steps required for each pixel to intersect the surface relative to standard marching.

Higher quality intersections can be obtained by analytically solving a ray/trilinear cell intersection (**Parker et al.**, 1998) that requires the solution of a cubic polynomial. As this is expensive we use a simple approximation. Given that a ray has been found to intersect the SDF where S^+_{t} and $S^+_{\mathrm{t}+\Delta t}$ are trilinearly interpolated SDF values either side of the zero crossing at points along the ray t and $\mathrm{t} + \Delta t$ from its starting point, we find parameter t^* at which the intersection occurs more precisely:

$$\mathrm{t}^* = \mathrm{t} - \frac{\Delta t S^+_{\mathrm{t}}}{S^+_{\mathrm{t}+\Delta t} - S^+_{\mathrm{t}}} \,. \tag{6.22}$$

We compute the vertex location and associated surface normal using this interpolated location in the global frame. Figure (6.5) shows a typical reconstruction, the interpolation scheme described achieves higher quality occlusion boundaries at a fraction of the computational cost of fixed step ray-casting.

6.3.2 Surface Extraction with Tiled Marching Cubes

The memory and computational cost of raycasting within a fixed view frustum is a function of the resolution of the image being rendered into and the SDF resolution, but is constant

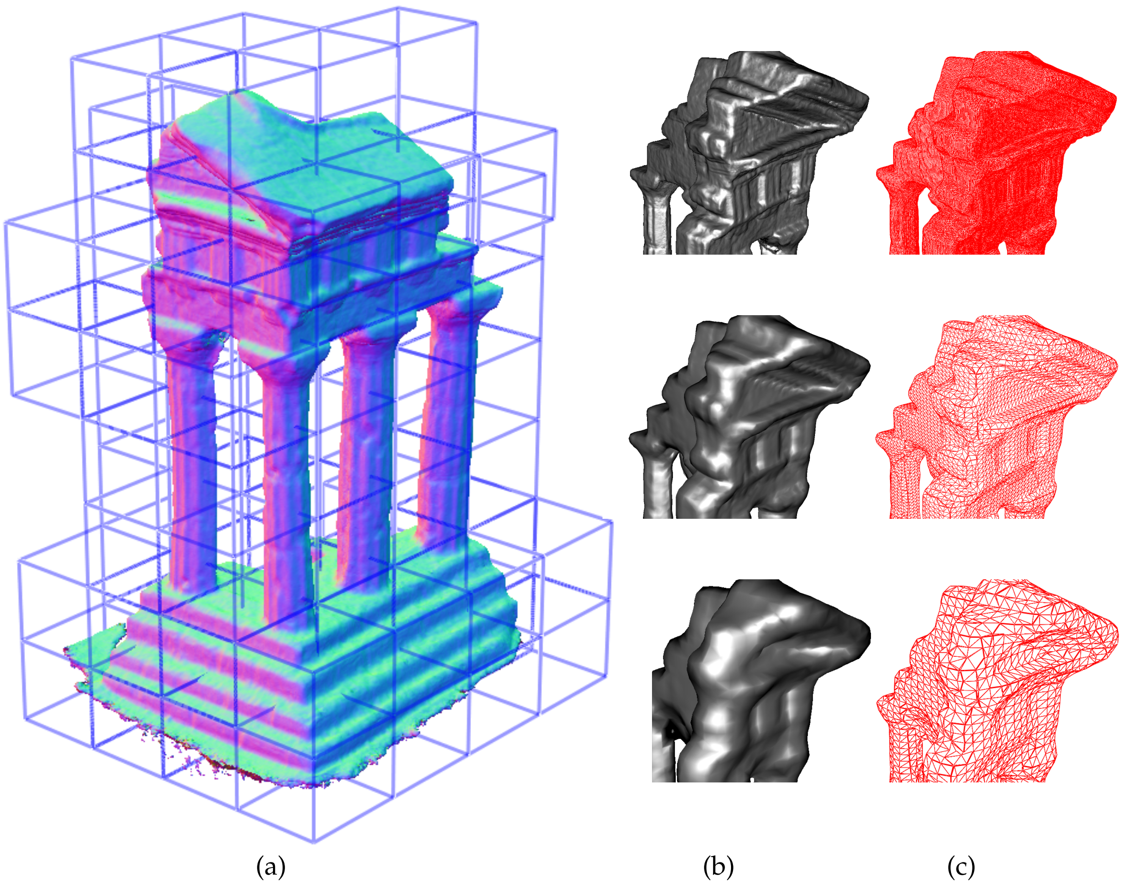


Figure 6.6: Illustration of the iso-surface extraction using a tiled marching cubes. For given tile in the volume, we detect if there is any possibility of a zero crossing in the sub-volume TSDF trivially by detecting blocks which contain only truncated or unrepresented SDF values. This results in a sub-set of tiles shown in (a) that are processed using the marching cubes pipeline. We perform detection and surface extraction for each tile sequentially. The resolution of a single tile can be chosen to fit the available memory resources on a given GPGPU platform while optimising for full occupancy of the computing resource. Rows in (b) show the resulting Phong shaded renderings for a close up view of the iso-surface extracted at three extraction resolutions: top to bottom shows full resolution at 512 × 256 × 256; 128 × 64 × 64; and 64 × 32 × 32 voxels. The TSDF used in this illustration results from dense reconstruction using the pipeline detailed in Chapter (7).

in the extracted model complexity, which ensures a constant maximum time operation surface fusion and prediction important for frame-rate dense SLAM. However, rendering is by definition obtained for a single view. If instead we are interested in extracting the full iso-surface from the current TSDF, for use in visualisation of the partial or complete model outside of the reconstruction pipeline, or to achieve a high rate of compression for parts of the scene which have been deemed fully reconstructed, we must instead perform a global extraction of the iso-surface.

A widely used approach to extracting a surface model in the form of a mesh from a signed distance function is to perform polygonisation of the desired level-set using the marching cubes algorithm (**Lorensen and Cline**, 1987). Marching cubes proceeds by taking each neighbouring 8 SDF values, forming the vertices of a virtual cube, and determining the planar face passing through the cube that best approximates the isosurface there. The importance of the marching cubes algorithm, and later corrections to it (**Nielson and Hamann**, 1991), was to reduce the space of possible face configurations passing through a cube to the deterministic set of 256 combinations, exploiting reflections and rotations, that can be tabulated and indexed based on the level-set sign computed at the cube vertices. Having computed the correct configuration, the intersection between the planar facet and the cube is computed using interpolation of the signed distance function. The marching cubes algorithm is a highly parallelisable, since classification of the face configuration can be performed independently within each local neighbourhood. Furthermore, the truncated signed distance function used in the surface representations here contain large homogeneous regions which further reduce the locations at which classification must be performed, which is easily detected in the initial stages of the algorithm. While it is possible to copy the SDF to host memory and perform all or part of marching cubes on the CPU, we instead fully utilise the parallel hardware, by tiling the marching cubes over sub volumes to extract the mesh incrementally. Figure (6.6) illustrates multiple resolution isosurface extraction of the zero level set which is possible simply by altering the spacing used between elements in each marching cubes.

Given the extracted mesh, fast view prediction is possible using a standard rasterising graphics pipeline. It is important to notice the advantages of the polygonisation based rendering pipeline in applications where the extracted mesh is reused multiple times before any surface update is performed. The computational cost associated with performing the initial tiled marching cubes extraction is higher than raycasting for the same resolution of extraction, but enables fast rendering of new views. We will exploit this property for passive image prediction in the next section where a large number of high quality geometry views must be obtained.

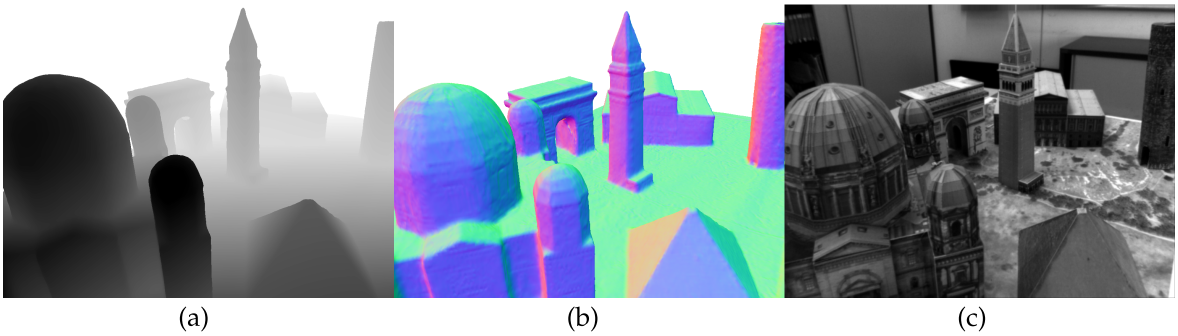


Figure 6.7: The geometric prediction computable from a dense surface representation provides an estimate of the scene depth (a) and surface normal (b) for every pixel in a given reference frame. In Section (6.4) we are interested in computing a photometric (or texture) greyscale or colour value for each pixel that predicts the appearance of the associated reference frame image (c), or any other novel view.

6.4 Predicting Photometric Measurements

Given the ability to render a geometric prediction, we now assume that for each frame in an input video we have available both a depth and surface normal estimate at every pixel in the associated photometrically calibrated image, shown in (Figure 6.7). In this section we are interested in rendering into a novel view a photometric prediction. Photo-realistic model rendering of novel views is a useful output in itself for mixed and augmented reality applications. Here we are further interested in obtaining the predicted appearance of a novel view for use in the dense camera tracking paradigm detailed in Chapter (8).

Traditionally, efficient rendering of a scene's photometric appearance was achieved by computing a static texture map for the surface geometry. Used within a rasterising graphics pipeline, the texture mapped surface results in a *view independent* prediction of the scene appearance. In Section (6.8) we will introduce a simple photometric fusion extension to the volumetric SDF integration process that allows incremental construction of a static surface texture using all available input video frames. Rendering of a novel view is then achieved using a basic extension to any of the isosurface extraction algorithms outlined in the previous section. Such static texture maps fail to capture a number of important aspects of the light-surface interaction in real scenes, including local illumination changes, inter-surface reflections and global illumination properties such as radiosity and shadowing. Two distinct paradigms have been researched that overcome such limitations: physically based rendering paradigms and image based rendering techniques.

Physically based rendering paradigms extend the modelled geometry to include accurate surface material and light source properties in the scene. Rendering is achieved by propagating light rays forwards in time starting from light sources, computing the complex interactions with the surfaces and generating further inter surface reflections, culminating in the interaction of reflected, refracted and emitted rays with the modelled camera lens forming an image (**Pharr and Humphreys**, 2004). Since only a small fraction of rays will end in intersection with the camera lens the most widely used physically based rendering systems approximate the process and reverse it by to performing ray-tracing; this process follows the path of one light ray for each pixel of the image to be generated *out* into the scene. The initial ray may give rise to a tree of several ray-surface interactions, following to some level of approximation the evolving paths of rays as they continue to interact with the scenes surfaces and lighting structures until a set number of ray-surface interactions has occurred.

Image based rendering (IBR) methods provide a efficient alternative to modelling the full physics in a scene and instead exploit the ability to interpolate the appearance of a novel view from captured images of the scene (**Szeliski**, 2010). At one end of the IBR spectrum, techniques use densely sampled images with associated poses that attempt to capture a sampling of all possible light rays in the scene, called the light-field (**Levoy and Hanrahan**, 1996). At the other end of the IBR spectrum, techniques make full use of available surface geometry to massively reduce the sampling space, performing view interpolation with a representative set of key-frame textures. In Section (6.4.2) we detail a key-frame based photometric prediction mechanism based on the view dependent texture mapping method developed by **Debevec et al.** (1996).

6.4.1 Photometric Fusion

An approach that elegantly extends the depth map fusion algorithm in Section (6.2), simply augments the SDF value and weight stored at each voxel with a colour parameter $C_k(p)$ and additional weight W_k^C:

$$\mathbf{S}_k(p) \mapsto [S_k(p), C_k(p), W_k^F(p), W_k^C(p)] \ . \tag{6.23}$$

Here we have denoted the previous weighting function for the TSDF as $W_k^F(p)$. We can then update the voxel colour for each surface measurement during signed distance function integration given a corresponding depth $\mathcal{D}_k$ and irradiance $\mathcal{I}_k$ image pair. During normal depth map fusion, each voxel p is projected into the sensor frame resulting in the pixel x, enabling computation of the SDF value $S_{\mathcal{D}_k}(p)$ given the depth value $\mathcal{D}_k(x)$. We extend the integration by picking up the colour value $\mathcal{I}_k(x)$, and compute a weighted average update, where the weighting function is:

$$W^C_{\mathcal{D}_k}(p) \propto \exp\left(-|S_{\mathcal{D}_k}(p)|\right) \ . \tag{6.24}$$

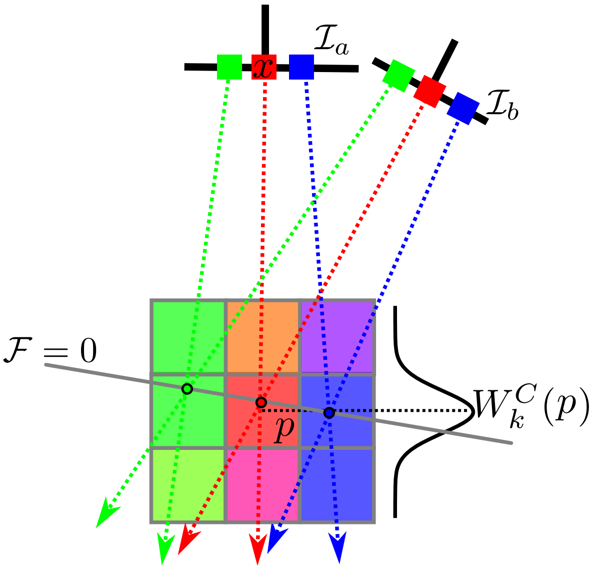


Figure 6.8: For lambertian surfaces, voxels located at the intersection of the rays have a correct colour average since they result from averaging corresponding pixels. The windowing function updates the weight W_k^C and reduces the influence of non-corresponding pixel values at neighbouring off-surface voxels that otherwise result in blurred, erroneous photometric predictions when extracting the colour values at interpolated isosurface locations.

This weight function decreases in value away from the currently estimate surface interface (where the SDF value is near 0). This is required to ensure the integration of the photometric data is local to the surface element.

As discussed in section (6.1.4), surface geometry is efficiently represented using the signed distance function through voxel interpolation. Interpolation of the colour values on the other hand results in blurred, lower quality predictions since the colour values stored in voxels neighbouring the zero crossing site contain a weighted average of *all* rays passing through the voxel, not only those that intersect the surface element being rendered, illustrated in Figure (6.8).

The photometric approach leads to doubling in memory requirements, and a small increase in computation proportional to the surface measurement area, since the window function can be safely truncated, avoiding small value updates. Rendering a photometric prediction is also very simple, since the colour values associated with the surface can be extracted along with the level-set using either raycasting or marching cubes.

Figure (6.9) provides a visual comparison between the photometric fusion based rendering and a real image masked using the geometry of the view shown in Figure (6.7). As can be

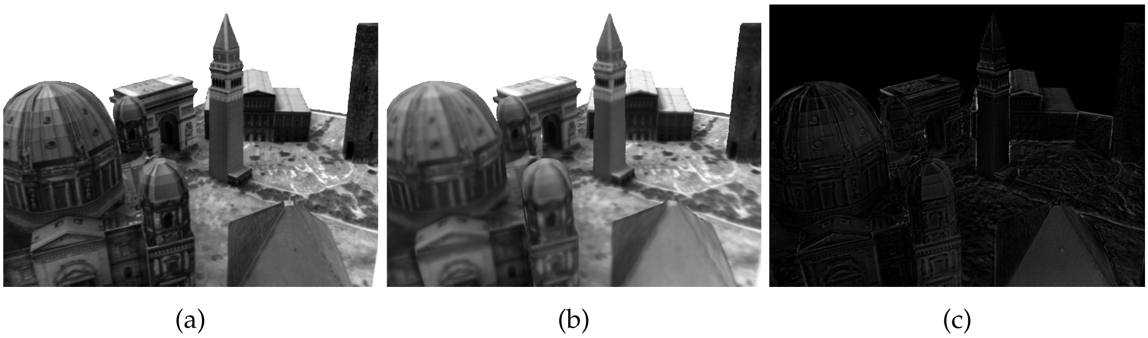


Figure 6.9: Photometric fusion: (a) Real Image masked to valid geometry. (b) Rendering of the grey value computed using photometric fusion at the corresponding isosurface location. (c) Absolute difference between (a) and (b) showing both high frequency image prediction error for the view and lower frequency global illumination errors. Errors result from errors in the geometry; the use of a photometric fusion volume with a lower sampling rate to the equivalent image space sampling rate; global illumination changes when observing the scene; and integration of observations over non-Lambertian surfaces.

seen, the higher frequency texture is smoothed away. This results from a lower sampling rate in the colour fusion volume than the image resolution and is alleviated when using a higher resolution colour fusion volume. Furthermore, integration of photometric data from observing non-Lambertian surfaces leads to an weighted average colour that is physically inconsistent with the views used in the integration. Finally, we note that by altering the photometric fusion to make use of the moving average mechanism previously introduced for the SDF integration in Equation (6.19), we can increase the local photo-consistency of the model when performing photometric fusion and prediction in a real-time moving camera setting, by essentially tracking the local illumination over time as the camera moves.

6.4.2 Key-Frame Textures

Debevec et al. (1996) introduced view dependent texture mapping (VDTM), a hybrid image based rendering technique that also makes use of explicitly available surface geometry. Unlike the photometric fusion approach and standard texture mapping techniques that construct a single texture map, VDTM uses the ability to interpolate a novel view from a sparse set of texturing keyframes. At the heart of VDTM the method of projective texturing enables the synthesis of a new view by warping a near keyframe texture via geometry predicted into the virtual camera.

Projective Textures

Given a depth map predicted into a novel view and a single key-frame texture co-observing the scene, a predicted appearance image can be obtained simply by projecting each of the

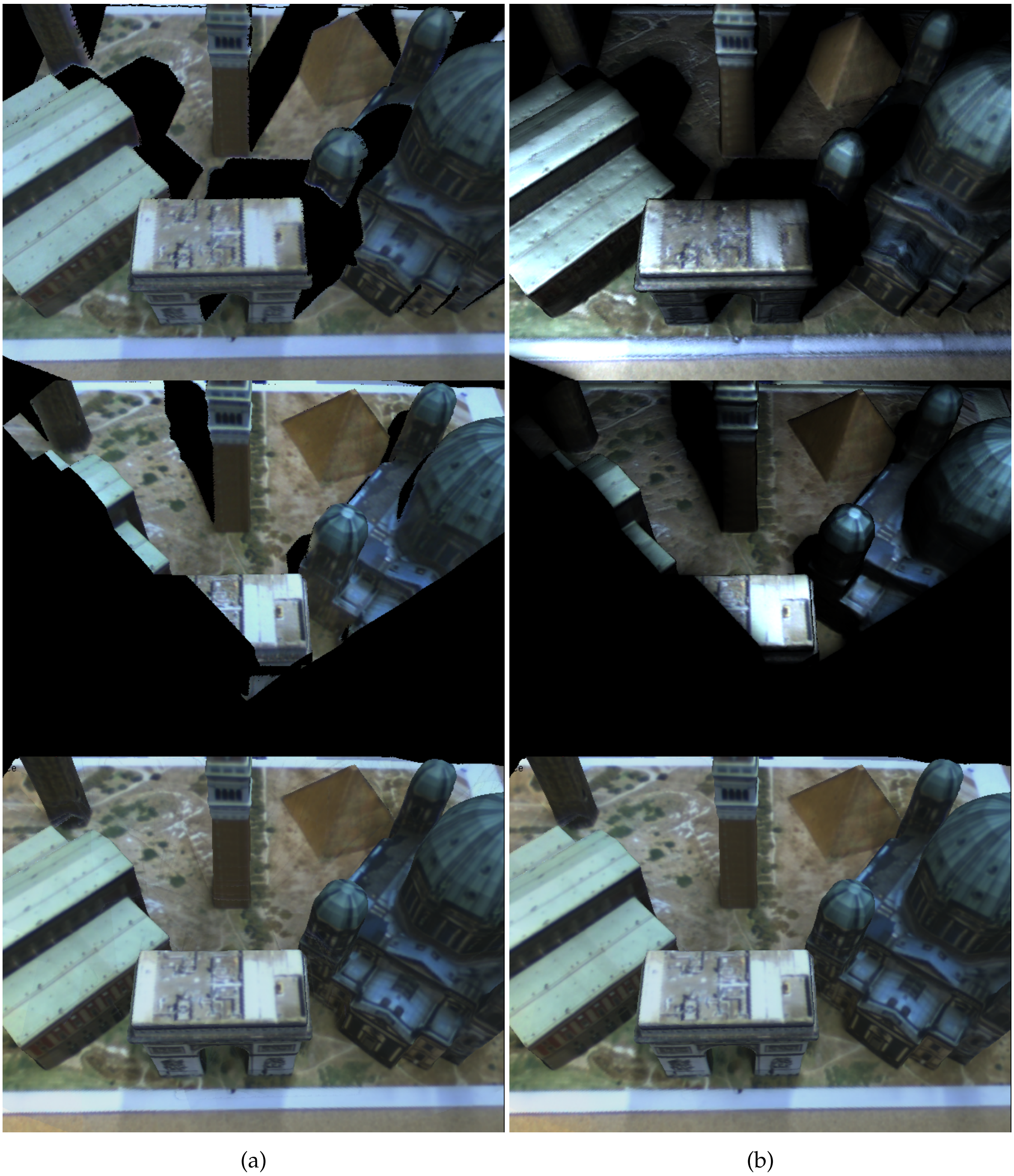


Figure 6.10: Projective textures from two views and the resulting view dependent texture mapped model. Row (a) without predictive texture weighting, (b) using the per pixel weighting functions. The top two views are averaged into the view synthesis. Imperfect surface geometry and camera calibration, together with view dependence for Lambertian surfaces result in seams forming in the synthetic view. It is recommend to zoom into the comparison views to view the reduced seams in the weighted average.

points in the depth map into the texture image, taking into account self occlusions. This can be thought of as turning the texturing keyframe into a slide projector, projecting its texture onto the global geometry. Rendering of the projectively textured scene results in the synthesised image, illustrated in Figure (6.10a).

By using more than a single texturing keyframe, regions in the novel view for regions that are not co-observed with a single keyframe can be covered. Synthesised pixels in the novel view will then typically have multiple projections from overlapping keyframes which are blended together using a weighting function.

To obtained a predicted view from a single keyframe we first predict the depth and normal maps (D_k, N_k) into the keyframe with pose T_{wk} and into the novel view (D_s, N_s) using the synthetic camera pose T_{ws}. We synthesise the pixel colour $\mathcal{I}_s(x)$ together with a pixel validity mask $V_p(x)$ in the novel view using the keyframe texture $\mathcal{I}_k$ via the projective warp function **w** (eqn. 4.1),

$$\mathcal{I}_s(x) = \begin{cases} \mathcal{I}_k(\mathbf{w}(x, k, D_s(x)), & \text{iff } V_{pk}(x) = 1 \\ 0, & \text{otherwise} \, , \end{cases} \tag{6.25}$$

where we have set $V_{pk}(x) = 1$ for co-visible geometry. $V_{pk}(x) = 0$ if there is no valid geometry in the predicted pixel, or if the surface geometry in the keyframe location is incompatible with the predicted view geometry due to occlusion, or if there is no valid geometry in the keyframe. Validity is trivially evaluated using a surface distance and normal similarity check.

View Dependent Texture Mapping

Debevec et al. (1996) showed that by dynamically changing the set of key-frames used to those nearest the virtual camera center and blending the resulting overlapping textures a higher quality appearance prediction is gained. In comparison to a single static texture, which is unable to capture the view dependent nature of of non-lambertian reflectance, the set of local keyframes can approximate the *local* surface-light interaction. Given a dense enough selection of texturing key-frames near to the desired novel view, the complex interaction between real world surface and lighting structures can be efficiently approximated.

Furthermore, VDTM and in particular real-time versions (**Debevec et al.**, 1998; **Porquet et al.**, 2005) are designed for use with simple efficiently rendered geometry proxies, which if classically textured mapped, result in large amounts of distortion and texture flattening. By choosing texturing key-frames with a similar camera center to the novel view, projective distortion is minimised. Finally, unlike the photometric fusion approach, IBR methods mitigate sampling and interpolation issues associated with a fixed volumetric representation of the surface colour by performing pixel transfer with interpolation in the image space at the original resolution of keyframe texture.

In the remaining sections of this chapter we will detail the three specific components required for online fully automatic texturing: Determining the subset of video frames that make up the texturing key-frame set; selection of a subset of keyframes for use in VDTM; and the composition function used to reduce photometric artefacts when blending the texture predictions in the novel view.

Inserting and Selecting Key-Frame Textures

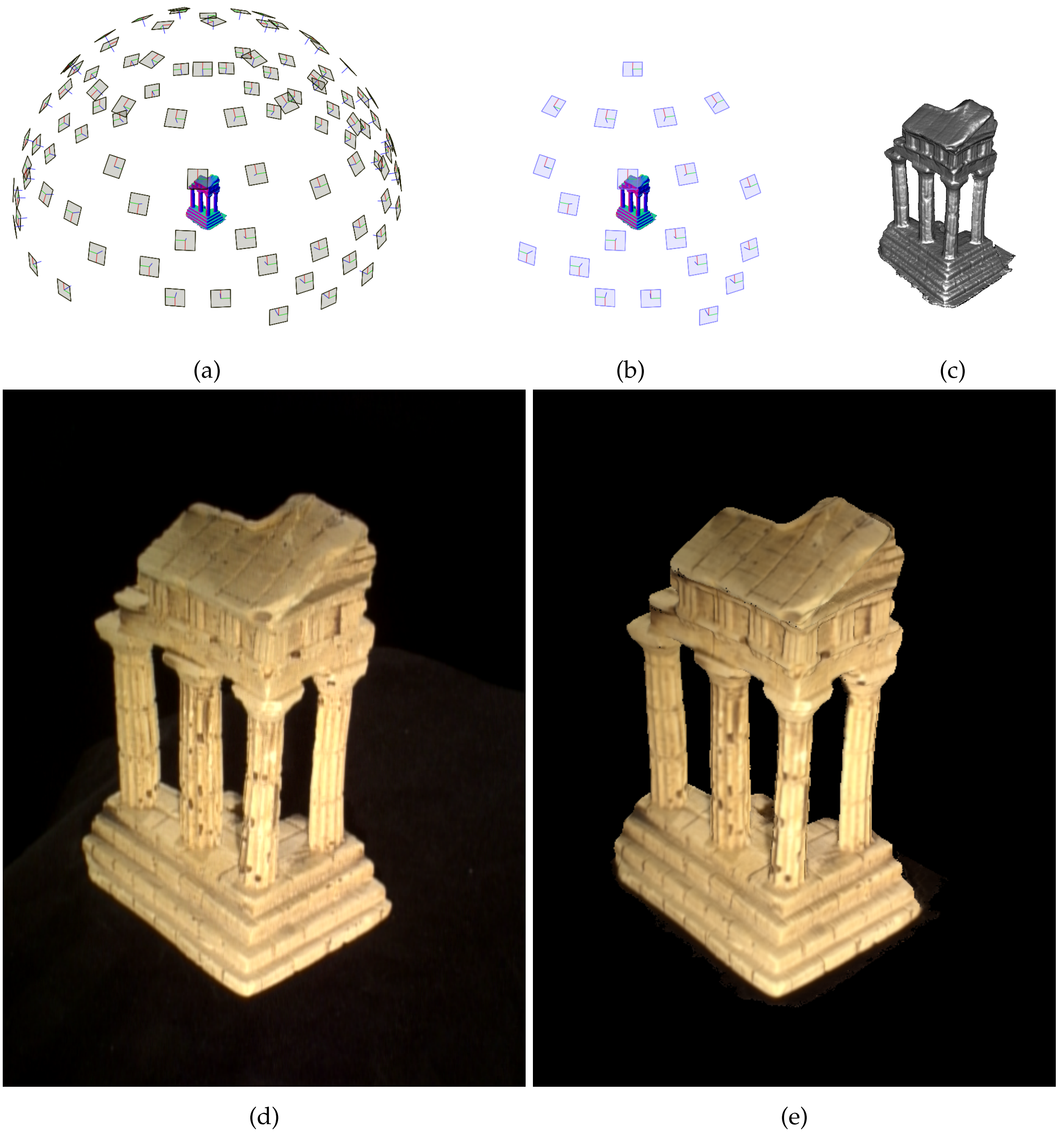
To build a VDTM model with live video we must decide online which frames should be selected from the video for inclusion in the texturing keyframe set $\mathcal{K}$. A second mechanism is required to select which key-frames are used in the view synthesis.

Insertion: The primary goal of a key-frame insertion mechanism is to decide if a new video frame should be inserted into a current key-frame set by some measure of the possible gain in predictive quality that would be accrued. **Klein and Murray** (2007) developed a simple but effective keyframe insertion mechanism for use in their sparse visual SLAM system to enable efficient bundle adjustment; simplify feature description and visual correspondence; and to facilitate relocalisation. Computing the closest keyframe to the new frame's camera center using a Euclidean distance, they attempt to distribute key-frames such that sampling density increases proportionally to the predicted distance to the observed scene. The camera center distance is therefore divided by the mean depth of features observed in the new frame and the frame is added if the scaled distance is over a given threshold. Unfortunately, insertion can become too conservative when the camera is moved near to a surface which has not successfully induced predictive features either due to texture homogeneity or visual correspondence failures due to perspective distortion.

The availability of dense surface prediction goes a long way to mitigating these problems and enables a more definitive specification of the key-frame insertion mechanism. We can simply replace the mean depth estimate obtained from the feature correspondences with the minimum depth computed in the dense surface prediction, which provides occlusion correct behaviour, independent of the image texture.

Selection: Given the key-frame set we take the nearest N keyframes using the scaled distance metric described above. Figures (6.11) and (6.13) demonstrate a key-frame set together with view selection.

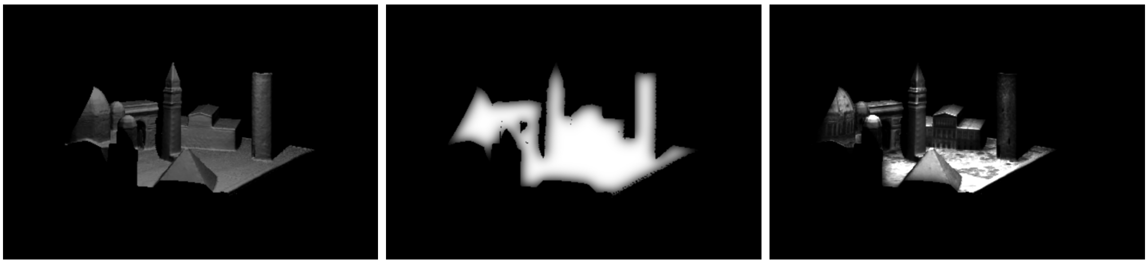
Given multiple projected textures from the selected key-frames, a number of factors hinder



(a) (b) (c)

(d) (e)

Figure 6.11: Photometric prediction for an image from the Middleburry temple data-set used in the model reconstruction from Chapter (7). The reconstructed model geometry is rendered with Phong Shading in (b). The set of key-frame textures showing a subset of the frames from the complete data-set (a). Key-frame selection using the nearest 19 key-frame views are shown in (b), for the reference view with ground truth image (d). The photometric prediction for this view is shown in (e). Note the reference image was excluded from the key-frame set for the view prediction.



(a) Surface visibility weight: v_{sk}

(b) Feathered pixel validity mask: b_{sk}

(c) View dependent texture for novel view s.

Figure 6.12: Example view dependent texture map generation. Weighting masks (a) and (b) multiply the projective texture predicted into the frame s from a key-frame k to produce the weighted texture contribution (c). Example view dependent texture predictions using several key-frames are given in Figures (6.10b) and (6.13).

simply averaging the predicted irradiance images together to produce the synthetic view. Inaccuracies in the underlying geometry and imperfect camera pose estimation result in incorrectly predicted occlusion boundaries, mapping non corresponding pixels together in the novel view. Non-Lambertian reflectance of surfaces can also result in large differences of brightness values on the occlusion boundaries.

View Composition and Visibility Weighting

To reduce the view composition artefacts we perform a weighted average over the selected key-frames to predict the view s. Illustrated in figure (6.12), we compute a per-pixel weighting $w_{sk}(x)$ to weight the predicted pixel value $\mathcal{I}_{sk}(x)$ obtained using key-frame k:

$$\begin{aligned} w_{sk}(x) &= V_{sk}(x) \cdot b_{sk}(x) \cdot v_{sk}(x) \cdot \|t_{ks} + c\|_2^{-1} && (6.26)\\ b_{sk}(x) &= (\mathcal{N}_\sigma * V_{sk})(x) && (6.27)\\ v_{sk}(x) &= \left\langle N_k(\mathbf{w}(x,k,D_p(x))), R_{ks}K^{-1}\dot{x} \right\rangle \; . && (6.28) \end{aligned}$$

The surface visibility weight v_{sk} at each pixel decreases proportionally to the angle between the pixel ray and the surface normal N_k in the key-frame texture, while b_{sk} down weights the texture component at depth discontinuities. Finally $\|t_{ks} + c\|_2^{-1}$ uses the distance between novel and texturing camera centres and acts over the whole image, decreasing the influence of the texture with increasing distance from the novel view, note c is a small constant to limit the influence of the a reference texture if the novel and keyframe camera centres are very close. The resulting photometric prediction is then obtained using a weighted average:

$$\mathcal{I}_p(x) = \frac{1}{\sum_{k\in\mathcal{K}} w_{sk}(x)} \sum_{k\in\mathcal{K}} w_{sk}(x)\mathcal{I}_{sk}(x) \; . \qquad (6.29)$$

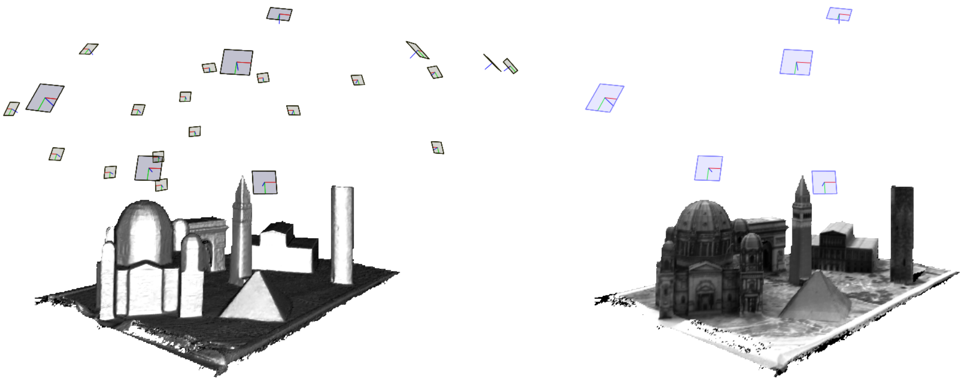


Figure 6.13: Example Key-frame Textures available for the reconstructed City of Sights model (a). View dependent texture mapping example of the model rendered using five key-frame textures (b). Details on the model reconstruction are given in Chapter (7).

Figure (6.10b) provides a visual comparison between the standard projective texture based VDTM and the per-pixel weighted version where seams in the predicted image due to imperfect camera calibration and reconstructed scene geometry are reduced. Figure (6.13) shows the resulting view prediction for the City of Sights model using five key-frames. Figure (6.11) illustrates the view dependent texturing on the Middleburry Temple dataset.

6.4.3 Remarks on Photometric Prediction

Despite the reduced quality of the photometric fusion algorithm, it is clear that in comparison to the key-frame based VDTM approach that it presents a simple and effective prediction mechanism that can use the overwhelming amount of image data produced from a video stream to mitigate the need to store and manage key-frames insertion and selection or require processing of textures to remove effects of geometric error. While it is true that in the wider context of photo-realistic rendering for novel view synthesis the key-frame based approach renders higher quality images, we must put into context the use of the prediction within the dense SLAM paradigm we are investigating. Within a live tracking and mapping application, if each new frame's image data can be utilised then the moving average photometric fusion will to some extent enable prediction of non-Lambertian surfaces simply by integrating over a local window of time. In Chapter (8) we will compare the two prediction approaches and show that the simpler mechanism is adequate for high quality camera tracking.

7

Incremental Surface Reconstruction from Video

Contents

Research into *multiview stereo* (MVS) has resulted in a multitude of techniques that, to some level of automation, infer the surface geometry from overlapping views from passive cameras. Recently, the possibility of using MVS reconstruction pipelines in a real-time setting has become increasingly realistic due to the massive increases in computational power and the availability of high-quality but affordable digital video cameras. The potential for such *live dense reconstruction* on commodity mobile computing platforms including smartphone and tablet computers offers an opportunity for numerous new applications, but also presents interesting new challenges that must be faced to enable high quality reconstruction outside of the research lab.

MVS pipelines have traditionally been developed to compute reconstructions from a sparse set of high resolution still images, where camera calibration is achieved to a high quality in an offline setting. Most importantly, the basic image sequences input to the system are fixed and no user feedback is assumed. This leads to difficulties in knowing if enough images with sufficient coverage of the scene were captured for successful reconstruction.

Live dense reconstruction (LDR) offers an alternative approach to model acquisition in an interactive setting. Here the dense reconstruction is obtained at an interactive rate providing *any-time* model output during the sequence capture. A user or robot can continue to evaluate the reconstruction and obtain more data where needed, in contrast to processing a fixed length input sequence and acquiring the model output only after all frames have been processed. This real-time not only puts pressure on algorithmic efficiency but also requires increased robustness to lower quality camera calibration and, when real-time video input is used, the system must cope with lower resolution noisier image data.

However, live dense reconstruction opens the door to applications using online model acquisition in augmented reality and holds great promise in model based robotics. As a perceptual layer, live dense reconstruction will be crucial for robots that must interact with, rather than avoid, their environment; providing a stepping stone to physically predictive models of the environment. For augmented reality to become truly immersive the correct rendering of artificial objects relies on the accurate reconstruction of surfaces and their discontinuities. And with the advent of 3D display and printing technologies live reconstruction will provide the ability to virtualise everyday scenes for gaming or to obtain prototypical input for modelling and home manufacture.

7.1 Chapter Outline

The *any-time* feature is the single property that distinguishes a live dense reconstruction system from multiple view stereo systems that work in a strongly offline mode. Research into dense reconstruction has seen an explosion of systems and techniques and it is therefore important to understand which MVS methods are most elegantly employed within an interactive setting. In Section (7.2) we review state-of-the-art methods developed for offline dense reconstruction and introduce the basic methodology with which they are evaluated. Specifically, we are interested in understanding the feasibility of the techniques within a real-time and incremental setting.

In Section (7.3) we then focus on the much smaller number of techniques with the potential for LDR usage that demonstrate the any-time or incremental reconstruction properties. In particular we detail relevant work from multi-view stereo and visual SLAM systems using the depth map fusion framework, which provides a modular reconstruction pipeline readily able to deal with real-time data input to produce models at interactive rates, with immediate feedback of the current dense reconstruction.

We detail our passive reconstruction pipeline in Section (7.4), producing a dense reconstruction in real-time using the depth map fusion framework described in Chapter (6). We use the ability to obtain a frame-rate prediction of the dense model geometry to reduce the computational burden of depth map estimation using the methods developed in Chapters (4) and (4).

In Section (7.5) we then look at the challenge of evaluating such LDR. We provide a qualitative comparison of reconstruction results for an offline statically captured MVS data-set, comparing the LDR pipeline to the state-of-the-art offline multi-view stereo methods, but importantly we also evaluate the system on a newly available video data set and compare the results obtained with the state-of-the-art LDR system developed by **Graber et al.** (2011).

We finish the chapter in Section (7.6) by summarising limitations of the approach developed in the broader context of large scale dense reconstruction and visual SLAM and look to towards research that addresses these issues.

7.2 Multi-View Stereo

The large number of reconstruction systems developed within both the computer vision and photogrametric research communities harness numerous techniques, surface representations and optimisation strategies. To enable comprehension of this growing body of work **Seitz et al.** (2006) developed a taxonomy of multi-view stereo techniques, their categorising of methods includes the scene representation used (many of which are discussed in Chapter 6); the photo-consistency and visibility models employed in stereo data terms (discussed in Section 4.2.2); the shape priors or regularisation frameworks used to ensure consistent surface reconstruction in the presence of ambiguous, missing and noisy data (discussed in the context of depth map estimation in Section (2.3) and Chapters 4 and 5); as well as a categorisation of the algorithm that uses these assumptions and components to achieve the dense reconstruction, along with the initialisation requirement of the algorithm. Abstracting from the taxonomy of **Seitz et al.** (2006) and extending to include newly developed approaches, we separate the algorithms into three broad classes.

Direct Optimization: In this class are methods which either implicitly or explicitly compute photometric cost volume given all views, followed by extraction of a surface from the volume which is consistent with prior assumptions about the solution smoothness. Within the class, methods differ in whether cost is computed in totality in one step followed by extraction of the most photo-consistent surface which can be achieved by global optimisation (**Vogiatzis et al.**, 2005); or whether the computation of photometric cost and surface extraction is interleaved, resulting in an iterative optimisation process. Examples include the voxel colouring (**Seitz and Dyer**, 1999) and space carving (**Kutulakos and Seitz**, 2000) approaches which dominated early dense reconstruction methods. In space carving, a reconstruction of the maximal surface consistent with a given photo-consistency measure, called the photo-hull, is computed. Within the reconstruction volume, all voxels are initially labelled as occupied. The algorithm then incrementally carves away visible voxels which are below a given photo-consistency. Another successfully developed methodology, related to space carving, uses a partial differential equation (PDE) based gradient directed optimisation to evolve a representation of the surface interface directly by minimising an energy functional consisting of a photo-consistency based data term and a regularisation term providing a prior over surface smoothness. Within this setting (**Pons et al.**, 2005) made use of a multi-scale optimisation framework to avoid getting stuck in local minima, while **Kolev et al.** (2009) presented a globally optimal convex optimisation solution. State of the art implementations utilising commodity parallel hardware have been developed to speed up the expensive photo-consistency computations in both space carving (**Zach et al.**, 2004), and PDE based optimisation (**Labatut et al.**, 2006b,a), demonstrating multiple orders of magnitude reductions in computation time over CPU implementations.

Surface Fusion and Fitting: In a second broad class, multiple views are used to compute local surface or point measurements which are there combined into a consistent surface reconstruction. In this class, the abstraction of the passive image data from image space into geometric measurements enables a more modular pipeline. Examples include sparse feature extraction and matching methods which are followed by surface fitting. These include the more sophisticated free-space carving techniques (**Hilton**, 2005; **Pan et al.**, 2009) discussed in Subsection (2.2.2). However, the dominating technique in this class modularises the reconstruction problem into a depth map fusion pipeline: first local depth maps are robustly estimated from subsets of the input sequence, essentially abstracting the passive image input to geometric surface estimates, and then these depth maps fused into a global surface model. Abstraction of images to surface measurements in the form of depth maps enables the surface reconstruction strategies based on occupancy mapping or volumetric SDF integration discussed in Chapter (6) to be used directly within the MVS pipeline (**Narayanan et al.**, 1998; **Koch et al.**, 1998; **Pollefeys et al.**, 2004; **Zach et al.**, 2006; **Goesele et al.**, 2006) , and can further use a global optimisation framework with surface smoothness priors (**Zach**, 2008; **Zach et al.**, 2007b; **Hernández et al.**, 2007). We review depth map fusion techniques in detail in the next section.

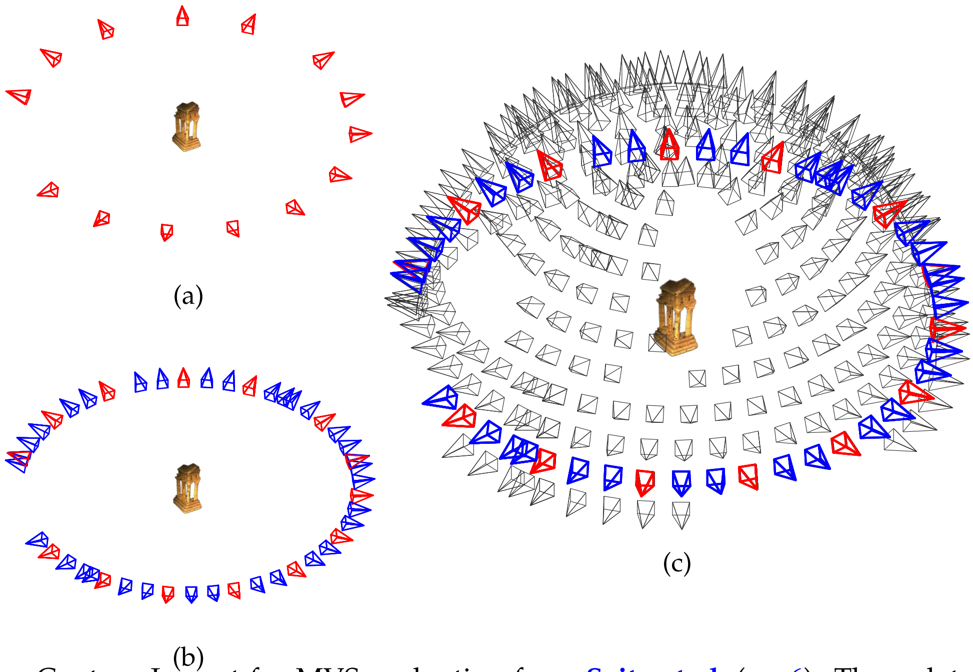
Region Growing: A further important class that does not sit neatly in either of the above classes but shares aspects of both exploits the surface patch (surfel) representation of surface geometry. Given an initial set of seed elements that can be obtained using sparse feature extraction and matching, these methods incrementally increase the surface coverage by propagating the current points into neighbouring uncovered pixels (**Otto and Chau**, 1989), locally optimising the surface normal and depth estimated at each new pixel. Since the surfel representation is free from a discrete volumetric representation, the region growing approach does not suffer from restrictions to the reconstruction volume. **Furukawa and Ponce** (2007); **Habbecke and Kobbelt** (2007); **Goesele et al.** (2007) obtained state of the art performance while the later authors also showed the efficiency of the surfel representation, using the approach to reconstruct entire buildings from community photo-collections. **Beljan et al.** (2011); **Chang et al.** (2011) further demonstrated that such region growing pipelines, while not as trivial to parallelise as the depth map fusion pipelines, can still benefit from GPGPU implementation.

Hybrid Pipelines Many of the most successful methods combine the surface fusion and fitting paradigm with a direct optimisation to increase reconstruction accuracy. Examples include **Vu et al.** (2009) who first construct a dense point cloud using feature detection and matching, from which they extract the coarse surface topology by computing a mesh over the point set consistent with induced visibility constraints. They then use a direct variational refinement of the surface mesh, minimising a regularised energy functional with a photometric error over clusters of co-visible frames.

Similarly, **Campbell et al.** (2008) combined a depth map fusion pipeline with a direct photo-consistency based optimisation of the fused surface. They compute local depth maps using the occlusion robust normalised cross correlation stereo measure (**Hernández and Schmitt**, 2004), and fuse these into a volumetric implicit surface, from which the iso-surface is extracted using a graph cut based global optimisation. The resulting surface is then directly optimised using the snake smoothing approach from (**Hernández and Schmitt**, 2004), making use of the full photo-consistency measure across co-observing views.

7.2.1 Evaluating Multi-View Stereo

To evaluate the growing variety of techniques, **Seitz et al.** (2006) also developed a multiple-view data set. Complementing the two-view Middleburry stereo benchmark (**Szeliski and Scharstein**, 2004), the dataset has ensured researchers have a common evaluation, enabling specific problems over classes of algorithms to emerge and advances from new algorithms to be highlighted. Two data sets are provided, consisting of a single reference object with known dense geometry acquired using an active stereo system, together with a set of calibrated image sequences captured on the Stanford light gantry, illustrated in Figure (7.1). Dense reconstructions are evaluated for accuracy by computing a $3D$ Euclidean distance



(a)

(b)

(c)

Figure 7.1: Capture Layout for MVS evaluation from **Seitz et al.** (2006). Three datasets are provided for two reference objects, shown here with the temple statue. The sparse datasets (a) have a total of 16 frames shown in red, with two redunant frames capturing the same view. The ring datasets (b) expand to include the blue frames totalling 47 frames. The full data set (c) comprising 312 frames also include all grey frames. Despite the potential advantages to be gained in reconstruction using the full dataset (c) over the sparse version (a), the majority of algorithms on Middleburry benchmark only show times for the sparse data set.

using a triangle mesh representation of the estimated and ground truth model. Moreover, measures of model completeness, and computation time lend to categorisation of MVS techniques as being more or less suitable for a given application with requirements on these qualities. Rapid developments in high resolution digital image capture, together with a desire to compare algorithms on real world scenes lead **Strecha et al.** (2008) to develop a further higher resolution dataset with ground truth geometry captured using LIDAR.

Exploiting Dense Image Sequences: An important commonality of both datasets discussed above and the majority of algorithms tested on them, is the relative sparsity of the image sequences. This is in stark contrast the high density of frames that can be acquired from digital video. It is particularly noteworthy that out of over 60 systems presented on the Middlebury multi-view stereo evaluation website, only 18 show a result for the dense sequence. This is despite the opportunity presented by the greatly increased view count (from 47 in the ring dataset in Figure (7.1a) to over 300 in the dense version shown in Figure (7.1c)). We note that algorithms evaluated on the dataset generally show an increase in reconstruction completeness and accuracy as a function of the number of views used, indicating the value of increased frame density. As we will discuss in Section (7.4.2), the availability of increased frame density need not lead to an increase in reconstruction time since dynamic view selection can be used to select the useful subset of frames for reconstructing regions of a model more optimally.

7.3 Live Dense Reconstruction

From those systems evaluated on the Middleburry datasets, only two demonstrate real-time, or interactive reconstruction capability: the mesh based depth map fusion approach from **Merrell et al.** (2007) discussed in Section (2.2.3) in the context of the dense visual SLAM system developed by **Pollefeys et al.** (2008); the second approach by **Zach** (2008) uses a volumetric signed distance function integration within a global convex optimisation framework that we will discuss in detail in subsection (7.3.2). Both methods utilise a form of incremental depth map estimation and fusion which is highly parallelisable, enabling implementation of the systems on commodity GPGPU hardware.

Beyond the middleburry dataset, in Section () we discussed the free-space carving LDR systems from **Pan et al.** (2009); **Lovi et al.** (2010), and note that the quality achievable using these systems is far below those using the depth map fusion approach, in large part due to the dependence on the sparse feature-based correspondences used to obtain free-space constraints.

We have also discussed the dense video rate point cloud estimation technique developed by **Vogiatzis and Hernández** (2011), that exploit the density of frames in video to obtain dense correspondences using feature-based detection and tracking. The system was later developed into LDR system (**Woodford et al.**, 2011), using camera pose estimation from PTAM. The results generated by this real-time system are promising when using high resolution motion blur free image data, however it is reasonable to believe that the reliance on sparse feature detection and tracking will result in sparsity of the point cloud reconstruction when presented with less accurate camera poses or motion blurred images.

Motivated by the quality of results produced by MVS systems that use depth map fusion, and the potential for optimisation of the modular approach it defines, we will now look in more detail at the systems that have used the depth map fusion pipeline in the multi-view stereo setting.

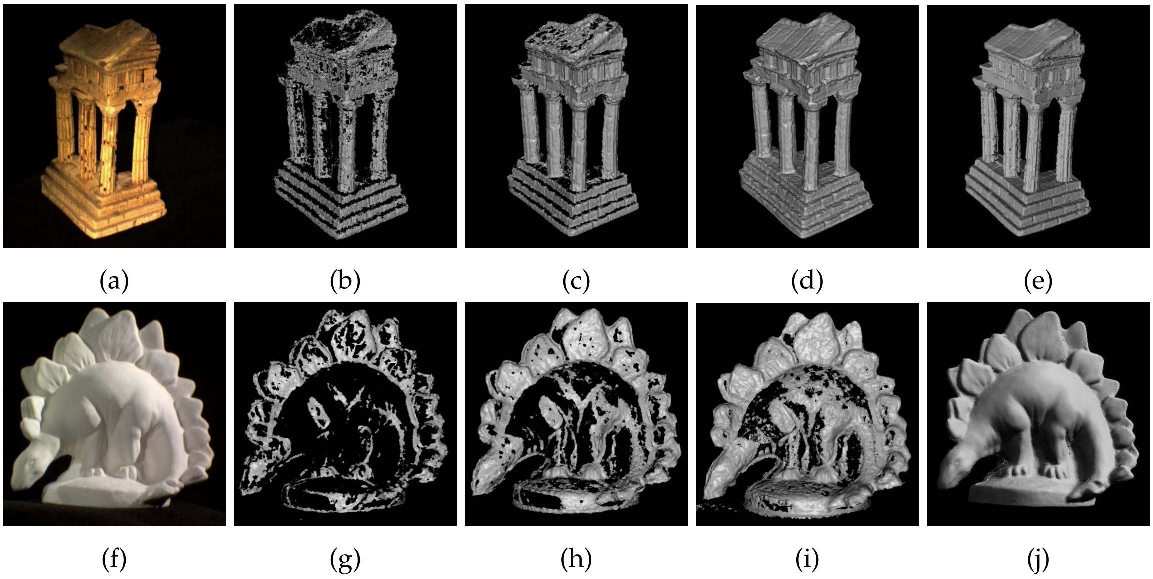
7.3.1 Volumetric Depth Map Fusion

In Chapter (6) we detailed the depth map fusion framework originally developed for reconstruction using active depth sensors (**Curless and Levoy**, 1996; **Hilton et al.**, 1996; **Wheeler et al.**, 1998). In this subsection we review the development of passive reconstruction systems which replaced active depth sensing with local depth map estimation. The earliest systems were not capable of real-time performance simply due to the computational restrictions of the time (**Koch et al.**, 1998) **Narayanan et al.** (1998); **Pollefeys et al.** (2004). Over time, the re-implementation, evaluation and extension of this pipeline by a number of researchers, taking advantage of the trivial parallelisability of the technique, has resulted in depth map fusion being at the core of top performing systems where incremental, high speed, reconstruction is an important goal (**Zach et al.**, 2006, 2007b, 2008; **Graber et al.**, 2011).

Narayanan et al. (1998) first demonstrated the use of a passive stereo based depth map fusion using the volumetric SDF representation in the context of dynamic scene capture. They used multiple cameras fixed in a hemispherical arrangement overlooking a finite volume in which reconstruction was performed. Dense depth maps were generated for a given synchronised time instant from each of the cameras using the multibaseline stereo method of **Okutomi and Kanade** (1993) and fused into a global surface reconstruction using the SDF integration method of **Curless and Levoy** (1996). The final surface for a given snapshot in time is recovered from the zero level set using the marching cubes method of **Lorensen and Cline** (1987).

Pollefeys et al. (2004) presented a complete visual modelling pipeline designed for image data captured from a hand held camera. They use feature-based structure from motion to obtain camera poses, followed by pair-wise dense stereo estimation with correspondence linking across multi frames to increase depth map accuracy, and to reason about occlusions. The dense depth maps are then fused into a global implicit surface model using the SDF integration approach.

Goesele et al. (2006) revisited the multi-view stereo problem amidst the numerous intricate techniques that had been developed in achieving state of the art results on the newly introduced dataset of **Seitz et al.** (2006). For each image in a dataset they compute a depth map using a subset of neighbouring frames using the occlusion robust stereo method introduced by **Hernández and Schmitt** (2004) with a 5×5 pixel patch normalised cross correlation based data cost. They refine the depth maps using an iterative interval search where the a new depth map is estimated by restricting the discretised epipolar search within a iteratively reduced neighbourhood of the previous depth estimate. They eliminate



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 7.2: Dense reconstruction from the Middleburry multi-view stereo datasets achieved by passive depth map fusion reproduced from **Goesele et al.** (2006). (a) shows a single frame from the temple dataset, (b) from the dinosaur dataset, which have a resolution of 640 × 480 pixels. Reconstruction results are shown for the temple sequence using the sparse (b), ring (c) and full (d) datasets with the vast range in frame sparsity shown in Figures (7.1a- 7.1c). Similarly, reconstructions are shown for the three levels of dataset density for the dinosaur model (g-h). The ground truth models for the rendered views are given in (e) and (j). The increasing level of reconstruction completeness and accuracy is clearly visible between the sparse and full datasets. Computation times (for sparse to dense datasets) using an Intel Pentium 4 class processor were approximately 11 minutes, 34 minutes and 4 hours.

depth values with low confidence and fuse these resulting conservatively estimated depth maps into a global surface model using the weighted SDF integration approach. This approach has both the *any-time* and incremental reconstruction properties since the newest model reconstruction is available directly after integration of a local depth map, both of which have a bounded constant computational complexity.

While often overlooked when evaluating the method in contrast to more sophisticated techniques, the results of local depth map fusion demonstrate the potential of dense overlapping input frames. Illustrated in Figure (7.2), their method shows drastic improvement in reconstruction as the density of input frames is increased. These results show that if the input data is of sufficient density and texture, and camera calibration is of a sufficient accuracy for local stereo estimation to succeed (stated for this dataset to be less than 1 pixel reprojection error), then high quality dense reconstruction can be achieved by a truly incremental pipeline.

The method is still one of the state of the art methods on the Middlebury dataset, as one of a select set of techniques on the benchmark website that have achieved reconstruction from the full dataset, Figure (7.1c). The processing time for the full sequence of 312 input images is stated at over 4 hours in 2006 on a single core Pentium 4 class process, and is often cited for this reason in intervening years, by competing methods, as comparatively inefficient. However, this pipeline has been shown to be eminently suitable for live or real-time operation, and as discussed in the remainder of this section, the trivial parallelisability and inherent modularity of the pipeline has enabled the full use of GPGPU hardware to obtain orders of magnitude increase in reconstruction throughput.

Zach et al. (2006) simultaneously developed an efficient GPGPU approximation of the depth map fusion pipeline of **Goesele et al.** (2006). Their multi-view stereo depth map computation included a 5×5 pixel patch based error score, which was truncated to provide implicit occlusion handling, and operating on mean normalised input images to improve robustness to illumination changes. They also exploit the temporal sequence of video input with the best half-sequence frame selection method originally proposed by **Kang et al.** (2001).

The *plane-sweep* multi-view depth estimation algorithm introduced by **Collins** (1996) is trivially parallelisable on GPGPU hardware and **Zach et al.** (2006) combined their robust version with an implementation of a GPGPU based volumetric SDF integration procedure to obtain a considerable speed up over the CPU depth map fusion version. The pipeline is discussed within an interactive processing framework, in which all depth maps are computed and then integrated into the volume, so their system does not demonstrate incremental dense reconstruction. However, they do achieve a processing rate of approximately 1 frame per second for their modified SDF integration approach. Depth maps were computed at a resolution of 512^2 pixels using a depth quantisation along each ray of 200 depth samples, and fused into a volume with a resolution of 256^3 voxels. As with previous implementations of the depth map fusion pipeline, the ability to utilise an incremental surface reconstruction is hampered by a requirement to preform iso-surface extraction using a CPU implementation of the marching cubes method. However, they enable interactive rendering of the iso-surface using real-time volume rendering (**Stegmaier et al.**, 2005).

7.3.2 Globally Optimal Depth Map Fusion

SDF integration using conservatively estimated depth maps results in an extremely simple and efficient MVS pipeline. It does however have two major drawbacks. First, since neither the depth map estimation nor the weighted SDF integration approach employ spatial regularisation, the resulting reconstruction can contain holes in regions with low texture, or if the signal to noise ratio is low due to sparsely overlapping views. This is vividly illustrated by the inability of the data only depth map fusion method to correctly reconstruct the low texture dinosaur model, even for the full dataset shown in Figure (7.2). Second, errors in the depth map become baked into the reconstruction. While the weighted average approach of **Curless and Levoy** (1996) is optimal under small amounts of Gaussian noise distributed along the depth measurement ray, depth maps estimated using multi-view stereo depth maps often contain non-Gaussian distributed errors.

Zach et al. (2007b) addressed both of these issues by posing dense reconstruction as a volumetric de-noising problem. Within their globally optimal range fusion framework, each estimated depth map is notionally transformed into a noisy volumetric measurement consisting of the signed distance f_i and weighting function w_i using the projective TSDF approximation (**Curless**, 1997). In the globally optimal approach, the desired volumetric signed distance function u is obtained as the solution to minimising a $3D$ convex denoising functional:

$$\int_{\Omega} \left\{ \|\nabla u(x)\|_1 + \lambda \sum_{i \in \mathcal{D}(x)} w_i(x) |u(x) - f_i(x)| \right\} dx \, . \tag{7.1}$$

In contrast to the weighted averaging approach, the ℓ_1 penalty over the data term provides resilience to outliers in the depth map in contrast to the non robust quadratic penalty which must rely instead on the explicitly performed down weighting of measurements to reduce their influence non-linearly. Furthermore, the total variation regularisation in Equation (7.1) enables areas with weak or missing data terms to be filled in. Importantly, since the functional is convex a gradient descent optimisation can be used to obtain the solution. **Zach et al.** (2007b) utilise a primal-dual formulation using the quadratic splitting technique resulting in a trivially parallelisable optimisation which is efficiently implemented on GPGPU (see Section (5.1) for an outline on the equivalent $2D$ splitting based primal-dual optimisation). They achieve a state of the art result evaluated on the Middleburry dataset with processing times on commodity GPGPU hardware, which at less than two minutes for the ring datasets, are orders of magnitude faster than the next best MVS method evaluated.

Recently **Schroers et al.** (2012) investigated the globally optimal range fusion approach using anisoptropic regularisation and replacing the projective TSDF with a more accurate approximation to the true Euclidean signed distance function demonstrating a small increase in quality for objects with a low quality data term. Unfortunately their improvement on the original technique leads to slower processing, requiring over one hour on the temple dataset.

The globally optimal TV-ℓ_1 range fusion achieves both robustness to outliers in the surface measurements and enables water tight reconstruction of models despite missing data in depth maps. However, in contrast to the weighted average depth map fusion approach that does not require the depth map to be stored after it has been integrated, the summation over all depth maps in the TSDF form under the ℓ_1 penalty requires revisiting the data term in every iteration of a gradient descent on Equation (7.1). It is for this reason that **Zach et al.** (2007a) was limited to evaluation on the reduced ring data sets, since global memory needed to store the measurements was limited to the maximum memory of commodity graphics cards at the time of the evaluation.

Zach (2008) introduced the novel TV-hist formulation removing this limitation by replacing the data term in Equation (7.1) with an ℓ_1 penalty over a histogram of discretised signed distance function measurements recorded at every voxel:

$$\int_\Omega \left\{ \|\nabla u(x)\|_1 + \lambda \sum_{q \in \mathcal{Q}} w_q(x) |u(x) - q)| \right\} dx \, . \tag{7.2}$$

Here $\mathcal{Q}$ is the set of quantised SDF function values and the weight function w_q stores the frequency of observing a measurement q at the specified voxel. With the incrementally updatable histogram representation, the data term has a constant memory requirement and the resulting globally optimisable energy functional has a computational cost for optimisation that is independent of the number of input depth maps.

Graber et al. (2011) further improved on the method by replacing the quadratic splitting method in the original primal-dual approach with the first-order method of **Pock and Chambolle** (2011), halving the required memory used for the gradient descent optimisation. Furthermore, they integrated the method into a full live dense reconstruction system. The real-time visual SLAM system by **Klein and Murray** (2007) provides camera pose estimation along with a set of bundle adjusted key-frames. By setting each new key-frame as a depth map reference frame, and estimating a depth map using plane-sweep with a fixed set of neighbouring key-frames, their pipeline interleaves integration of new surface measurements into the histogram data term with a fixed number of iterations of the primal-dual optimisation on the the regularised energy.

Most recently **Wendel et al.** (2012) demonstrated the system in use in a real-time distributed processing context, enabling dense reconstruction of indoor and outdoor scenes by a micro air vehicle. They also added a TV-ℓ_1 depth map denoising stage to the depth map estimated pipeline, to increase resilience to noisy image data observed in data acquired from a real-time moving camera. The hand held camera version of their live dense reconstruction system (**Graber et al.**, 2011), has also been demonstrated live during a recent workshop on the subject (**Newcombe et al.**, 2011a), where the dense reconstruction approach detailed in this chapter was also demonstrated. We briefly compare the results on a live dense reconstruction data set at the end of the chapter.

7.4 Passive Reconstruction Pipeline

The volumetric TSDF reconstruction method described in Chapter (6) enables assimilation of the massive number of surface measurements that can be estimated in each frame of a video. Moreover, due to the density of video frames, a given surface patch can be observed multiple times before leaving the camera frustum or becoming occluded. Since each frame is integrated into the TSDF at frame-rate the best global surface estimate is available to predict the next surface view given a known camera pose. In this section, we now detail our live dense reconstruction pipeline that exploits this ability to predict and update the current globally consistent surface at frame-rate.

7.4.1 Pipeline Overview

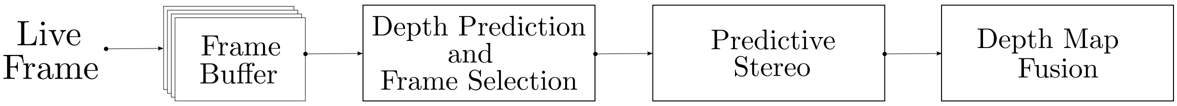


Figure 7.3: **Pipeline Overview:** The passive reconstruction pipeline broken into its main three components. A new reference frame is selected as the next available frame from a temporal image buffer and is then processed into a depth map using a selection of nearby frames also within the buffer. The depth map is then integrated using weighted TSDF fusion into the global model. Our reconstruction pipeline runs entirely on GPGPU hardware.

We discuss the pipeline, consisting of the three components outlined in Figure (7.3), from the point at which a new frame furnished with an estimated pose, has become available. In practice, real-time tracking mechanisms including the PTAM feature based tracker used in the real-time experiments in this section and the dense tracker detailed in the following Chapter (8), provide a quality of estimation output associated with an estimated pose. Based on a threshold of tracking quality we therefore neglect to process frames with poor quality pose estimate, e.g. when tracking is lost in PTAM and re-localisation is required (**Klein and Murray**, 2008).

The aim of the passive reconstruction pipeline is to compute at frame rate a new depth map using the multiple views available within a temporal window of the live frame, and to fuse that depth map into a global model using the volumetric TSDF representation detailed in Section (6.2). Importantly, to take advantage of the multiple-view stereo techniques discussed in Chapters (4 and 5) we do not compute a depth map for the live frame but instead use the frame at the center of a temporal buffer. To that end, the pipeline begins with selecting the next available reference frame and proceeds in three stages: (**1**) A surface prediction is computed into the reference frame from the current global model. This is used to guide a *frame selection mechanism* detailed in Section (7.4.2) that picks suitable frames neighbouring the reference for the proceeding MVS pipeline. (**2**) A depth map is then estimated for the reference frame using the selected multiple views. Depth map estimation also makes use of the per-pixel prediction from the current model and is detailed in Section (7.4.3). (**3**) The estimated depth map is then fused into the global model (subsection 7.4.4).

7.4.2 Depth Prediction and Frame Selection

Incremental Reconstruction

All incremental modelling pipelines must successfully address the challenge of utilising a finite memory and processing resource while dealing with a potentially endless stream of input images. Dense modelling using a depth map fusion pipeline, in particular, provides a clear opportunity for open-ended incremental reconstruction, by ensuring that both the acquisition and fusion of depth maps occurs using constant memory storage, and with a maximum processing and memory bandwidth ensures a timely reconstruction that is useful for the application it is serving.

To achieve this, a pipeline must make a basic trade-off. Given finite resources, and relative to the frame rate of image capture, either a lower rate of higher quality depth maps can be computed and fused into the reconstruction or a higher rate of lower quality depth maps can be utilised. The computation time of a depth map is a function of the depth map resolution and the number of pixels in total used in the optimisation, computed as the resolution of input images times the number of frames used in the estimation. We can therefore realise the trade-off by making a selection over those variables of image resolution and quantity of neighbouring frames used in the estimation of each depth map, together with selection of the input frames into which a depth map is estimated.

View Selection

Within the multi-view depth map estimation pipeline from Chapters (4) and (5), the relationship between the number and of views used in stereo estimation and its computational cost is acute. Given a reference frame for which a depth map is to be computed, neighbouring view selection is therefore a critical component in ensuring quality live incremental operation. Video acquired using a moving camera routinely contains frames with motion blur, image defocus, calibration errors and effects from dynamic illumination sources and non-Lambertian surfaces which together with image noise make some frames less useful than others. Figure (7.4) illustrates the resulting depth maps estimated for a single refer-

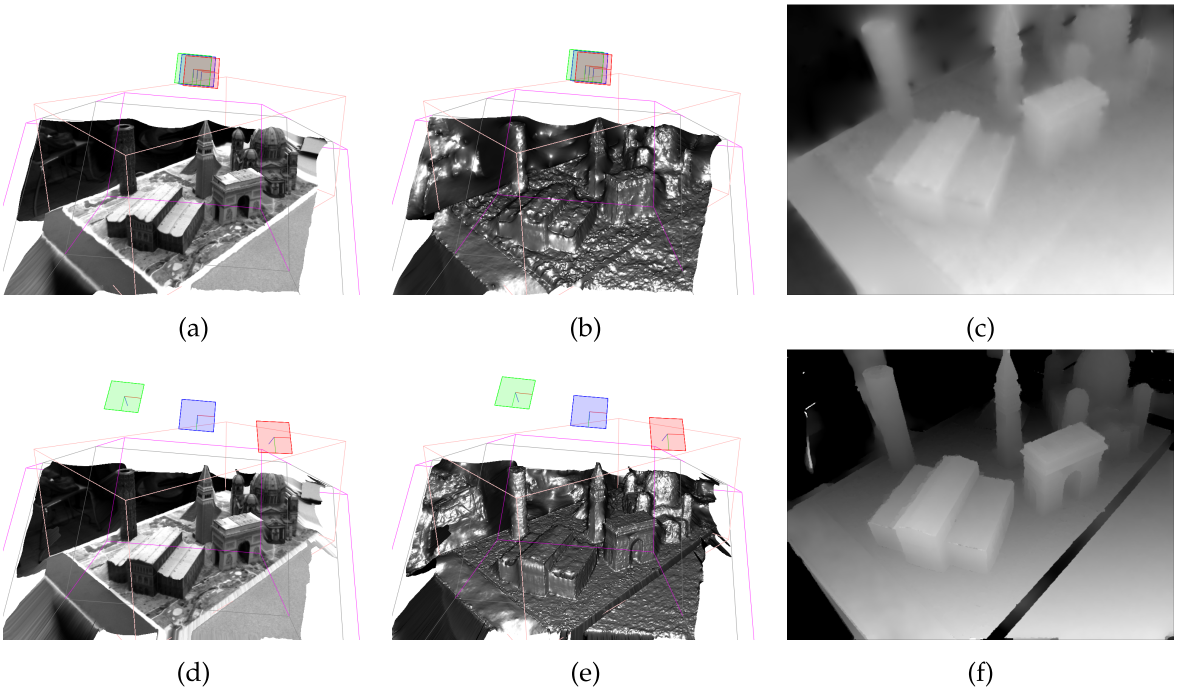


Figure 7.4: Demonstration of the difference in depth map quality when using very narrow and small baseline frames. Depth maps are computed using the multi-view stereo depth estimation pipeline developed in Chapter (5). In (a-c) temporally consecutive frames from the Graz City of Sights video dataset are used in the estimation of a depth map for the central (blue) reference frame: showing a texture mapped mesh of the reconstructed depth map with relative frame poses (a), Phong shaded mesh (b) and the depth map (c). In (d-f) the same reference frame is used with more widely space frames leading to increases in depth map quality for the surface observed. We note however that the baseline can not be increased indefinitely due to the simplifying assumptions made in the patch based stereo data term.

ence frame using two different sets of neighbouring frames using the full multi-view stereo pipeline developed in Chapter (5). In the example, the increased baseline of the second set results in drastically improved geometry estimation given the same processing resource but requiring a larger temporal buffer of frames.

Hornung et al. (2008) developed the first fully automatic view selection mechanism specifically for use in off-line MVS where the input image sequence is fixed in advance, noting the interplay between modelling quality and processing times of reconstruction pipelines. They perform a trade-off between using a greater number of observations of a scene which enables the reconstruction of fine detail and deep concavities with reducing processing times by discarding frames with redundant information. They adapted a greedy next best view selection framework (**Scott et al.**, 2003), which first builds a coarse geometry proxy from all views using a basic occupancy grid mapping approach, and then proceeds to incrementally add views to an active set for use in higher quality multi-view stereo. Views are added by reasoning about the reduction in photo-metric error that could be achieved by including the view. Using the view selection as a pre-processing step on a given input dataset, they demonstrated reduced processing times with equivalent or even improved reconstruction accuracy on the full Middleburry multi-view datasets for several MVS pipelines including the feature matching and patch expansion based MVS (**Furukawa and Ponce**, 2007), surface growing (**Habbecke and Kobbelt**, 2007), deformable models (**Hernández and Schmitt**, 2004), and volumetric graph-cuts **Hornung et al.** (2008).

More recently **Hoppe et al.** (2012) developed an online feedback mechanism for high quality structure from motion in which a sparse point cloud produced from the current SFM result is used to build a coarse mesh-based geometry proxy over which a surface visibility measure can be computed. The result is a low quality reconstruction, updated in an interactive manner that enables a user to decide if there are sufficient frames captured to achieved the desired level of reconstruction in an offline setting.

Using Video Rate Input

In general, optimal view selection for any quality of metric is a combinatorial problem and is necessarily sub-optimal given only a finite set of frames to be processed in an incremental fashion. Working instead in an on-line MVS estimation setting, **Gallup et al.** (2008) recognised the usefulness of dense video frame input over the comparatively sparse frames used in offline reconstruction pipelines. Noting again the differing quality of reconstructions in Figure (7.4), the increased error in a stereo pair can be expressed as a result of uncertainty in camera calibration and image measurements that propagates into error in pixel correspondence ϵ_d leading to error in depth ϵ_z:

$$\epsilon_z = \frac{z^2}{bf} \cdot \epsilon_d \,. \tag{7.3}$$

Here z is the surface depth, and b and f are the baseline and camera focal length. **Gallup et al.** (2008) demonstrated that a constant error in depth can be achieved by exploiting the density of video to set b dynamically. Video rate data enables setting of b by selecting appropriately spaced neighbouring images relative to a given reference frame.

Predictive Frame Selection

We combine the variable baseline methodology with the predictive capability of an incrementally reconstructed model to dynamically select wider or shorter baseline frames for use in the depth map denoising framework. By computing depth maps for *every* input frame we mitigate the problem of depth map frame selection, and focus instead on exploit-

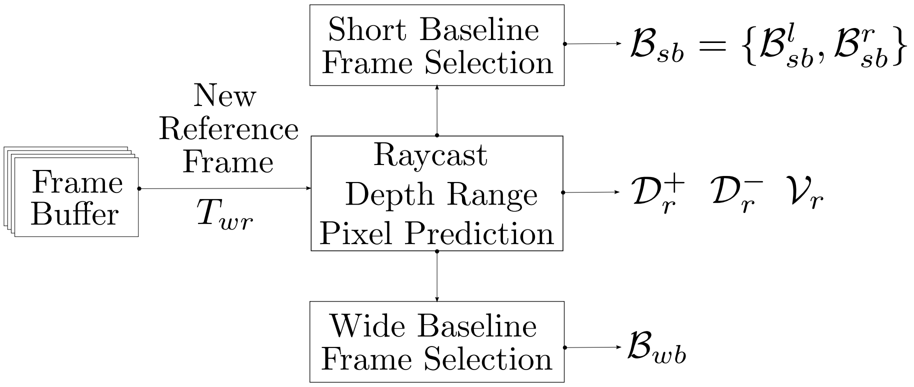

Short Baseline Frame Selection
$\mathcal{B}_{sb} = \{\mathcal{B}^l_{sb}, \mathcal{B}^r_{sb}\}$
New Reference Frame
T_{wr}
Frame Buffer
Raycast Depth Range Pixel Prediction
$\mathcal{D}^+_r$ $\mathcal{D}^-_r$ $\mathcal{V}_r$
Wide Baseline Frame Selection
$\mathcal{B}_{wb}$

Figure 7.5: **Depth Prediction and Frame Selection**: Depth map prediction into reference frame T_{wr}, and Image Selection from rolling frame buffer.

ing the redundancy in the video rate input, to obtain high quality estimation under the constraint of frame-rate operation. The depth prediction and frame selection mechanism is outlined in Figure (7.5).

The *frame buffer* stores a rolling buffer of the last N_{max} frames updated with each live frame. The pipeline proceeds to select the frame at buffer location $\frac{N_{max}}{2}$ as the next frame into which a depth map is computed, referred to throughout this chapter as the reference frame with pose T_{wr}.

A surface geometry prediction is then computed from the current dense reconstruction into a virtual frame with pose T_{wr}. We compute a predicted range over the depth at each pixel in the reference frame in the form of two depth maps, $\mathcal{D}^+_r$ and $\mathcal{D}^+_r$, providing a per-pixel estimate of the minimum and maximum depth given the current reconstruction. A prediction validity mask $\mathcal{V}_r$ is also computed, where $\mathcal{V}_r(u) = 1$ if the depth prediction for pixel u is valid; otherwise $\mathcal{V}_r(u) = 0$. Details for computing $\mathcal{D}^{+/-}_r$ and $\mathcal{V}_r$ are given in the next subsection.

Taking into account the extent of the reconstruction volume viewed by the reference frame, if an estimate for a valid depth range is successfully produced into more than half of the predictable reference frame pixels, we compute the minimum and maximum predicted scene depth, $\{d_{min}, d_{max}\}$ from $\mathcal{D}^{+/-}_r$. Using this range we select two subsets of frames from the temporal frame buffer for use in multi-view depth map estimation.

The first subset, $\mathcal{B}_{sb}$, will be used in a depth map denoising pipeline. Given the minimum depth expected to be observed in the reconstructing frame, we select the *n closest* frames that are within a scaled Euclidean distance $\delta_{\mathcal{B}_{max}}/d_{min}$ of the reference camera and are greater than $\delta_{\mathcal{B}_{min}}/d_{min}$ apart from each other. Scaling the maximum threshold by the distance to the predicted minimum ensures that a minimum visual angle to the closest

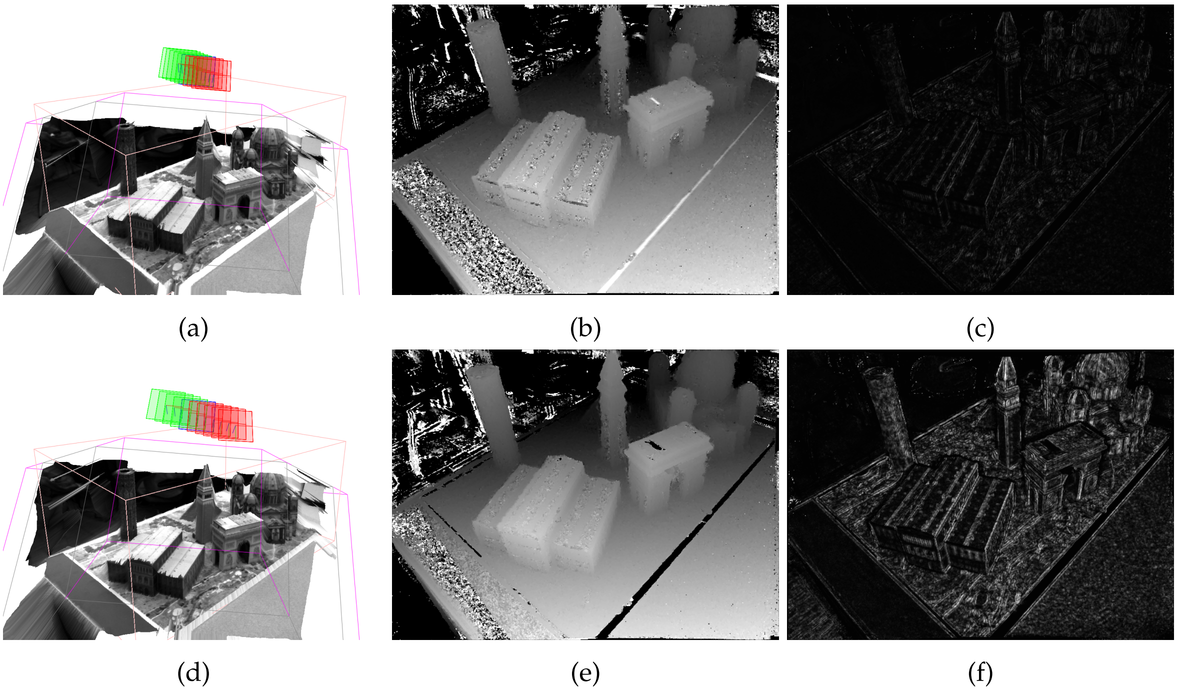
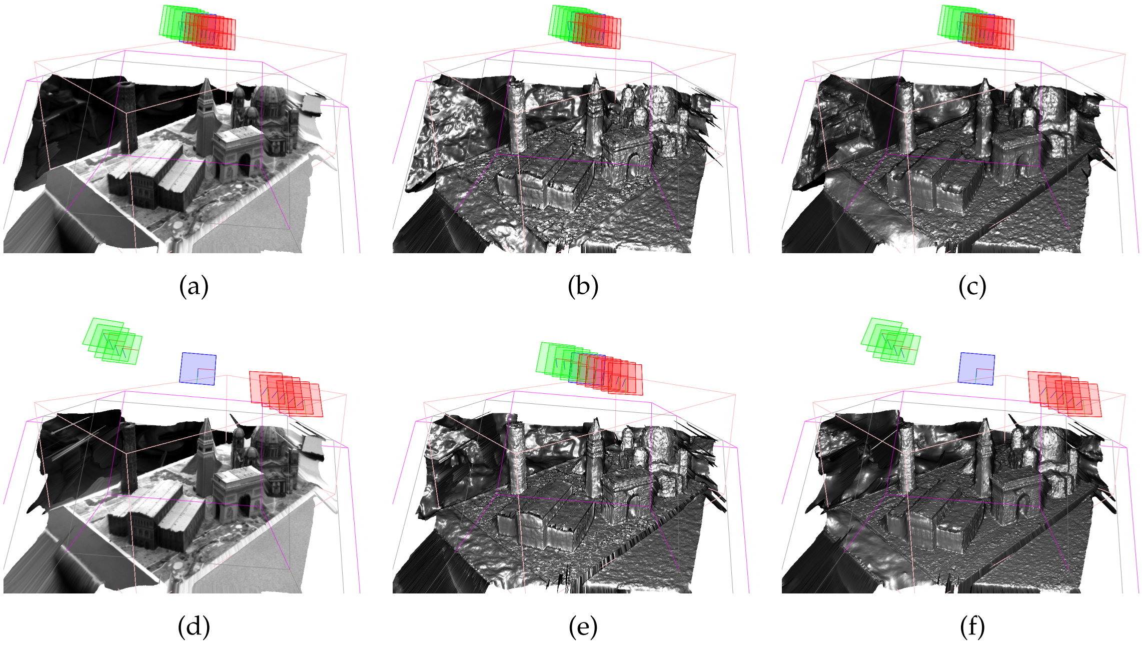


Figure 7.6: Short baseline view selection mechanism: (a) Frame selection using closest temporal views. (b) Resulting depth map data term minimum using frame selection from (a) and associated confidence measure (c). (d) Dynamically selected short baseline views. (e) Resulting depth map from improved selection in (d) and the increased confidence measure (f). Depth maps are estimated using the normalised patch based data-term with the occlusion robust data-term minimum, Section (4.4.3), and the per-pixel depth map confidence measure is computed using Equation (4.13).

reconstructing surface is maintained, effectively fixing the expected minimum baseline; decreasing the error in the depth map estimate according to Equation (7.3), and therefore increasing the quality of correspondence that can be obtained. While the threshold $\delta_{\mathcal{S}_{min}}$ ensures that overly redundant views are culled.

We separate all selected views into two sets, partitioned given a reference frame with pose $\mathtt{T}_{wr}$, that enable the occlusion robust depth map data-term minimum introduced in Section(4.4.3) which is used in depth map denoising developed in Chapter (4). Using the frames temporally situated before and after the reference frame we compute the camera translation delta direction $v_r = \nu\left[\mathtt{t}_{wr+1} - \mathtt{t}_{wr-1}\right]$, and compute the sets $\mathcal{B}^l_{sb} = \{I_i | \langle t_{wi} - t_{wr}, v_r \rangle\} \leq 0$ and $\mathcal{B}^r_{sb} = \{I_i | \langle t_{wi} - t_{wr}, v_r \rangle\} > 0$. Figure (7.6) illustrates the selection mechanism in use showing the increased confidence in the depth estimate obtained.

A second subset $\mathcal{B}_{wb}$ is also selected for use in a multiple view stereo estimation we refer to as MVS polishing from Chapter (5). This second depth map optimisation is initialised with the depth map denoising solution, and exploits the redundancy in a wider baseline



(a) (b) (c)

(d) (e) (f)

Figure 7.7: Demonstration of full two stage view selection mechanism in use with the depth map denoising (DMD) and multi-view stereo depth map estimation. (a) Frame selection using closest temporal views. (b) DMD using frame selection from (a). (c) MVS polish using frame selection from (a). (d) Dynamically selected wider baseline frames. (e) DMD using the improved short baseline selection from Figure (7.6) for comparison with the MVS polish result using the wide baseline selection from (d) shown in (f). We note that using the wider baseline view set with the MVS polish results in higher quality depth map estimates.

of views. Using the estimated minimum depth expected in the reconstructing frame, we select the *furthest* $m \geq 1$ frames using an increased threshold $\alpha\delta_{\mathcal{B}_{max}}$ and reduce the inter-frame threshold. This selection scheme is motivated by the fact that the denoised depth map provides a good initialisation to the global multi-view stereo optimisation, enabling the exploitation of wider baseline views to obtain greater reconstruction accuracy. The reduced inter-frame threshold enables sets of closely clustered views around the maximum threshold distance to be selected, providing redundancy in the gradient computation used in the linearised patch data-term in Equation(5.6). The resulting higher quality depth map estimation is illustrated in Figure (7.7).

In Figure (7.8) we demonstrate the multi-view depth map estimation for using the left-right view sets $\mathcal{B}_{sb}$, computed using the view selection mechanism on a reference frame from the Middlebury temple dataset that results in selection of 5 nearest frames to the reference from the full dataset. The occlusion robust data term results in reduced outliers in the depth map shown in Figure (7.8b). This can be compared to using the same frames without occlusion handling resulting in the depth map in Figure (7.8a) shown with a reduced confidence in

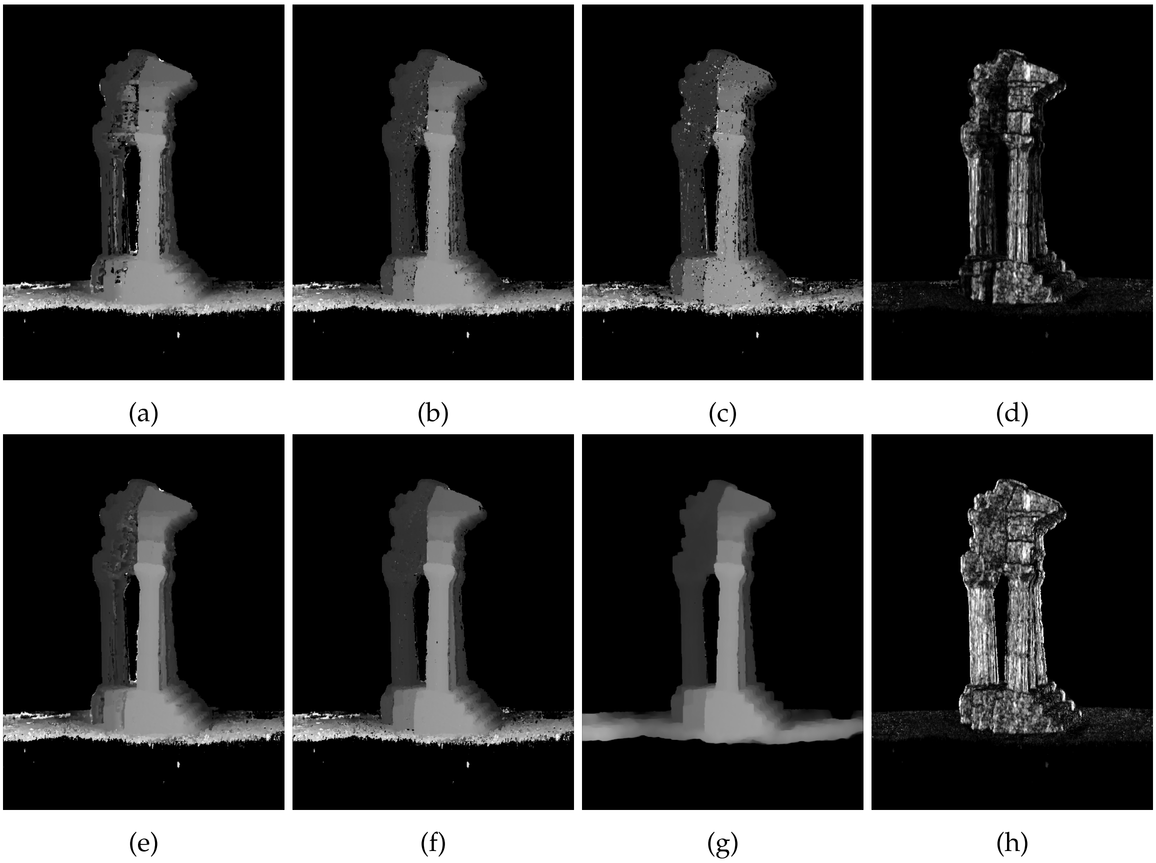


Figure 7.8: Demonstration of the depth map denoising pipeline from Chapter (4) using the neighbouring view selection applied to a reference from the Middlebury temple dataset. (a-c) Shows example data term minimum for the normalised patch based multi-view stereo data term computed: (a) without explicit occlusion handling (b) using an optimal left-right view mixture from Equation (4.19), and (c) using the minimum from the left or right frame set dataterm minimum. Using a per-pixel depth range prediction from a partially reconstructed model leads to reduced correspondence errors. (e) Demonstrates the data term minimum equivalent of (a) but where the epipolar search is restricted within the predicted interval obtained from the final model reconstruction. (f) Further uses the the per-pixel depth range prediction with the occlusion robust data term minimum. (g) Show the depth map denoising result applied to (f) which together with the confidence map (h) constitute the predictive depth map input that is integrated into the TSDF surface.

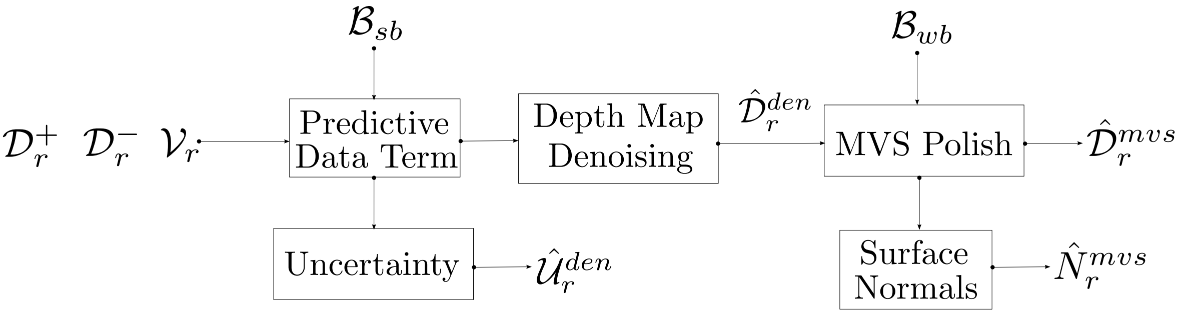

$\mathcal{B}_{sb}$
$\mathcal{B}_{wb}$
$\mathcal{D}_r^+$ $\mathcal{D}_r^-$ $\mathcal{V}_r$
Predictive Data Term
Depth Map Denoising
$\hat{\mathcal{D}}_r^{den}$
MVS Polish
$\hat{\mathcal{D}}_r^{mvs}$
Uncertainty
$\hat{\mathcal{U}}_r^{den}$
Surface Normals
$\hat{N}_r^{mvs}$

Figure 7.9: **Predictive Stereo**: search for the stereo data term minimum is restricted on a per pixel basis using the predictive depth bounds. The multi-view stereo optimisation is then initialised with the denoised depth map result.

non co-observable pixels shown in Figure (7.8d).

7.4.3 Predictive Stereo

Building on the multiple view stereo estimation techniques introduced in Chapters (4) and (5), the predictive stereo component exposes the full potential of the incremental depth map fusion pipeline which enables full frame predictions of the most up to date dense model in real-time. In this subsection we begin by detailing the per-pixel prediction mechanism that depends on the most up to date model reconstruction. We then describe the full predictive stereo component outlined in Figure (7.9).

Per-pixel Depth Range Prediction

In Chapter (6) we detailed the incremental fusion approach that enables depth map estimation and integration into a weighted volumetric TSDF surface representation. We also detailed the ray casting prediction mechanism that enables a direct rendering of the current surface into a virtual view, obtained as the zero level set of the implicit surface $\mathcal{S}(\mathbf{x}) = 0$. Associated with the volumetric TSDF, the function $\mathcal{W}$ holds a weight at each voxel computed from the integration of the uncertainty from all observations made on each voxel.

When integrating a surface measurements into the TSDF, truncation of the weighting function for negative SDF values beyond magnitude $\epsilon_{\mathcal{S}}^-$, is required to prevent interference of front and back surfaces. Also, due to the limited uncertainty on the surface measurement along a ray, positive SDF values are truncated beyond $\epsilon_{\mathcal{S}}^+$, while within the non truncated region the integration of hundreds of surface measurements leads to an approximation of the true signed distance function. As illustrated in Figure (7.10) we can render the extracted weight function associated with the level sets of a partially complete reconstruction. Here we look at the sets $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^-$, $\mathcal{S} = 0$ and $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^+$, where the level γ enables tuning of

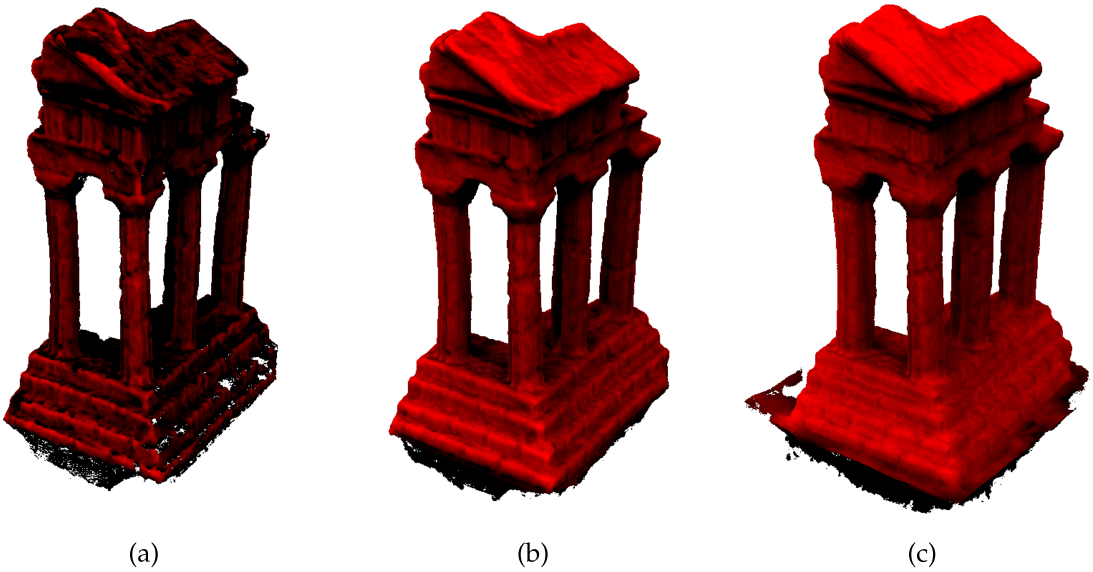


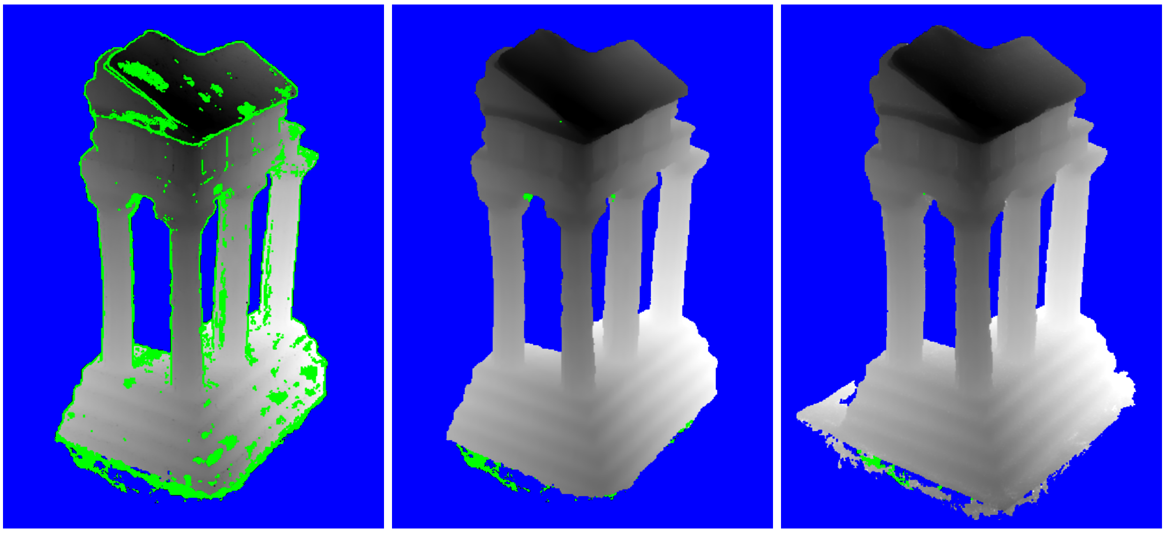
Figure 7.10: Representation of the surface confidence is captured over increasing level sets of the TSDF representation. Surface interfaces are shown for a partially reconstructed model, rendering the level sets (a) $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^{-}$, (b) $\mathcal{S} = 0$ and (c) $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^{+}$, where the intensity of the pixel value on the surface represents the estimated confidence of the voxel. The confidence for regions behind the surface which should not be directly observable $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^{-}$ is reduced relative to voxels in free space $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^{+}$.

the extracted range.

The implicit surface therefore provides not only a per-pixel depth prediction with an associated weight, but importantly, the prediction is available for the interval $\epsilon_{\mathcal{S}}^{+/-}$ either side of the zero crossing of the function. By taking a prediction either side of the zero-level set it possible to obtain a current depth map prediction with a depth range that captures the non-Gaussian, multi-modal nature of occlusion boundaries which is not captured explicitly using a point or surface element based representation.

Each of the three level sets and the associated depth maps, illustrated in Figure (7.11), can be extracted in a single ray traversal, starting with detection of the depth and weight for $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^{+}$. During ray traversal, detection of the specific level set can lead to three conditions. If a level set is not detected, no depth or weight are extracted at that pixel. If a surface is detected at the specified level set, but the extracted weight at the interface is below a threshold $\mathcal{W} < \epsilon_{\mathcal{W}}$, we also flag the prediction for the level as invalid. Otherwise, if the level set is detected at the pixel and $\mathcal{W} \geq \epsilon_{\mathcal{W}}$ then the weight and depth are extracted.

The resulting depth range prediction $\mathcal{D}_r^-$, $\mathcal{D}_r^+$ and validity mask $\mathcal{V}_r$ are set as follows. $\mathcal{V}_r(u) = 0$ if *either* any level set extraction weight value was below the specified threshold



(a) (b) (c)

Figure 7.11: Corresponding depth map predictions with pixel validity mask for the three extracted level sets shown in Figure (7.10). (a) Corresponds to the level set $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^{-}$, predicting the far range of the surface. Blue pixels indicate no predicted surface (estimated free space within the reconstruction volume), green pixels indicate a predicted surface exists but with an associated confidence estimate that is under a threshold. Equivalently (b) corresponds the zero-level set ($\mathcal{S} = 0$) surface prediction, and (c) shows the predicted distance to the near surface estimate $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^{+}$. The TSDF representation enables prediction of possible surface discontinuity changes given an small alteration in the surface location, this enables high quality prediction of the possible range over which a data term should be searched for. This view predictive capability can not be captured using a point based scene representation even when point-covariance is estimated since the uncertainty over the depth map given an uncertain surface is view dependent.

or free-space was detected. Otherwise, $\mathcal{V}_r(u) = 1$ and the maximum predicted depth $\mathcal{D}_r^{+}$ is set to set to the ray intersection with the far surface prediction $\mathcal{S} = \gamma \cdot \epsilon_{\mathcal{S}}^{-}$, and $\mathcal{D}_r^{-}$ is set to set to the ray intersection with the near surface prediction $\mathcal{S} = \epsilon_{\mathcal{S}}^{+}$.

Predictive Depth Map Estimation

The per-pixel depth range prediction $\mathcal{D}_r^{+/-}$ provides a bound on the depth measurement. Using the short-baseline frame selection, $\mathcal{B}_{sb}$, we therefore compute a depth map into the reference but restrict the search for the per-pixel dataterm minimum to lie within the bounds provided by $\mathcal{D}_r^{+/-}$. The data term minimum depth map is then denoised using the weighted Huber-$-\ell_1$ model from Section (4.5.4) producing an initial depth map $\hat{\mathcal{D}}_r^{den}$. Next we initialise the warp function in the continuous MVS framework from Section (5.2.2), and make use of the wider baseline frame selection $\hat{\mathcal{B}}_r^{mvs}$ to compute the MVS polished depth map $\hat{\mathcal{D}}_r^{mvs}$ which is used in the remaining depth map fusion component of the pipeline.

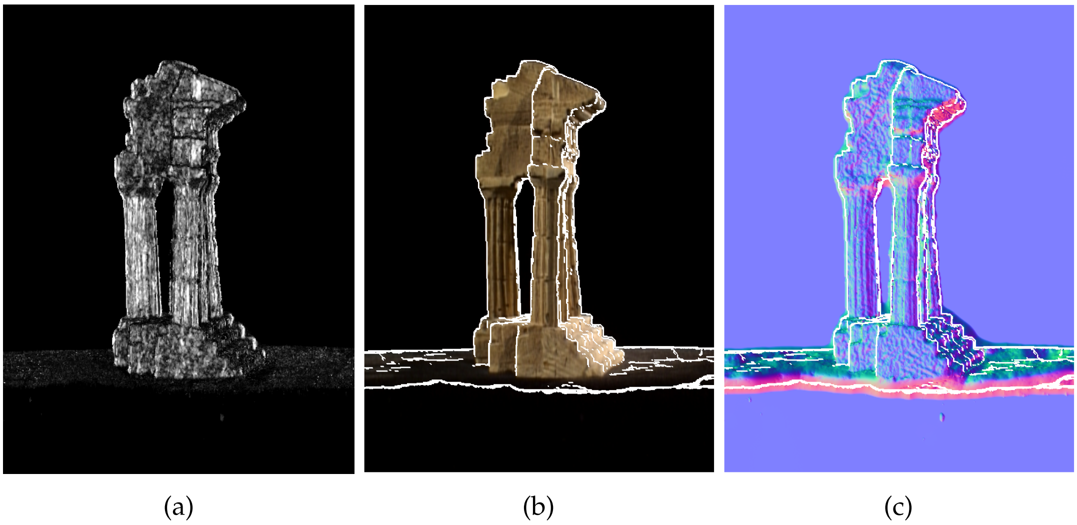


Figure 7.12: Depth map uncertainty (a) and normal map estimate (c) which together the depth map estimated for the reference frame (b), and used in the TSDF fusion. We note that culling of estimated values with reduced visibility results shown on the reference frame (b) results in the removal of potentially valid estimated depth map values.

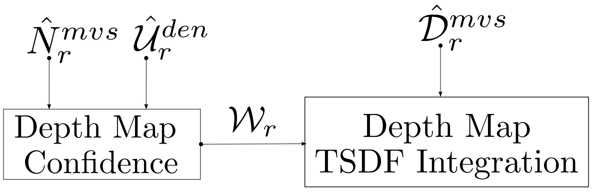
In Figure (7.8) we demonstrate the potential benefit from using range prediction in computing the multi-view stereo data term in comparison to using a complete search over the maximal range of depths $[d_{min}, d_{max}]$.

Depth Map Processing

Together with the depth map $\hat{\mathcal{D}}_r^{mvs}$, the final output of the predictive stereo component includes the depth map uncertainty $\hat{\mathcal{U}}_r^{den}$, computed on the data term minimum using Equation (4.13). We also compute a surface normal estimate using the method described in Section (6.1.1). We then measure the angle between the normal and viewing ray vectors at each pixel, to determine depth values of low visibility which are culled if the normal is near perpendicular to the viewing ray. In Figure (7.12) we show the resulting valid surface measurement values on a reference image from the Middlebury temple dataset estimated using the predictive depth estimation method. The associated depth map is shown in Figure (7.8g) resulting from the depth map denoising and multi-view stereo depth map estimation method.

7.4.4 Depth Map Fusion

As outlined in Figure (7.13a), the final stage of the passive reconstruction pipeline fuses the estimated depth map into a global surface reconstruction using the volumetric SDF approach presented in Chapter (6). Given the surface estimate uncertainty and normal estimates, we first compute a final depth map confidence $\mathcal{W}_r^m(u)$ for a pixel u in the reference


$\hat{N}_r^{mvs}$
$\hat{\mathcal{U}}_r^{den}$
$\hat{\mathcal{D}}_r^{mvs}$
Depth Map Confidence
$\mathcal{W}_r$
Depth Map TSDF Integration

(a)

Figure 7.13: **Depth Map Fusion**: Multiple view stereo depth map is integrated into the global model using weights computed from the data term uncertainty and estimated surface normals.

frame r:

$$\mathcal{W}_r(u) = w_{vis}(u) \cdot w_{atan}(u) \cdot \hat{\mathcal{U}}_r^{den}(u) \; , \tag{7.4}$$

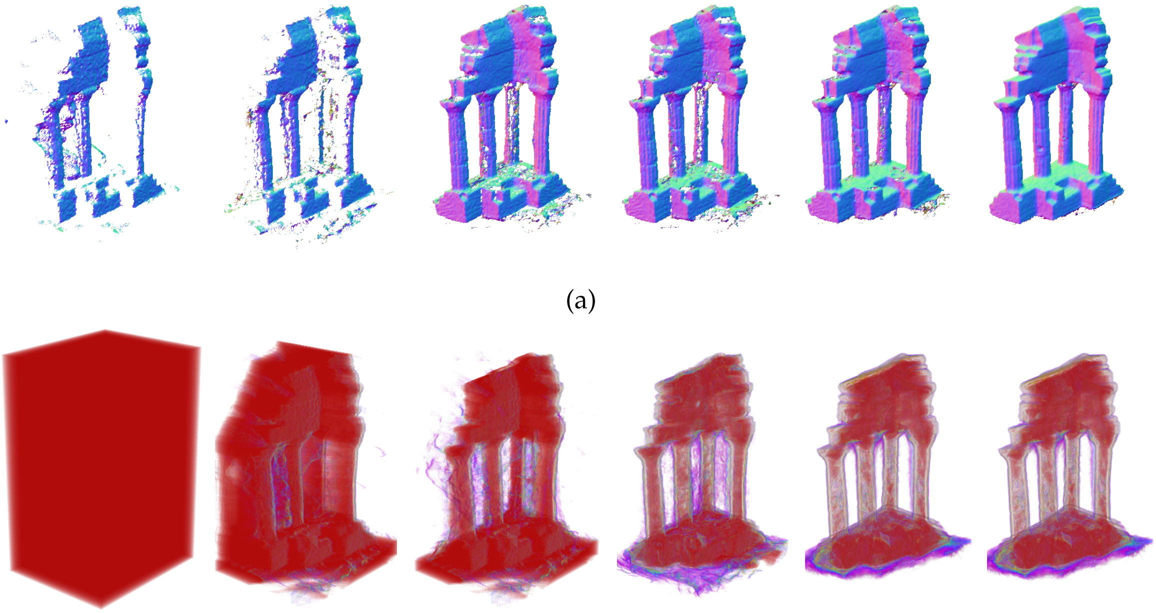
where w_{vis} is the thresholded visibility measurement:

$$w_{vis}(u) = \begin{cases} \left| \left\langle \nu \left[K^{-1}\dot{u} \right], \hat{N}_r^{mvs}(u) \right\rangle \right| & \text{iff } \left| \left\langle \nu \left[K^{-1}\dot{u} \right], \hat{N}_r^{mvs}(u) \right\rangle \right| \geq 0.05 \\ 0 & \text{otherwise} \; . \end{cases} \tag{7.5}$$

In Section (3.2) we outlined the radial lens distortion model which we use to obtain a rectilinear image. This is achieved in our system in practice by precomputing a warping function from each pixel in the rectilinear image to the sub-pixel location in the original image using Equation (3.14). The radial distortion in our wide angle lens results in pixels in the rectified image which map outside of the radially distorted original image. The weight $w_{atan} \mapsto \{0,1\}$ provides a binary mask to zero those invalid pixels in the depth map, since they have no contributing data. We note that in practice this is a very narrow pin-cushioning region. Finally, the depth map is integrated into the global volume with the pixel weighting using the incremental weighted average update in Equation (6.17).

In Figures (7.14a) and (7.14b) we show the continuous integration of depth maps into the TSDF for the full Middlebury temple dataset. The evolving surface reconstruction in rendered in increments of 30 integrated depth maps, rendering the surface normals, together with a visualisation of the volumetric SDF using front-back volume rendering.

In Figures (7.15b-7.15g) we capture the state of the fusion process at three slices through the volume for an evolving reconstruction of the temple dataset shown in full reconstructed form in Figure (7.15a). In Figures (7.15b - 7.15d), plot the value of the truncated signed distance for any voxel into which at least one measurement has been integrated. As surface integration continues visualise a current conservative estimate of free space (show in white), corresponding to saturation of the positive SDF, as well the decreasing region of voxels which are yet to have any integrated measurement (shown in blue). We also visu-



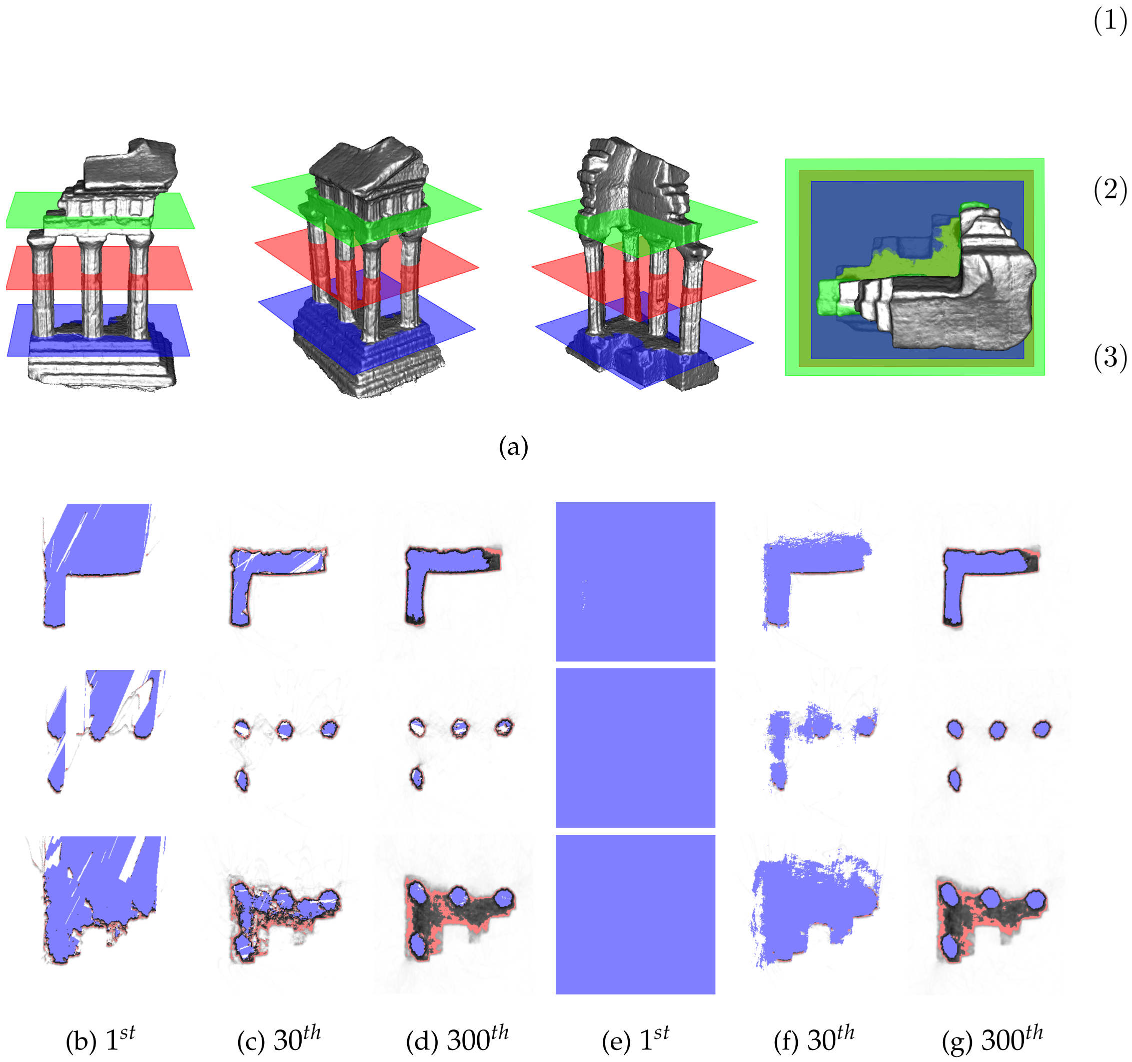
(a)

(b)

Figure 7.14: Surface reconstruction showing progression of 30 depth maps being incremental fused into the volumetric SDF, starting with 1 depth map (left). (a) Raycast surface normal rendering, using thresholding on the weight function $\mathcal{W}_r$ such that voxels with associated weights below a confidence threshold are treated as free space. (b) Volume rendering of the same surface evolution, where red shading highlights free space and surface interfaces are shown as a blue to green transition.

alise the evolving interface using a scaling of the signed distance value to show positive and negative regions, and highlight a thin band near the zero crossing (shown in red). We progressively show the volume slice at 30 depth maps, Figure (7.15c), and then 300 depth maps in Figure (7.15d) near model completion.

In Figures (7.15e-7.15g) we render the same surface evolution but where we utilise the voxel weighting function $\mathcal{W}_r$ to threshold low confidence SDF values, also shown in blue in the volume slices. For this reason, after one depth map is integrated the volume slices remain empty as illustrated in Figure (7.15e). As the surface evolution progresses the result is a conservative estimate of the reconstructed surface, as used in the prediction mechanism detailed in the previous subsection. Comparing the final volume with and without the thresholding, without the threshold free space errors are shown inside the temple columns shown in Figure (7.15d), while Figure (7.15g) demonstrates that these erroneous measurements are successfully attributed with a low confidence measurement and can be thresholded away.



(a)

(b) 1st (c) 30th (d) 300th (e) 1st (f) 30th (g) 300th

Figure 7.15: Complete reconstruction of the Middlebury temple model performed on the full dataset (a). We illustrate the volume slices from which the volumetric SDF evolution is tracked in figures (b-d) and (e-g), where each plane (top, middle, bottom) is associated with the slices (1,2,3). In (b-d) we show the TSDF evolution where freespace is shown in blue which zero level set show in red. (b) Shows the result after integrating 1 depth map, followed by (c) 30 depth maps and (d) shows the near complete model with 300 depth maps integrated out of 312. (e-g) show the equivalent evolution using confidence thresholding of the voxel weight function $\mathcal{W}_r$. Confidence based thresholding results in a a more conservative estimate of the known surface location used in model prediction as described in Subsection (7.4.3). We will also make use of the conservatively estimated geometry in the dense frame-model camera tracking method detailed in Chapter (8).

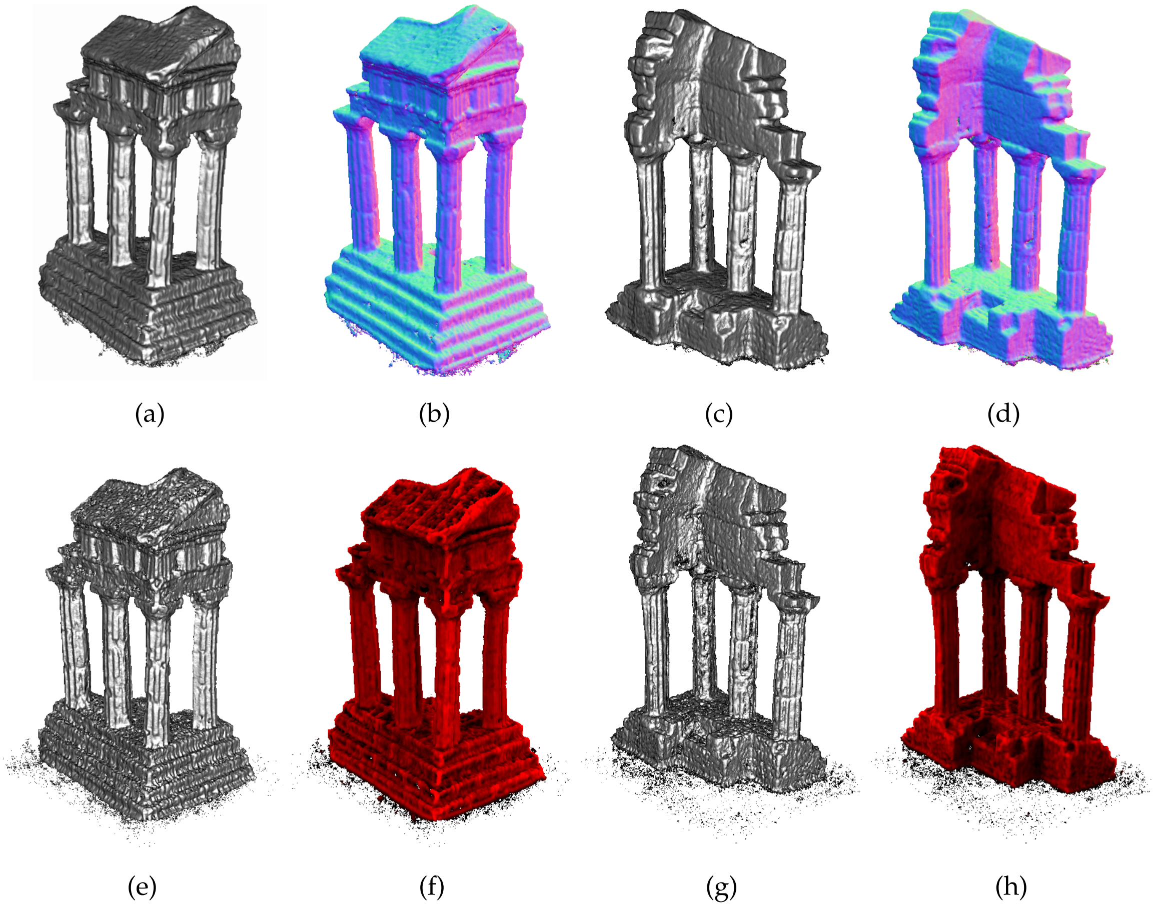


Figure 7.16: Reconstructed model using the full Middleburry temple sequence. Results shown for the complete predictive pipeline including depth map denoising and MVS polish shown in (a-d). (a,c) Model front and back views rendered with Phong shading. (b,d) Surface normal rendering. Resulting reconstruction fusing the data term minimum only depth maps, without depth map denoising or MVS polish are shown in (e-h), where (f,h) illustrate the reduced confidence of the surface estimation compared with the full pipeline confidence shown in Figure (7.10b).

7.5 Evaluating Live Dense Reconstruction

The passive reconstruction pipeline described in this chapter has been developed to exploit the dense frame capture of real-time video. Beyond the idealised input used in highly calibrated, fixed input, off-line MVS, the system has also been designed for interactive reconstruction, using the ability for a user in the loop to capture more data and complete the reconstruction as required for a given live application. This ability in turn raises a number of methodological challenges to system evaluation since one live dense reconstruction may exploit feedback to a user in a very different way to another system, resulting in the process in different input sequences being used.

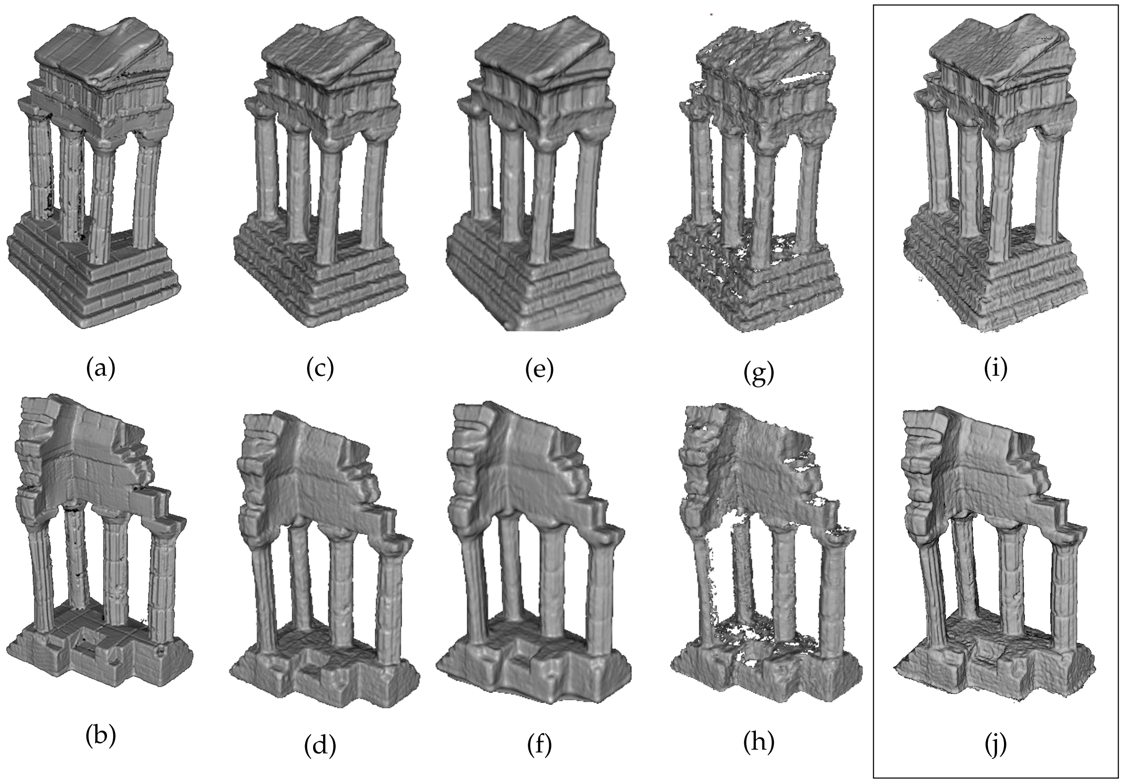


Figure 7.17: Qualitative comparison of reconstruction results for the full Middleburry temple sequence. (a,b) Ground truth model. (c,d) a top performing off-line method by **Hernández and Schmitt** (2004) which makes use of silhouttes. (e,f) Result from TV-hist (**Zach**, 2008) as used in the live dense reconstruction system of **Graber et al.** (2011). (g,h) Result from **Merrell et al.** (2007) available only with the 47 view ring dataset. (i,j) Reconstruction results from the pipeline described in this chapter.

Static Image Dataset Evaluation

Of the two MVS dataset discussed in the introduction to this Chapter, the newer dataset developed by **Strecha et al.** (2008) provides higher quality imagery but the sequences consist of only 20 wide baseline frames of high resolution. While we used subsampled versions of the fountain dataset in evaluation of the depth map estimation methods developed in Chapters (4) and (5), they are unsuitable for the pipeline developed here. In contrast, the MVS dataset introduced by **Seitz et al.** (2006) provides a starting point for comparison of our pipeline with other state-of-the-art systems due to the density of the full sequences.

In Figure (7.16) we present reconstruction results using the *full* Middleburry temple dataset. System parameters include a bounding box of size of $0.12 \times 0.18 \times 0.1$ meters with a resolution of $512 \times 256 \times 256$ voxels. View selection of a maximum of 4 nearest frames was used. Figures (7.16a-7.16d) illustrates results from the full pipeline consisting of depth map denoising and MVS polishing followed by TSDF integration with all 312 frames. The depth estimation pipeline was configured with a quantisation on the solution of 200 depth samples evenly spaced in inverse depth where the frustum range was clamped to the bounding box limits in each frame. The multi-view stereo pipeline was configured using the 3×3 mean subtracted SSSD cost and was optimised in both the depth map denoising and MVS polishing components using the gHuber-ℓ_1 model. In Figure (7.16e-7.16h) we demonstrate the same reconstruction parameters but where integration of the raw depth map data term minimum was used without de-noising or MVS polishing. In Figure (7.17) we provide a qualitative evaluation of the reconstructed with the ground truth model and the two systems on the evaluation page which are real-time capable alongside a state of the art offline reconstruction result. We achieve a real-time *capable* live dense reconstruction on the dataset: Computation time for the estimation and fusion of a single depth map using the pipeline detailed is approximately $32ms$ using the full resolution 640×480 image input, and scales accordingly to approximately $20ms$ when the input images are cropped tightly to the temple object. All depth map processing, fusion and rendering was performed on an NVIDIA 680GTX GPGPU with 8 Multiprocessors $\times$ 192 CUDA Cores/MP with a total of 1536 CUDA Cores.

Video Dataset Evaluation

Directly tackling the absence of useful evaluation aids for iterative systems, **Gruber et al.** (2010) created the *City of Sights* evaluation framework. Their goal is to enable and improve replicable evaluation of mixed and augmented reality research including real-time camera tracking performance, live dense reconstruction, and various real-time graphics applications within augmented reality including object relighting. They provide a model of an imaginary city scene comprising six buildings in both a virtual form as a ground truth mesh model shown in Figure (7.19c), and also as digital textured blueprints which can be printed out onto card and constructed into a physical model as illustrated in numerous figures throughout this chapter and also shown in use in Figure (7.18a). The authors provide evaluation of the physical model and record an accuracy of $2 - 3mm$, with mean, root mean square, and maximum errors of $1.93mm$, $2.46mm$, and $9.58mm$ respectively, achieved by scanning the paper construction to an accuracy of approximately $0.12mm$ and using a Hausdorff distance measure against the virtual ground truth mesh. The ability to construct a physical model opens up evaluation of live dense reconstruction methodology in ways which are impossible with a closed, static dataset.

In practice there are very few live dense reconstruction systems with which to compare the results in an interactive setting. To further understand the specific qualities of evaluating these new systems we engaged the visual SLAM and computer vision communities with the *1st IEEE Workshop on Live Dense Reconstruction from Moving Cameras* (LDRMC)

(**Newcombe, Davison, and Vogiatzis**, 2011a), at the International Conference on Computer Vision, 2011.

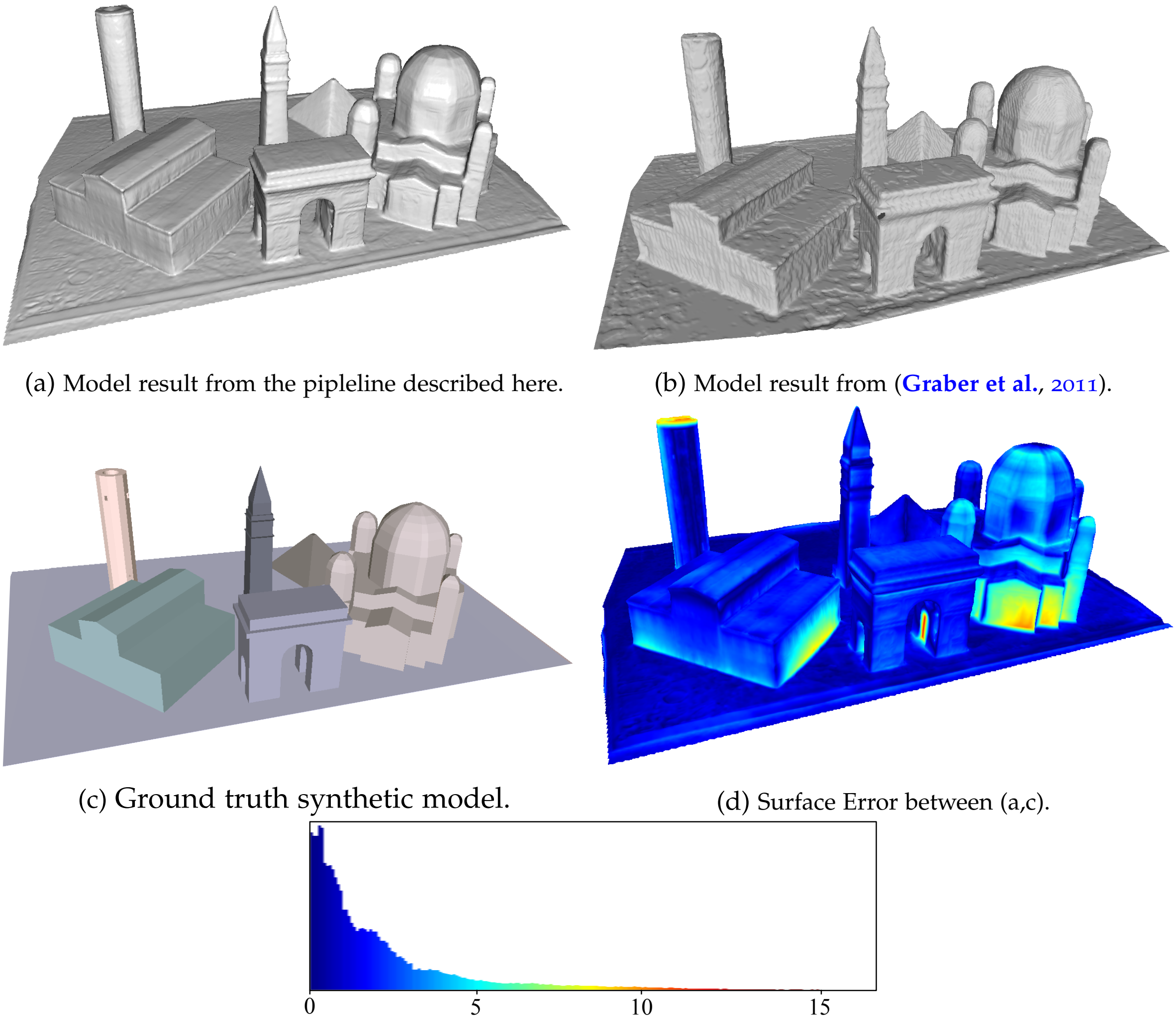
In Figure (7.18) we provide snapshots from the work shop where the system developed in this Chapter was run in a live demonstration side-by-side with the system developed by **Graber et al.** (2011). In both systems the parallel tracking and mapping system from **Klein and Murray** (2008) provided the real-time camera pose estimates with the key-frames being used as described in Section (2.1.2).

We also evaluated our system on a video sequence captured of the City of Sights dataset consisting of a $3''30'$ of loopy motion browsing the scene, again using pose estimates from PTAM. In Figure (7.19a) we render the resulting model geometry computed using our pipeline and for comparison with the reconstruction obtained using the $TV - hist$ convex optimization method used in **Graber et al.** (2011) shown in Figure (7.19b). Following the evaluation methodology used by **Gruber et al.** (2010) and employed in the model reconstruction evaluation used in **Graber et al.** (2011) and **Wendel et al.** (2012) we align the reconstruction with the virtual ground truth model using ICP, minimising the point-plane error (**Chen and Medioni**, 1992), including compensation for the scale ambiguity in reconstruction from a monocular camera. The Hausdorff distance measure is then computed against the ground truth mesh and we evaluate the resulting surface error, illustrated in Figure (7.19d) with the estimated error histogram. The resulting reconstruction accuracy is commensurate with the constructed paper model with increased error over larger stretches of supporting geometry where the paper model can flex. There is also reduced accuracy at surface edges. It is important to note that without improved physical model accuracy and ground truth pose estimation, further quantitative evaluation of the system is of questionable importance. We instead present the system as one of the first demonstrations of live dense reconstruction and will employ further effort in understanding how such evaluations can be formalised.

(a) (b)

(c) (d)

Figure 7.18: Live Dense Reconstruction in practice at the LDRMC workshop at ICCV 2011 (**Newcombe et al.**, 2011a). (a) *City of Sights* model by **Gruber et al.** (2010). Simultaneous live demonstration on the City of Sights model with (b) **Graber et al.** (2011). (c) the pipeline developed in this Chapter. (d) Live demonstration of the system by **Woodford et al.** (2011) .



(a) Model result from the pipleline described here.

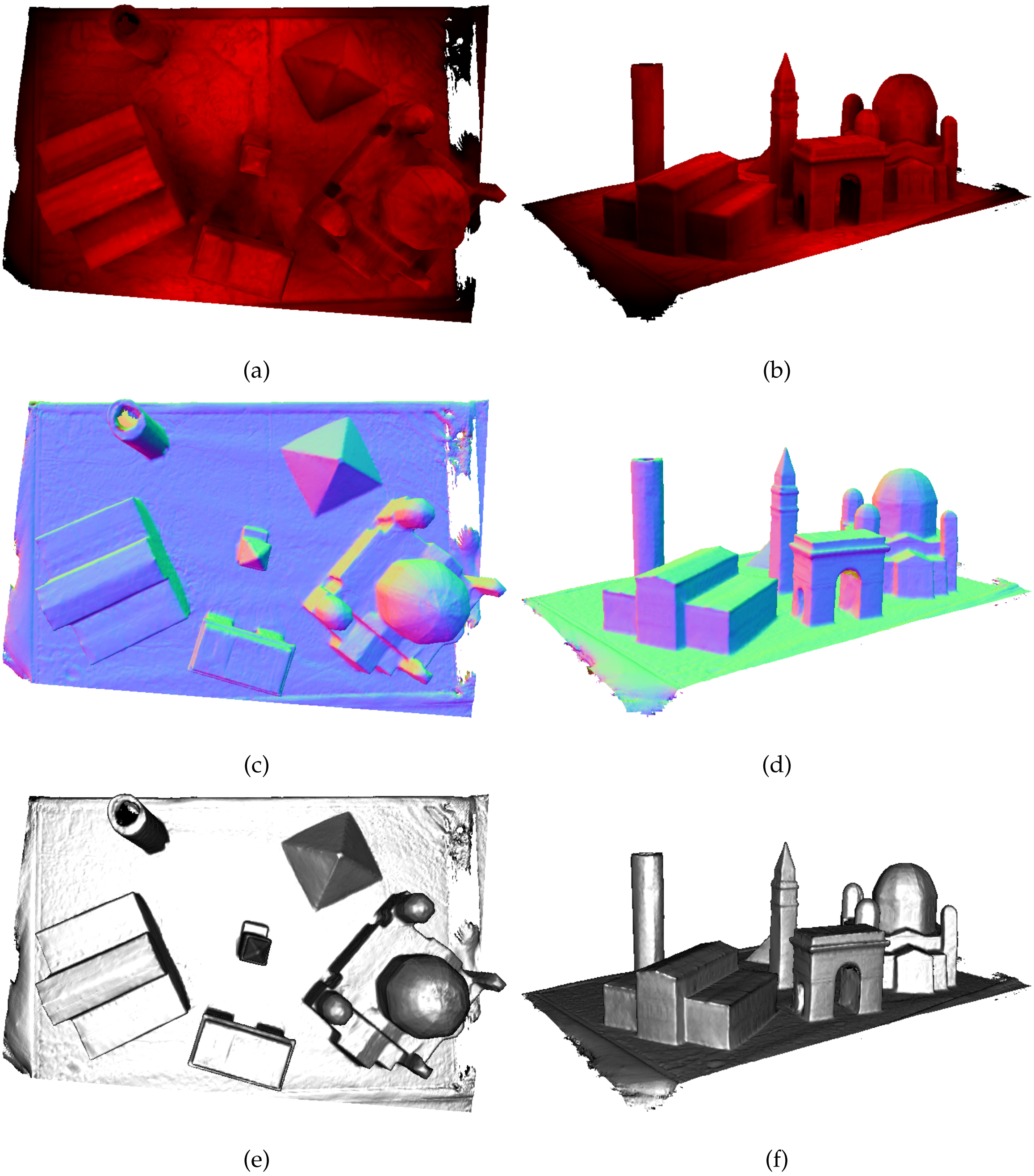
(b) Model result from (Graber et al., 2011).

(c) Ground truth synthetic model.

(d) Surface Error between (a,c).

(e) Surface error (mm) histogram and colouring for (d).

Figure 7.19: Reconstructing the City of Sights from Gruber et al. (2010). (a) Reconstruction from the pipeline described in this Chapter. (b) Result from Graber et al. (2011). (c) Virtual ground truth model. (d) Error surface visualisation between (a) and (c). (e) Error histogram with associated colour key for the surface error in (d), ranging from 0mm - 17mm error, quantised into 256 bins. We note that the cited root mean square error for the constructed paper model relative to the ground truth is approximatley 3mm (Gruber et al., 2010).



(a) (b)

(c) (d)

(e) (f)

Figure 7.20: Top and side views of a second reconstructed Graz City of Sights dataset with confidence map prediction (a,b) together with the normal map (c,d) and Phong shaded rendering (e,f).

7.6 Summary and Future Work

We have demonstrated the effectiveness of a depth map fusion paradigm in live dense reconstruction with a single moving camera, but a serious limitation of all of the depth map fusion pipelines discussed in the introduction of this chapter, and including the method described here, stems from the use of a regular grid implementation of the volumetric SDF, making a trade-off between reconstruction accuracy and volume size inevitable. In Chapter (1) we advocated dense reconstruction and dense SLAM approaches as a component within a more complete scalable SLAM system by exploiting submapping techniques. In the light of the advances made in the last decade into high quality dense reconstruction researchers have begun to tackle large scale scalable dense reconstruction. In Section (9.3) we outline recent work for real-time dense reconstruction using a commodity depth camera where the sparsity of surfaces in the embedding volume are exploited. Advanced work by **Fuhrmann and Goesele** (2011) goes beyond replacing the regular grid SDF with a multi-resolution or adaptive tree structure. They also tackle a major problem associated with integration of locally estimated projective depth measurements over vastly different scales of measurement uncertainty, a critical problem for dense reconstruction over a large range of scales since error scales quadratically with depth. This problem in particular is of special interest for passive reconstruction pipelines since in principle it is possible to reconstruct objects scaling from mountain ranges to leaves with a single moving camera.

8

Direct Tracking from Surface Models

Contents

In this chapter we describe how to make full use of the geometric and photometric predictions available from the dense surface model and perform model based tracking of a live sensor as it browses the scene. The result is a camera tracking pipeline that enables all pixels in a live image to be used directly in the optimisation of the camera pose, removing the need for the explicit feature extraction and matching components used in sparse visual SLAM pipelines.

The chapter is composed of two halves. First we develop a model based tracking framework for use with a single passive camera. Section (8.1) provides an overview of successful real-time tracking methodologies. In Section (8.2) we then formulate the *direct* tracking method for estimating the full $SE3$ pose of a single passive camera, making full use of both the geometric and photometric predictive capabilities described in Chapter (6). We make use of this *direct* tracking approach within the dense tracking and mapping (DTAM) visual SLAM system discussed in Section (9.2), where we compare the direct tracking approach in a real-time comparison with the feature-based tracking capabilities of PTAM.

In the second half of the chapter we turn our attention to tracking using a commodity depth sensor that provides live depth map estimates at frame-rate. In Section (8.3) we provide the background to tracking using geometric measurements alone and detail a direct approach for depth camera tracking using only the geometric predictions available from a dense model. In Section (8.4) we then develop a depth camera tracking pipeline that can directly exploit a volumetric SDF representation of the surface and provide a comparison of the basins of convergence for these trackers. In Section (9.3) we will describe the Kinect-Fusion dense SLAM system that combines direct depth map pose estimation with surface reconstruction, demonstrating the dense tracking and mapping components detailed in this thesis can be used to achieve online, real-time, SLAM without the need to explicitly extract and track features.

All of the methods developed perform a form of incremental pose estimation, and strongly rely on the small inter-frame motion which can be expected in live frame-rate imagery as a user or robot browses a scene. In Section (8.5) we describe a re-localisation strategy that is used to recover the camera pose if the incremental tracking mechanisms fail.

8.1 Motivation for Dense Photometric Tracking

Tracking from a known 3*D* model can be accomplished in a multitude of ways that span a range of feature-based to direct methods.

8.1.1 Monocular Model-Based Tracking

Lepetit and Fua (2005) provided an extensive overview of 3*D* tracking research with a particular emphasis on methods that achieve real-time or online operation using only a single passive video camera. As with the incremental dense reconstruction pipeline we discussed in Chapter (6), the emphasis here is on achieving robust and accurate estimation within a constant time frame applicable for real-time operation.

Tracking approaches can be categorised into techniques that compute a camera pose through a form of recursive or iterative estimation, relying on the previous camera pose to produce a strong initial estimate of the new state through a motion model, versus methods which perform tracking by detection, producing a reliable pose estimate without the need for such initialisation.

The online camera tracking systems that we are interested in make use of prediction mechanisms that drastically simplify the correspondence problem; whether correspondence is obtained explicitly, as in feature-based tracking frameworks e.g. extended Kalman Filter based tracking (**Davison and Murray**, 1998), or when estimating the camera pose by minimising a whole-image error using a direct optimisation approach introduced in Chapter (1) and detailed in this Chapter. In both types of system the ability to predict the location of features to be matched in the live frame results in more efficient estimation, due to the reduced image area over which the explicit correspondences are sought or by reducing the number of iterations required to minimise the direct tracking objective function (**Handa et al.**, 2012). If however live tracking does fail, tracking by detection provides a solution to the harder camera relocalisation problem; when little or no information is available about the current camera pose.

The general framework for tracking by detection has its roots in structure from motion estimation using robustly acquired feature correspondences introduced in its modern form by **Torr and Zisserman** (1999). Geometric primitives such as points or line segments with associated descriptors based on the photometric appearance of the model are extracted in an off-line learning phase (**Lepetit et al.**, 2005). Estimation of the relative transform between the model and camera is then obtained by matching the model descriptors with equivalent primitives extracted from the live frame leading to $2D - 3D$ correspondences. Correspondence can be efficiently achieved using vocabulary trees (**Nister and Stewenius**, 2006) based on inverted file indexing (**Sivic and Zisserman**, 2003). Assuming enough model features can be detected in a live frame, the tracking by detection framework eliminates any pose error accumulated in the previous frame, producing an independent pose estimate in each new frame. While such methods have been demonstrated to work in real-time (**Lepetit and Fua**, 2006), they are typically more computationally demanding than the recursive estimation techniques that assume small inter-fame motion.

Coming out of earlier non-visual target tracking (**Bar-Shalom and Fortmann**, 1988), recursive estimation methods instead make use of a strong prior on the incremental nature of the pose update. **Harris and Stennett** (1990) introduced RAPID (Real-time Attitude and Position Determination), the first monocular model-based tracking system to demonstrate real-time performance. Performed either on-line or off-line, the system first extracts feature locations in the geometric model that project to salient image features such edges. In each new frame, the system computes an estimate of the model features using a pose prediction through a motion model. $2D - 3D$ correspondences are then sought by searching *locally* near the predicted feature locations. Given explicit correspondences, a Kalman filter style update on the camera pose state is then performed. A multitude of modern systems work using essentially the same predictive correspondence seeking framework.

Both of the approaches outlined above make use of explicit feature correspondences, and this leads to questions of whether a sufficient set of features can be extracted and matched in the live frame. As we discussed in Chapter (1), the explicit correspondence using the feature extraction and matching pipeline can result in catastrophic failure for a visual SLAM system undergoing rapid agile camera motion or when tracking through regions of low texture. In Section (1.4) we presented an alternative *direct* approach that instead takes advantage of a whole image prediction from a dense generative appearance model, enabling a pixel-wise error to be defined over the whole image for a given camera pose. Camera tracking is then be achieved using standard gradient descent style optimisation on the energy function, obtaining an implicit dense correspondence between the surface model and image.

8.1.2 Direct Alignment

All direct approaches formulate a pixel value based error measure between a target (model) image and the current view under a parametrisation of motion between the images. In this simplest case, motion between the model and current views is parametrised directly in the image plane. For example, if the current view is nothing but an in plane $2D$ translation of the model image, then from the brightness constancy assumption we have $\mathcal{I}_l(u+t) = \mathcal{I}_r(u)$ for $u \in \Omega$. A simple whole image error can be computed by summing up all pixel errors under a quadratic penalty. The estimation of the parameter vector can be achieved by searching for the $t \in \mathrm{R}^2$ that minimises this whole-image alignment error:

$$\underset{t \in \mathrm{R}^2}{\operatorname{argmin}} \left\{ E(t) = \sum_{u \in \Omega} \left(\mathcal{I}_l(u+t) - \mathcal{I}_r(u) \right)^2 \right\} . \tag{8.1}$$

Although impractical in general, for the above $2D$ image motion it is possible to search a small region of the parameter space explicitly, an example of the error surface for a local region around a known solution was given for the image alignment example in Chapter (1), from which the global minimum can be extracted.

Lucas and Kanade (1981) introduced the now widely used gradient descent based incremental approach to solving the alignment problem. Since $E(t)$ is non-convex in the parameter vector t, they first obtain a first order Taylor series expansion of $I_l(u + t_0 + \Delta t)$ given an initial estimate t_0. The linearised error function under a quadratic penalty function is then readily solved for Δt via the normal equations, that then is used to update the current estimate $t \leftarrow t_0 + \Delta t$.

Importantly, gradient descent on the whole-image error generalises to arbitrary transformation of the pixel locations. Replacing the simple translation vector by a general pixel transform $\mathbf{w}(x, u)$, and inserting a robust metric ψ in place of the quadratic penalisation in Equation (8.1), we can define the more general error function taking as an argument a parameter vector $x \in R^n$:

$$E_w(\mathbf{x}) = \sum_{u \in \Omega} \psi\left(e(u, \mathbf{x})\right) \tag{8.2}$$

$$e(u, \mathbf{x}) = \mathcal{I}_l(\mathbf{w}(u, \mathbf{x})) - \mathcal{I}_r(u) \; . \tag{8.3}$$

Following the parametric optimisation approach outlined in Section (3.3), we approximate the second order Taylor series expansion of $E_w(\mathbf{x}_0 + \Delta x)$ in Equation (3.34), replacing the Hessian with its Gauss-Newton approximation in Equation (3.35). Setting the derivative of this locally convex approximation to zero results in the weighted normal equations which are then solved for the parameter increment Δx:

$$\Delta x = -\left(\sum_{u \in \Omega} J(u, \mathbf{x}_0)^\top J(u, \mathbf{x}_0)\right)^{-1} \sum_{u \in \Omega} \psi'(e(u, \mathbf{x}_0)) J(u, \mathbf{x}_0) \; . \tag{8.4}$$

Here the per pixel Jacobian term $J(u, \mathbf{x}_0)$ is:

$$J(u, \mathbf{x}_0) = \left. \frac{\partial \mathcal{I}_l(\mathbf{w}(u, \mathbf{x}))}{\partial \mathbf{x}} \right|_{\mathbf{x}_0} \; , \tag{8.5}$$

and $\psi'(e(\mathbf{x}_0))$ computes the derivative of the penalty term *wrt* to the error:

$$\psi'(e(u, \mathbf{x}_0)) = \left. \frac{\partial \psi(e(u, \mathbf{x}))}{\partial e(u, \mathbf{x})} \right|_{\mathbf{x}_0} \; . \tag{8.6}$$

The resulting incremental update Δx, is then added onto the current estimate of the parameter vector:

$$\mathbf{x} \leftarrow \mathbf{x}_0 + \Delta x \; . \tag{8.7}$$

This new estimate is then used in an updated linearisation point of the whole image error and the optimisation is iterated. In Section (8.2.3) we will look at the coarse-to-fine embedding of this iterative optimisation, to improve the correctness of the linearisation required for the gradient descent approach. The final parameter estimate can be acquired in a just-in-time manner given a fixed computational budget, or otherwise when convergence is achieved using the general criteria discussed for Gauss-Newton optimisation in Section (3.3). We will use this formulation in remaining sections of the chapter, where we detail the direct approach for camera pose estimation.

A number of essential works on the iterative direct estimation method have been produced.

Irani and Anandan (1999) summarised several of the important aspects of direct tracking, demonstrating the use of robust error functions in rejecting outliers outside of the modal motion known as the locking property (Irani et al., 1994); the use of coarse to fine optimisation to help in convexification of the error function (Bergen et al., 1992); and replacement of the single pixel brightness constancy based error term with more photometrically robust measures. Building on the practical success of the Luckas-Kanade technique, Baker and Matthews (2004a) and Szeliski (2006) provided a thorough exposition of the range of developments within the direct tracking framework, including more efficient formulations for warp functions that form a group, resulting in the inverse compositional method requiring only a single Jacobian computation (Baker and Matthews, 2004b); comparison of techniques for robust estimation using pixel confidence weighting of the normal equations and robust cost function (Baker et al., 2003b); robustness to linear changes in appearance (Baker et al., 2003a); and incorporating prior knowledge on the solution into the optimisation (Baker et al., 2004a).

8.2 Direct Photometric Tracking

In this section we detail the use of the direct approach described above for estimating the pose of a live camera. We assume that the camera is viewing a scene for which we have obtained a (partial) dense reconstruction.

We are interested in computing the *live* camera pose relative to a dense model in a common *world* frame of reference in which the model is anchored, $T_{wl} \in SE_3$. We will formulate all optimisations relative to a known *reference* frame with pose T_{wr}. Using the techniques developed in Chapter (6) we can compute a geometric prediction in the form of a depth map D_r and a photometric prediction $\mathcal{I}_r$. Solving for the relative transform T_{lr} between the reference and live frames, the live camera to world transform is then $T_{wl} = T_{wr}T_{lr}^{-1}$.

SO_3 and SE_3 Incremental Parametrisation

We parametrise the desired transform $T_{lr} = [R_{lr}|t_{lr}] \in SE_3$ as a composition of transformations starting from an initial estimate $\tilde{T}_{lr}$,

$$T_{lr} = \exp(\hat{\mathbf{x}}^n)\exp(\hat{\mathbf{x}}^{n-1})...\exp(\hat{\mathbf{x}}^0)\tilde{T}_{lr} \ . \tag{8.8}$$

Here, the minimal set of rotation and translation parameter $\mathbf{x} \in \mathbb{R}^6$s for the rigid body transform is a stacked vector comprising a translation and rotation component:

$$\mathbf{x} = \begin{pmatrix} v \in \mathbb{R}^3 \\ \omega \in \mathbb{R}^3 \end{pmatrix} \ , \tag{8.9}$$

and the operator ˆ forms the matrix:

$$\hat{\mathbf{x}} = \begin{pmatrix} [\omega]_\times & v \\ 0 & 0 \end{pmatrix} \in \mathfrak{se}_3 \, , \tag{8.10}$$

where $[\omega]_\times \in \mathbb{R}^{3\times 3}$ is a skew-symmetric rotation component:

$$[\omega]_\times = \begin{pmatrix} 0 & -\omega_2 & \omega_1 \\ \omega_2 & 0 & -\omega_0 \\ -\omega_1 & \omega_0 & 0 \, . \end{pmatrix} \, . \tag{8.11}$$

The exponential map $\exp : \mathfrak{se}_3 \mapsto SE_3$ takes the minimal parameter vector to the corresponding Euclidean rigid body transform SE_3 group (**Ma et al.**, 2003). When $\mathbf{x} \approx 0$ the the infinitesimal transform enables a locally valid first order linearisation of $\exp(\hat{\mathbf{x}})$ to approximate the transformation as a linear function of the incremental parameters $\mathbf{x}$.

Since we will be performing optimisation with respect to the composition of the incremental transformation, we replace the additive update notation $\mathbf{x} \leftarrow \mathbf{x}_0 + \Delta x$ with the compositional form $\exp(\hat{\mathbf{x}}) \exp(\hat{\mathbf{x}}_\mathbf{0})$. Importantly, we will also override the definition of $E_w(\mathbf{x} + \Delta)$. Specifically since we need to perform the Taylor series expansion of any error function E_w using the compositional update form, we will be performing linearisation around the identity transform defined with $\mathbf{x} = 0$, and replacing all additive terms in the usual expansion to the compositional form.

In the following two sections we look at estimating the 6DoF pose of a camera relative to a known reference frame. We will start by first looking at estimation of camera motion which is assumed to be undergoing pure $3D$ rotation. In this case the warp function utilised is independent from the observed scene depth. While the motion assumption is clearly violated by a camera under general motion, it is often the case that the largest component of pixel displacement between live frames of a moving hand-held camera are caused by camera rotation. We have found that by first approximating the full 6DoF problem with the lower dimensional one, we can obtain an initial estimate of the motion parameters that can better initialise the full SE_3 camera motion, reducing the number of iterations required for full 6DoF camera tracking.

8.2.1 Tracking Camera Rotation

For a camera undergoing pure rotation, the pixel transform from the reference frame into the live view takes the form of a homography parametrised by the SO_3 component of the rigid body transform:

$$u_l = \pi \left(K R_{lr} K^{-1} \dot{u}_r \right) \; . \tag{8.12}$$

Given an initial rotation estimate $\tilde{R}_{lr}$, our function is parametrised by $\omega \in \mathbb{R}^3$ using the the skew-symmetric rotation matrix in Equation (8.11):

$$\mathbf{w}_{SO_3}(u_r, \omega) = \pi \left(K \exp([\omega]_\times) \tilde{R}_{lr} K^{-1} \dot{u}_r \right) \; . \tag{8.13}$$

Inserting $\mathbf{w}_{SO_3}$ into the whole image error in Equation (8.2) we now perform the linearisation of $E_w(\mathbf{x}_0 + \Delta)$ around $\mathbf{x}_0 = 0$ with $\Delta = \omega$; hence we compute $J(u, \mathbf{0})$ in Equation (8.5) for $\mathbf{w}_{SO_3}$:

$$J(u, \omega) = \frac{\partial I_l(\mathbf{w}_r)}{\partial \mathbf{w}_r} \frac{\partial \mathbf{w}_r(u, \omega)}{\partial K \exp([\omega]_\times) \tilde{R}_{lr} K^{-1} \dot{u}_r} \frac{\partial K \exp([\omega]_\times) \tilde{R}_{lr} K^{-1} \dot{u}_r}{\partial \omega} \; . \tag{8.14}$$

Defining $(x, y, z)^\top = \hat{R}_{lr} \dot{u}_r$, the resulting 1×3 gradient vector for pixel u is computed as:

$$J(u, \omega) = \begin{pmatrix} \nabla_x I_l \\ \nabla_y I_l \end{pmatrix}^\top \begin{pmatrix} \frac{f_x}{z} & 0 & -\frac{x f_x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{y f_y}{z^2} \end{pmatrix} \begin{pmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{pmatrix} \; . \tag{8.15}$$

Evaluating the Jacobian at $\mathbf{x}_0 = 0$ together with chosen penalty term derivative (see later Section (8.2.4) for details on this derivative), the linear system is solved for ω, and the resulting incremental rotation is composed onto $\tilde{R}_{lr}$:

$$\tilde{R}_{lr} \leftarrow \exp([\omega]_\times) \tilde{R}_{lr} \; . \tag{8.16}$$

For warp functions that form groups including the SO_3 pose estimation described above, **Baker and Matthews** (2004b) introduced the efficient inverse-compositional approach that drastically reduces the practical cost for single core processors of computing an iteration by removing the need to recompute the per pixel terms $J(\mathbf{0})$. **Malis** (2004) introduced ESM (Efficient Second order Method) using a higher quality Hessian approximation in the approximation of E_w, resulting in quadratic convergence rates and more recently demonstrated within a visual SLAM setting for real-time spherical moasicing (**Lovegrove and Davison**, 2010) and real-time planar tracking (**Lovegrove et al.**, 2011). Within our live camera tracking pipeline we utilise the trivial parallelisability of the algorithm described above, implementing the evaluation of the Jacobian and error terms in Equation (8.5) and Equation (8.6) on commodity GPGPU hardware as described in Section (3.5). We note that within the parallel setting the computational cost of the direct tracking approach is dominated by the reduction operation which is linear in the dimensionality of the reduced vector. This provides the impetus to reduce the number of iterations required for the higher dimension 6DoF estimation, described below, by first attempting to minimise image motion resulting from rotation alone. We make use of this two stage tracking approach within all of the 6DoF tracking systems described in this chapter.

8.2.2 Tracking Camera Rotation and Translation

To track the full 6 6DoF of the camera we make use of the full geometric and photometric prediction and perform. We obtain the camera to world transform T_{wc} by estimating a relative pose T_{lr} between the live frame and a reference frame into which we have predicted a depth map D_r and image $\mathcal{I}_r$. **Baker et al.** (2004b) describe the 2.5D Lucas-Kanade algorithm, extending the forward additive approach for aligning an image into a reference frame using a template surface representation in the form of a 3D point associated with each pixel in the reference frame image.

Following the incremental estimation method, we assume an initial estimate $\tilde{T}_{lr}$ and parametrise the warp function with $\mathbf{x} \in \mathbb{R}^6$, transforming the geometry at each pixel in the reference frame into the live frame:

$$\mathbf{w}_{SE_3}(u, \mathbf{x}) = \pi\left(K \exp(\hat{\mathbf{x}}) \tilde{T}_{lr} \mathcal{D}_r(u) K^{-1} \dot{u}_r\right) . \tag{8.17}$$

This is exactly the warp function used throughout the previous depth estimation chapters, where here we are optimising *wrt* the pose parameters instead of the depth. Inserting $\mathbf{w}_{SE_3}$ into the whole image error in Equation (8.2) we perform the linearisation of $E_{SE_3}(\mathbf{x}_0 + \Delta)$ around $\mathbf{x}_0 = 0$ with $\Delta = \mathbf{x}$. Hence we compute the Jacobian $J(u, \mathbf{0})$ for $\mathbf{w}_{SE3}$:

$$J(u, \mathbf{x}) = \frac{\partial \mathcal{I}_l(\mathbf{w}_r)}{\partial \mathbf{w}_r} \frac{\partial \mathbf{w}_r(u, \mathbf{x})}{\partial K \exp(\hat{\mathbf{x}}) \tilde{T}_{lr} \mathcal{D}_r(u) K^{-1} \dot{u}_r} \frac{\partial K \exp(\hat{\mathbf{x}}) \tilde{T}_{lr} \mathcal{D}_r(u) K^{-1} \dot{u}_r}{\partial \mathbf{x}} . \tag{8.18}$$

Defining $(x, y, z)^\top = \tilde{T}_{\hat{l}r} \mathcal{D}_r(u) K^{-1} \dot{u}_r$, the resulting 1×6 gradient vector for pixel u_r is computed as:

$$J(u, \mathbf{x}) = \begin{pmatrix} \nabla_x I_l \\ \nabla_y I_l \end{pmatrix}^\top \begin{pmatrix} \frac{f_x}{z} & 0 & -\frac{x f_x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{y f_y}{z^2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & z & -y \\ 0 & 1 & 0 & -z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \end{pmatrix} . \tag{8.19}$$

Evaluating this Jacobian together with the derivative of the chosen penalty term at $\mathbf{x}_0 = 0$, the weighted normal linear system in Equation (8.4) are solved for $\mathbf{x} = [v, \omega]$ and taken to an element SE_3 via the exponential map. Composition onto the initial transformation estimate computes the new solution point:

$$\tilde{T}_{lr} \leftarrow \exp(\hat{\mathbf{x}})\tilde{T}_{lr} \tag{8.20}$$

8.2.3 Coarse to Fine Optimisation

To ensure that the quadratic approximation to the error function is meaningful and hopefully leads to minimisation of the original error, the inter-frame pixel displacements required to bring the images into alignment should not be too large. The criteria for too large in practice depends on both the geometry and texture appearance present in the scene. Given the initial transformation used in the linearisation of $\mathcal{I}_l$, the brightness constancy assumption relating the images produces the equivalent optic flow constraint on a pixel:

$$\mathcal{I}_l(\mathbf{w}(u, \Delta\mathbf{x})) \approx \mathcal{I}_l(\mathbf{w}(u, \mathbf{0})) + J(\mathbf{0})\Delta\mathbf{x} \, , \tag{8.21}$$

$$\Rightarrow \mathcal{I}_r(u) - \mathcal{I}_l(\mathbf{w}(u, \mathbf{0})) \approx J(\mathbf{0})\Delta\mathbf{x} \, . \tag{8.22}$$

For this linearisation to hold we must ensure that the magnitude of the displacement in pixel units is not larger than the characteristic width of the texture at the corresponding locations in the reference and live images.

By filtering out higher frequency components we can reduce the aliasing that occurs for pixels undergoing larger displacements (**Bergen et al.**, 1992). This forms the basis of the now standard hierarchical or coarse to fine motion estimation as described in the original paper by **Lucas and Kanade** (1981). A considerable computational saving is also achieved subsampling the filtered image to a sufficient resolution for representation of the frequencies present.

Figure (8.1) outlines the general coarse to fine optimisation scheme for both $SO3$ and $SE3$ tracking. Looking at first at rotation only tracking: given the initial pose parameters, we begin optimisation on the coarsest resolution of the image pyramid. The warp function at every level of the pyramid is associated with a scaled intrinsic calibration matrix, which is used in the linearisation of the error function. The resulting weighted normal equations are solved for the incremental rotation parameters, and the associated $SO3$ rotation transformation is composed onto the initial pose parameters to obtain a new linearisation point. We continue to iterate the non-linear optimisation on the coarse pyramid level until convergence criteria is met, or a fixed maximum number of iterations passes. We then move to optimisation on the next finest resolution of the pyramid, initialising the parameter vector with the current solution, and continue optimisation until convergence or a fixed number of iterations have passed. We continue moving to finer levels of processing, but note that this

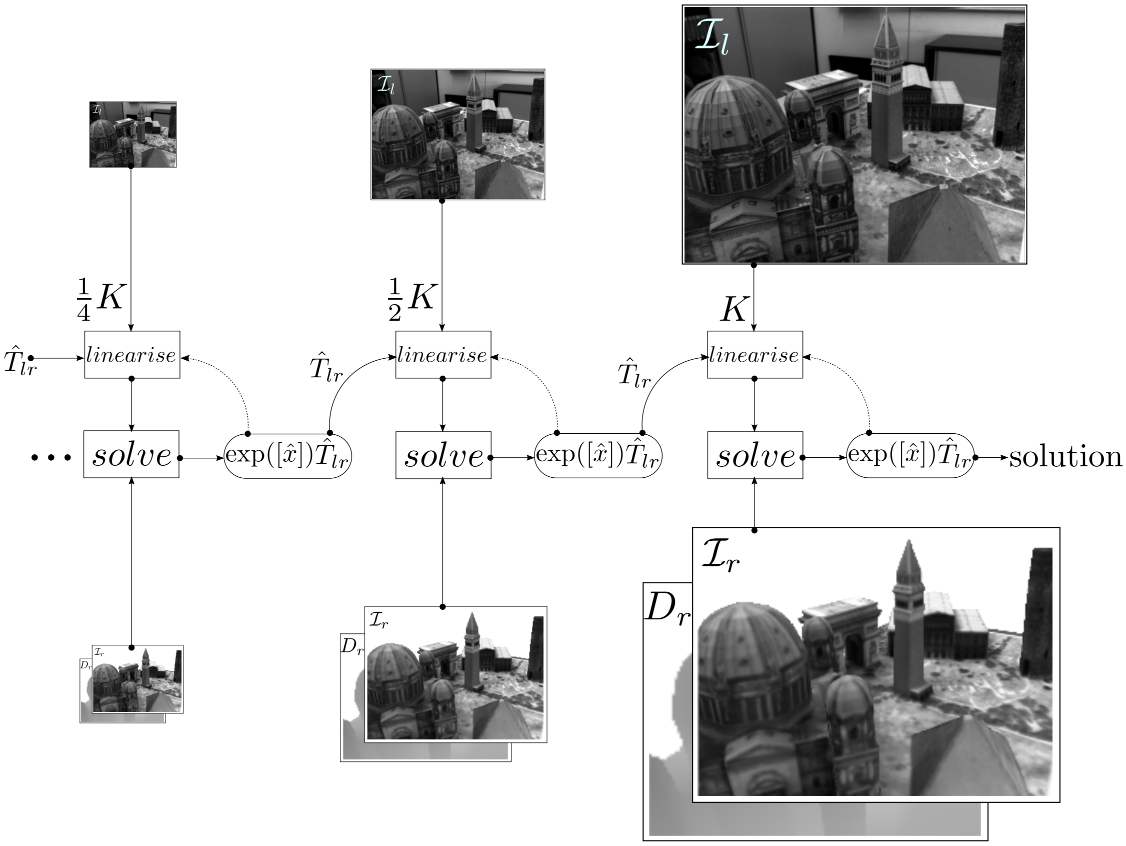


Figure 8.1: Dense tracking pipeline using a coarse to fine with warping scheme for both SO_3 and SE_3 optimisation. Given the reference image (and for SE_3 optimisation also the geometry) pyramids and also a live image we estimate the relative transform between the two frames starting with allignment between the coarest images. Gauss-Newton optimisation at each pyramid level requires correct scaling of the calibration matrix to account for the change in image resolution. Optimisation proceeds in a coarse to fine manner using the estimated pose from a previous coarser stage to initialise the linearisation of the error function at each new level.

need not be the original image resolution since even for the large amounts of sub-sampling in the image the linear system in each iteration can be massively overdetermined.

The coarse to fine optimisation for SE_3 pose estimation utilises the dense predicted geometry and requires a little more attention. As discussed in the Chapter (6), the depth map prediction from a partially completed model may contain pixels with no valid geometry prediction. Given the depth map validity mask V_r we perform sub-sampling using the a min operation on valid depth values within a 2×2 neighbourhood. The min operation replaces the low pass filter used in image sub-sampling to ensure that depth discontinuities are not smoothed over. Pose optimisation again begins at the coarsest pyramid level. It should be noted that after each Gauss-Newton gradient descent iteration for the SE_3

optimisation, we are faced with a decision of whether to update the the predicted geometry by setting the reference frame to the current estimated pose, or to continue with a constant reference frame. The main benefit to updating the reference frame given the estimated camera pose, is a potentially improved geometric and photometric prediction. Depth discontinuities that will occur with relative translation between the reference and estimated frames are reduced, and if using the view dependent texture mapping based photometric prediction described in Section (6.4), the photometric accuracy of the model can be increased as the parameters convergence on the correct solution. However, as noted in Section (8.2.6), there is a substantial computational cost associated with predicting the view at each iteration.

8.2.4 Robust Penalty Functions

Implementation of the tracking methods used in the preceding sections requires computation of the penalty function derivative $\psi'(e(u, \mathbf{x}_0))$ in Equation (8.6). A specific penalty can be derived in a principled way given assumptions over the error distribution in likelihood form. The original quadratic penalty term, $\psi(e(u, \mathbf{x}_0)) = \frac{1}{2}e(u, \mathbf{x}_0)^2$ used in **Lucas and Kanade** (1981) is obtained when assuming a Gaussian distribution over error in the its likelihood form, where the derivative is simply $e(u, \mathbf{x}_0)$. This derivative is also known as the *influence function*, since when for non-Gaussian distributed error over the likelihood, the penalty function obtained is not quadratic, resulting in the derivative ψ' being a non-linear function of the error which can be viewed as weighting the residuals in the standard least squares estimation procedure. As discussed in Section (3.4.1) the ℓ_1 penalty follows from assuming a Laplacian distribution over the error in likelihood form, and requires additional regularisation to ensure $\frac{\partial \psi(e)}{\partial e} \neq 0$ when $e = 0$, but provides increased robustness by constant weighting of residuals in comparison to the linearly increasing influence from the quadratic penalty. The Huber penalty provides a hybrid quadratic function for an inlier distribution, with an ℓ_1.

The Tukey bi-square function (**Tukey**, 1960), provides even great resilience to outliers. The penalty is defined piecewise as:

$$\psi(e) = \begin{cases} \frac{c^2}{6}(1 - [1 - (\frac{e}{c})^2]^3) & \text{if } |e| \leq c\,, \\ \frac{c^2}{6} & \text{if } |e| > c\,, \end{cases} \tag{8.23}$$

with a piece-wise influence function, ψ' :

$$\psi'(e) = \begin{cases} e[1 - (\frac{e}{c})^2]^2 & \text{if } |e| \leq c \,, \\ 0 & \text{if } |e| > c \,. \end{cases} \tag{8.24}$$

The tuning parameter of the bi-square function is set to $c = 4.685\sigma$, and for the Huber parameter in Equation (3.47) is set to $\alpha = 1.345\sigma$, using the estimated standard deviation of the error σ. In practice we provide a parameter for a user to alter the parameter. We have found that both the Huber and Tukey penalisation functions provide effective tracking performance in the presence of outliers, and provide demonstrations of their effectiveness in this setting later on in this chapter.

8.2.5 Illumination robustness

All of the error functions described in the preceding sections assume brightness constancy, and while the use of a photometrically calibrated camera mitigates variation caused by changes in exposure time it can not account for variation caused by changing global illumination, or when viewing surfaces which are not perfectly Lambertian.

Development of direct methods that capture a larger range of appearance variation can be traced back to the original Lucas-Kanade paper where the generative model is extended to include linear appearance variation modelled using a set of basis functions that modify the original template image (**Lucas and Kanade**, 1981). Optimisation is then performed on the extended set of parameters including the transform and coefficients for each of the appearance basis functions. State of the art direct alignment approaches that can cope with dynamic lighting scenarios go much further, obtaining illumination-invariant tracking by exploiting $9D$ spherical harmonic linear representations of the image (**Xu and Roy-Chowdhury**, 2007, 2008).

However, such explicit appearance modelling is unable to handle complex shadowing causing local appearance changes as well as more intricate inter-surface reflections that occur in non-Lambertian scenes. A more computationally efficient approach attempts to first pre-process the images to obtain illumination-invariant input which can then be aligned under the brightness constancy assumption, without any increase in parameters. One of the simplest models of appearance change assumes the warped reference is also altered by a global bias b and gain G: $e(u, \mathbf{x}) = (\mathcal{I}_l(\mathbf{w}(u, \mathbf{x})) \cdot G + b) - \mathcal{I}_r(u)$, which can be removed simply by pre-processing both the reference and live frame to have zero mean with unit variance.

Invariance to local illumination changes can be similarly achieved by pre-processing the images to remove the low frequency variations that break the brightness constancy assumption. This can be performed by subtracting the *smooth texture* component from a structure-texture decomposed image (**Wedel et al.**, 2009). Alternatively, the image Laplacian energy $|\nabla^2 I|_2$ removes low frequency image components. **Irani and Anandan** (1998) noted that while the Laplacian energy at a pixel is invariant to contrast inversion, gradient information useful for alignment is lost when using such a rotationally symmetrical operator. Instead they performed alignment with a vector image composed such that each component is the magnitude of one dimension of the the image derivative. They further extended the simple per pixel error to using a patch based normalised correlation, resulting in impressive multi-modal image alignment.

In practice we find that a simple pre-processing method suffices when using a photometrically calibrated camera within a small indoor workspace scenario, and in many scenarios we find there is no need to use pre-processing other than photometric normalisation given the calibrated camera. When it is deemed necessary we therefore filter out low frequency image content by subtracting from each image in the image pyramid, a Gaussian convolved version of the image with a filter twice the standard deviation of the filter used in the subsampling operation. In the following subsection we outline the full dense tracking pipeline which includes a further mechanism to increase robustness to illumination change.

8.2.6 Frame to Model Tracking Pipeline

The complete model based tracking pipeline is outlined in Figure (8.2). We begin estimation of the live pose $\tilde{T}_{wl}$ by initialising the parameters directly with the previous frames pose T_{wp}, equivalent to a constant position motion model. Then, relative to the previous image $\mathcal{I}_p$ the *frame-to-frame* SO_3 transform is estimated using the rotation only warp $\mathbf{w}_{SO_3}$ (subsection 8.2.1). This rotation is composed with the live pose estimate. Using this updated live pose we fix a predictive reference frame $T_{wr} \leftarrow \tilde{T}_{wl}$ and compute photometric and geometric predictions $\mathcal{I}_r$ and $\mathcal{D}_r$ with which to perform *frame-to-model* SE_3 estimation using the full translation and rotation warp $\mathbf{w}_{SE_3}$ (subsection 8.2.2), leading to the incremental update of the reference frame and new model prediction. There is a potential benefit in iterating the outer loop prediction-optimisation steps where the photometric and geometric predictions are re-rendered using the optimised live pose parameters. However, we find the using the prior frame-to-frame SO_3 optimisation to obtain the initial transform $\tilde{T}_{wp} \leftarrow T_{wp}[\tilde{R}_{pl}|\mathbf{0}]$ typically results in the first predictive model estimate being sufficiently close to the live pose that iterations of the outer loop do not substantially reduce error further.

The geometry independent frame-to-frame rotation estimation typically reduces the dis-

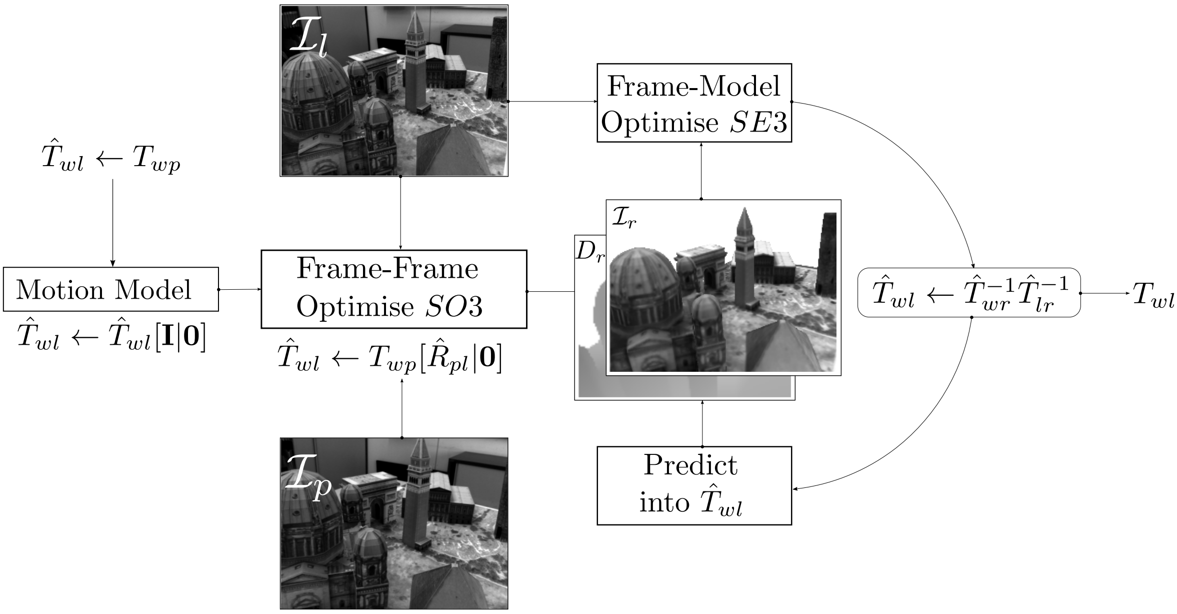

$\mathcal{I}_l$
Frame-Model Optimise $SE3$
$\hat{T}_{wl} \leftarrow T_{wp}$
$\mathcal{I}_r$
D_r
Motion Model
$\hat{T}_{wl} \leftarrow \hat{T}_{wl}[\mathbf{I}|\mathbf{0}]$
Frame-Frame Optimise $SO3$
$\hat{T}_{wl} \leftarrow T_{wp}[\hat{R}_{pl}|\mathbf{0}]$
$\hat{T}_{wl} \leftarrow \hat{T}_{wr}^{-1}\hat{T}_{lr}^{-1}$
T_{wl}
Predict into $\hat{T}_{wl}$
$\mathcal{I}_p$

Figure 8.2: Full dense SE_3 frame-to-model tracking pipeline. We track each live frame $\mathcal{I}_l$ against a current predicted reference frame comprising predicted depth map $\mathcal{D}_r$ and photometric prediction $\mathcal{I}_r$. We initialise the SE_3 optimisation using an estimate of the relative rotation between the live and previous frames using the coarse to fine SO_3 optimisation.

parity between corresponding image elements efficiently, reducing the number of iterations required with the more computationally expensive 6 parameter SE_3 transform while increasing the basin of convergence for the dense tracking pipeline. Furthermore, since the whole images are used, photometric appearance variations caused by illumination or non-Lambertian surfaces are lessened. A further route to robust tracking within the direct model-based tracking framework, is to construct a reference frame using the previous frame image in combination with the model geometry predicted into that frame. Since consecutive frames are likely to share local degradation due to illumination changes and motion blur, this frame-to-frame hybrid model can increase robustness for SE_3 tracking. This technique was recently demonstrated by **Comport et al.** (2011) to provide a good initialisation to the more standard frame-to-model tracking. In practice we did not find that this provided significant improvement over the frame-to-frame $SO3$ pose initialisation.

In our pipeline, the dense photometric prediction required for SE_3 tracking can be computed either through direct raycasting when photometric fusion is used, or by the key-frame based view dependent texture mapping mechanism. As detailed in Section (6.4.2) the key-frame approach provides a sparse unstructured light-field approximation of the scene, capturing useful predictions of non-Lambertian appearance changes. Tracking using this approximation is therefore of interest and is related to the method developed by

Heigl et al. (2000), where tracking is achieved by performing particle filtering (**Isard and Blake**, 1996), using a likelihood model based on photometric predictions obtained from a free-form light field representation of the environment (**Heigl et al.**, 1999). Samples from the camera pose distribution in the particle filter produce photometric predictions from the light field, where re-weighting of the particles can be achieved by computing a simple per-pixel difference with the live image, leading to the updated posterior distribution. From this potentially multi-modal distribution they extract the maximum likelihood estimate of the camera pose.

Basic Accuracy Comparison

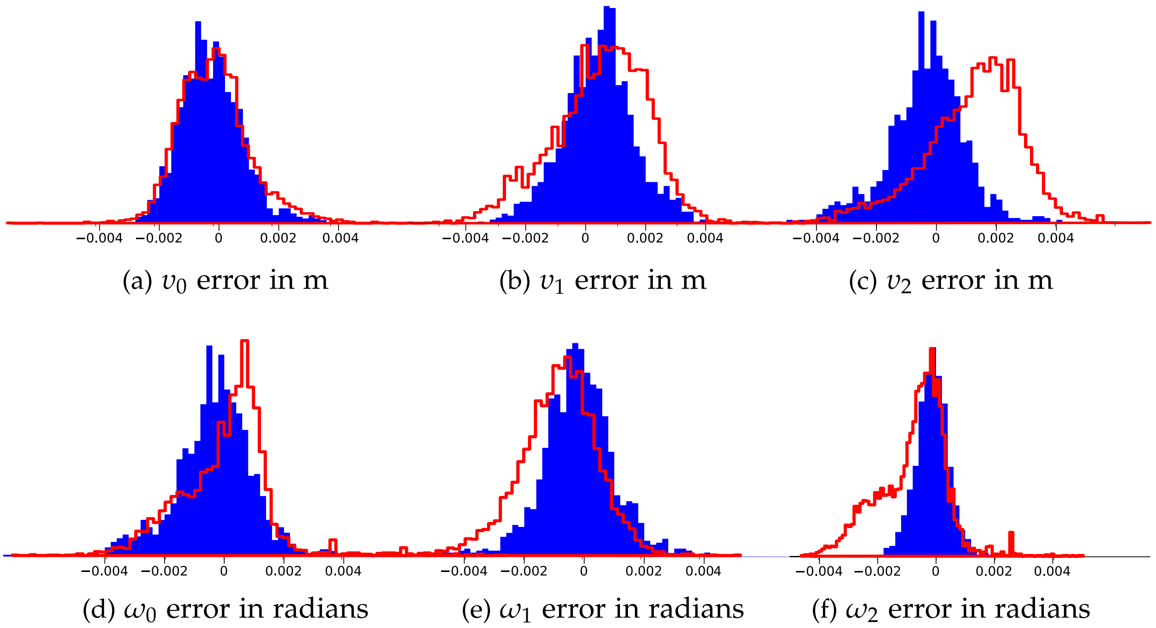
In Sections (4.5.8), (5.2.5) and (7.5), we evaluated the live dense reconstruction pipeline described in Chapter (7) using a video dataset of the City of Sights (**Gruber et al.**, 2010). In that experiment the camera trajectory was estimated using the bundle adjustment based PTAM system (**Klein and Murray**, 2008). The reconstruction accuracy was evaluated in Section (7.5) and shown to be near the limit of the physical paper model construction accuracy of $3mm$ RMS error. In this experiment we use the dense reconstructed model together with the PTAM camera pose for each frame of the $\approx 2''30'$ video sequence, and treat both as pseudo ground truth data. We compare this trajectory with that obtained using the direct $SE3$ tracking method described in this section, where the geometric prediction was obtained using direct ray-casting on the volumetric TSDF, described in Section (6.3). Here we are interested in evaluating basic accuracy of the direct tracking approach relative to the feature-based psuedo ground truth. For each frame of video sequence we therefore estimate a pose $T_{w\text{-fuse}}$ using photometric prediction with the fusion approach from Section (6.4.1), and a pose $T_{w\text{-vdtm}}$, using the key-frame based view dependent texturing predictions from Section (6.4.2). In this experiment we construct the photometric model (either through fusion or by dropping texture key-frames) using the pose estimated from PTAM. This is to enable comparison of the direct tracking approach together with either of the prediction methods in the best possible conditions. We compare these two trajectories against the PTAM pose at each frame $T_{w\text{-ptam}}$, by computing a relative $\mathfrak{se}3$ parameters from the relative PTAM to photometric fusion trajectory:

$$\mathbf{x}_{\text{ptam-fuse}} = \log(T_{w\text{-ptam}}^{-1} T_{w\text{-fuse}}) \ , \tag{8.25}$$

and likewise for the key-frame texturing predicted sequence:

$$\mathbf{x}_{\text{ptam-vdtm}} = \log(T_{w\text{-ptam}}^{-1} T_{w\text{-vdtm}}) \ . \tag{8.26}$$

Here, $log : SE3 \mapsto \mathfrak{se}3$ is the matrix logarithm that returns the 6 parameter vector representing the 3 element axis-angle rotation ω, and the inter-frame translation vector v. The



(a) v_0 error in m

(b) v_1 error in m

(c) v_2 error in m

(d) ω_0 error in radians

(e) ω_1 error in radians

(f) ω_2 error in radians

Figure 8.3: Histograms of relative frame error translation in meters (a,b,c) and rotation in radians (d,e,f) for the Graz City of Sights video dataset using the direct tracking approach, and comparing the method when either photometric fusion (blue) or key-frame based view dependent texture mapping (red line) photometric prediction was used, error is relative to a psuedo ground truth estimated using PTAM. We plot the histograms of the $\mathfrak{se}_3$ elements from the relative transform computed between the psuedo ground-truth PTAM pose and the two tracjectories obtained using direct frame-model tracking from Section (8.2.6), details for the error computation performed are given in the main text. We note that the resulting trajectory from either of the photometric prediction mechanisms results in a pose which induces sub-pixel accurate flows given the model geoemtry.

resulting histogram of each component over the sequence demonstrates that the estimated pose for both methods of prediction with the frame-to-model tracking pipeline generate high quality pose estimates, shown in Figure (8.3).

Since the estimation is obtained from a monocular sequence, we note that the translation component of each frame to world transform needed to be scaled to obtain a translation units in meters which was possible given the scale factor estimated between the reconstructed and ground-truth City of Sights model in Section (7.5). Both methods of photometric prediction yield a pose estimate that is close to the PTAM pose. We note that the photometric fusion approach to texture prediction surprisingly yields a less skewed error distribution over all 6 pose parameters relative to PTAM. However, given this level of accuracy further experimentation is required on a synthetic dataset for which there is a ground truth pose and geometry available since the error component is likely to be the result of error not only in the directly estimated trajectories but also in the PTAM pseudo ground truth.

This experiment shows the basic capability of high quality direct tracking given a model reconstructed and textured using the pipelines detailed in the previous chapters using the online pose estimate provided by PTAM. In Sections (9.2) and (9.4) we will go on to demonstrate dense reconstruction results obtained where we instead interleave LDR with direct camera tracking using the current partial reconstruction, demonstrating the completely dense tracking and mapping pipeline.

8.3 Direct Depth Image Tracking

There are situations where the assumptions made by the passive image based tracking framework described in the preceding section do not hold, a trivial case of which is when there simply is no useful measurement available, such as in complete darkness. More pervasive is the low level and dynamic lighting scenario presented by many modern living rooms with large bright screens that continue to provide challenges to real world visual SLAM systems. In other scenarios the SLAM system must cope with active dynamic illumination in the scene. One recent example (**Jones et al.**, 2010) demonstrates a mixed and augmented reality application where a sensor pose must be tracked for the purposes of geometry aware projection. In such an application, knowledge of the scene geometry and live sensor pose enable live renderings of a synthetic dynamic scene to be projected onto the surfaces that provide an illusion of being correctly placed 3D objects to a observer.

Very recently, commodity depth cameras that produce high quality active depth sensing have become available. Cameras that use either time of flight or structured light technology typically compute a 30fps VGA resolution depth image stream. In particular the Microsoft Kinect and Asus Xtion Pro devices provide depth camera solutions that achieve an active sensing range of 0.4 to 7 meters with an error variance which scales with depth of approximately 1% of the true depth measurement over the range. Moreover, the measurements are extremely robust to dynamic lighting, providing high quality dense depth maps for a range of textured or homogeneous surfaces with the exception of highly specular mirror like surfaces. It is therefore highly desirable to perform live dense reconstruction and dense SLAM with such devices. In this section we look at tracking from an available dense reconstruction using only geometric surface predictions from the model and only surface measurements obtained from a commodity depth camera. We will go on to make use of this direct depth camera tracking in Section (9.3) where we present KinectFusion, a fully dense SLAM system that combines the real-time volumetric TSDF reconstruction with direct tracking.

Successfully aligning a set of $3D$ point clouds or depth images forms the basis of many successful robot navigation and map building systems in robotics (**Thrun et al.**, 2005). In computer graphics, scan alignment is a critical component of model reconstruction pipelines used in augmented reality applications in film, computer-assisted surgery, crime scene modelling (**Sansoni et al.**, 2009) and digital archaeology (**Levoy et al.**, 2000).

Whether tracking a depth camera against a known environment map within a SLAM application or aligning scans of objects for reconstruction in a object modelling pipeline, techniques for automatic registration of 3D data sets lie on a similar spectrum to those outlined for passive tracking. To achieve fully automatic registration between frames where no information on relative alignment is available, pairwise *surface matching* pipelines perform $3D$ feature extraction and matching to obtain $3D - 3D$ point correspondences and then solve for parameters of a specified transform such as SE_3 that minimises matched correspondence distance (**Huber and Hebert**, 2003). A number of fully $3D$ descriptors that work only on surface geometry have been development for such pipeline which forms the basis of $3D$ object recognition. Examples include spin images (**Johnson and Hebert**, 1999); point signatures (**Chua and Jarvis**, 1997); and harmonic shape images (**Zhang and Hebert**, 1999); all of which attempt to produce a local transform invariant shape descriptor for a given region of the surface model or measurement. Matching of descriptors proceeds using a chosen metric within a robust fitting frame-work such as RANSAC **Fischler and Bolles** (1981); **Horn** (1987) produced the original solution for obtaining the rigid body transform from known $3D - 3D$ point correspondences.

When an initial transform estimate is available, blind surface matching is not need and incremental *surface alignment* strategies can minimise a pairwise surface error objective function similar to the photometric error based direct methods. The previously described surface matching pipelines often make use of such direct alignment strategies to obtain higher accuracy estimates, and since we make use of the strong prior on the camera trajectory available from tracking real-time depth map stream, we focus solely on direct methods here.

Horn and Harris (1991) introduced the range image analogue of the brightness constancy constraint equation known as the *elevation rate constraint equation* and produced the first direct alignment method for use explicitly with range images. They obtained 6DoF pose estimation using a least squares minimisation of the range error induced by motion between two frames using the spatio-temporal derivatives of the whole image error. Although the method was specifically designed to make use of small baseline motion and required no explicit correspondence matching between frames, researchers have only relatively recently revisited this direct formulation that was independently anticipated by **Gruen** (1985) and used by **Gruen and Akca** (2005). Instead researchers found increased performance and flexibility in teasing apart the implicit correspondence assumption made in a using spatio-temporal range derivative into a modularised pipeline known as Iterative Closest Points (ICP). In the following section we describe this well-known surface alignment strategy with the specific modifications that make it suitable for efficient real-time depth camera tracking.

8.3.1 Iterative Closest Point Optimisation

Given two points clouds with an initial transform estimate, the iterative closest point algorithm, developed independently by several researchers **Besl and McKay** (1992); **Chen and Medioni** (1992) and (**Zhang**, 1992), interleaves two basic components until convergence: **1.** Estimation of correspondences between the point clouds based on a measure of closest proximity (distance); **2.** Optimisation over the relative transform parameters that results in minimisation of the sum over the distance metric between those correspondences.

Rusinkiewicz and Levoy (2001) provided a concise overview of the many variations of the above two step algorithm that have been introduced. While the basic pipeline was developed over two decades ago, the core of any modern iterative surface alignment strategy retains these key features. Below we break the two step algorithm into finer grained components, and refer to original taxonomy produced by **Rusinkiewicz and Levoy** (2001) for the numerous alternative algorithms available within each component.

1. Sampling points: Rather than using all points in both meshes, sampling strategies can be used to reduce the set of points that will used in the rest of the pipeline. While uniform or random sampling can reduce the set to some predefined fraction of the total points (**Turk and Levoy**, 1994), more sophisticated sampling strategies make a selection based on some measure of local distinctiveness of a points region similar to feature extraction mechanisms and can be designed to increase the stability of the chosen error metric that will ultimately be minimised (**Gelfand et al.**, 2003).

2. Matching samples: In contrast to a full feature extraction and matching pipeline, the core of the original point based ICP algorithm uses a simple heuristic to obtain correspondences: given a sample in one point cloud, the closest point under a Euclidean distance in the other is sought (**Besl and McKay**, 1992). Here the distance refers to a geometric distance rather than a distance of an extracted descriptor vector. Closest point computation is computationally demanding and can be accelerated by storing the point cloud data in a k-d tree form requiring $\mathcal{O}(\log n)$ complexity per point look-up. Later in Section (8.4.3) we explore use of the volumetric signed distance function to avoid building such an acceleration structure within a dense reconstruction and tracking setting.

In the following section we instead make use of a projective closest point approximation introduced by **Blais and Levine** (1995) and **Neugebauer** (1997) that exploits the projective depth map structure. A projective closest point is obtained using an estimated relative transform by perspectively projecting a point from one depth map into the other and selecting the point at the projection location in the image. For small inter-frame motion this method has constant time complexity, and although clearly an invalid strategy for obtaining correspondences at depth discontinuities, holds well in practice for smoothly varying regions.

3. Correspondence weighting and rejection: Putative matches based the closest distance heuristic can be culled from or down weighted in the following error minimisation scheme if the associated points are deemed incompatible. We will further discuss basic compatibility heuristics based on geometric distance of the matched points in the next section.

4. Minimisation of an error metric: The parametrised transform between the point sets is then estimated by minimising the sum over all correspondence distances. If the selected pairs contain no incorrect correspondences and the error over each $3D$ point is Gaussian then the Euclidean point to point distance metric can be minimised using a number of closed form techniques (**Eggert et al.**, 1997). The original point to point distance based ICP developed by **Besl and McKay** (1992) used a sum of squared distances equivalent to a quadratic penalisation function with a Euclidean distance metric. **Zhang** (1992, 1994) replaced the quadratic penalisation with more robust error metrics. If normal estimates are available with each point, alternative distance metrics include point to tangent-plane (**Chen and Medioni**, 1992):

$$\text{Dist}_{\text{point-plane}} = \mathbf{n}_r \cdot (\mathbf{p}_0 - \mathbf{p}_r) \ , \tag{8.27}$$

which computes the shortest Euclidean distance of the point $\mathbf{p}_0$ from the plane defined by the unit normal $\mathbf{n}_r$ passing through the point $\mathbf{p}_r$. More recently **Segal et al.** (2009) introduced the plane-plane distance metric formulating a fully probabilistic account of the ICP process. All distance metrics can be further modified for optimisation of an incremental transform and solved using iterative non-linear minimisation techniques, making use of the robust estimation process when robust penalty terms are selected, as described in Section (8.2.4).

Fast ICP

Rusinkiewicz and Levoy (2001) performed a systematic evaluation of the variants of ICP looking at both computational complexity of the pipelines as well as practical performance measurements. Their evaluation led **Rusinkiewicz et al.** (2002) to a selection of components most suitable for a high speed small-baseline ICP alignment of two projectively acquired depth map measurements. They used this pipeline in the first real-time surface alignment implementation using point clouds produced from a structured light based depth camera. Their fast ICP pipeline consists of: **A.** Random sampling a sub-set of points from one mesh; **B.** Projective data-association of the source points; **C.** Rejection of pairs which exceed a given point-point distance threshold; **D.** Minimisation of the remaining sum of point-plane errors under a quadratic penalty.

Notably, this early live dense reconstruction system achieved a $10Hz$ rate of alignment of near VGA resolution depth scans. However, computational limitations resulted in the need to sample the point sets to reduce the computational load of both the data association and the resulting non-linear optimisation procedure.

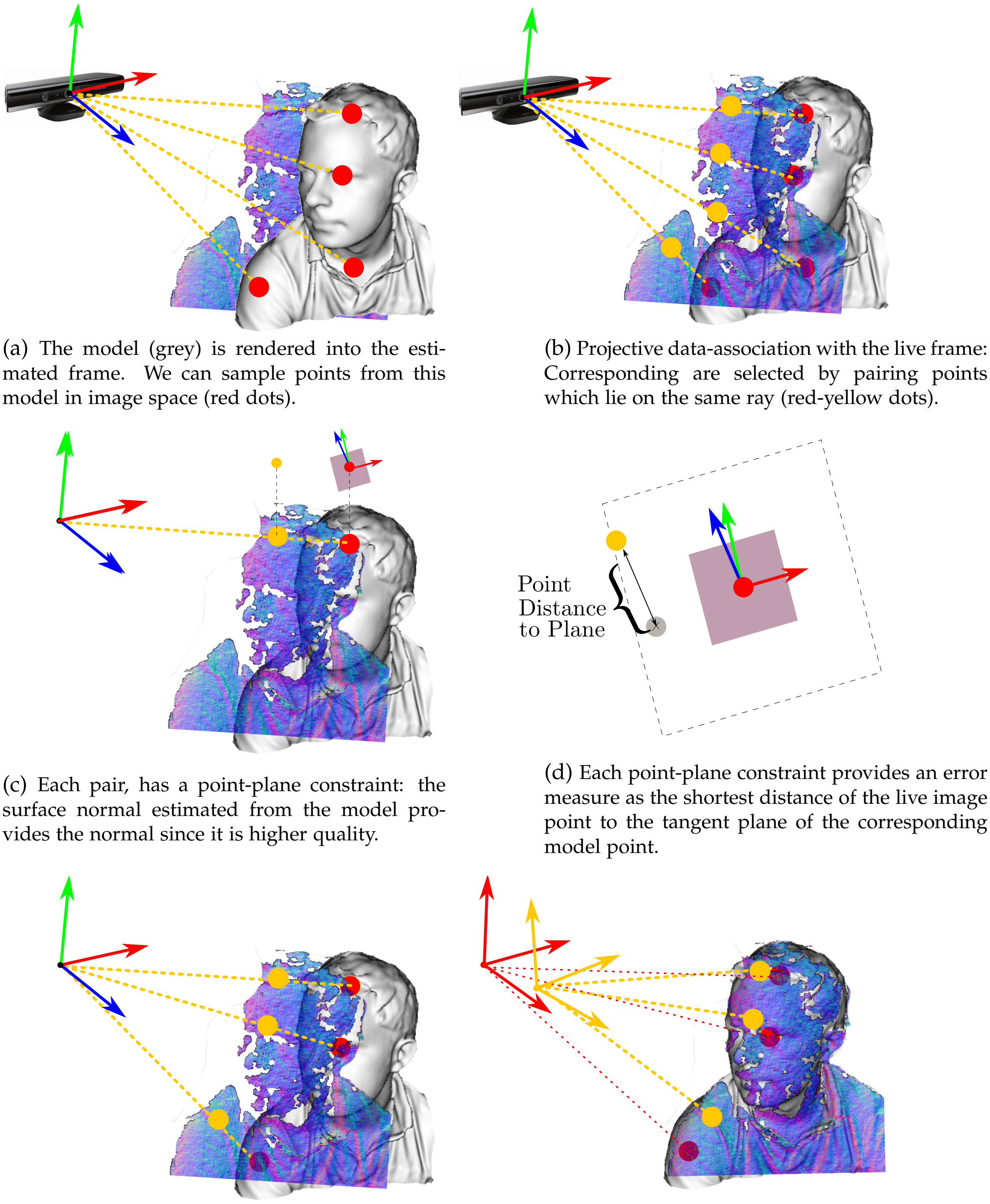
The Fast ICP pipeline is trivially paralellisable. We can therefore use the pipeline with all data available in a single depth frame in a GPGPU implementation, increasing robustness to outliers in the data. An illustrated overview of the fast ICP pipeline is given in Figure (8.4). We detail the optimisation procedure in the remainder of this section.

8.3.2 Projective Data Association and Point-Plane Error

Given an initial estimate of a live pose $\tilde{T}_{wl}$, and a dense surface model we can compute a depth map reference $\mathcal{D}_r$, with an associated normal map n_r and a geometry validity mask V_r described in Section (6.1.1). The depth map can be rendered using either the raycasting or mesh based geometric prediction mechanisms from Section (6.3). Initialising the relative reference frame to live frame transform $\tilde{T}_{lr}$ with $\tilde{R}_{lr} = \mathbf{I}_{3\times 3}$ and $\tilde{t}_{lr} = 0$, we proceed to align the measurement depth map $\mathcal{D}_l$ onto the surface model, combining projective data association with the point-plane error function in Equation (8.27) into a direct iterative scheme. To that end, we define a whole depth image error which we will optimise using the Gauss-Newton gradient descent approach:

$$E_w(\mathbf{x}) = \sum_{u \in \Omega} \psi\left(e(u, \mathbf{x})\right) \tag{8.28}$$

$$e(u, \mathbf{x}) = N_r^\top(u)(\exp(\hat{\mathbf{x}}) v_l(u') - v_r(u)) \,, \tag{8.29}$$



(a) The model (grey) is rendered into the estimated frame. We can sample points from this model in image space (red dots).

(b) Projective data-association with the live frame: Corresponding are selected by pairing points which lie on the same ray (red-yellow dots).

(c) Each pair, has a point-plane constraint: the surface normal estimated from the model provides the normal since it is higher quality.

(d) Each point-plane constraint provides an error measure as the shortest distance of the live image point to the tangent plane of the corresponding model point.

(e) Pairs fail a point-plane compatibility if the point-point Euclidean distance or normal-normal angle exceed thresholds.

(f) A Gauss-Newton based iterative gradient descent minimisation of the sum of point-plane error induced by the remaining pairs results in the new pose estimate.

Figure 8.4: Fast projective data-association based ICP steps. (a) Given the initial pose estimate we render the reference depth map which is used for multiple iterations of the fast ICP. Projective data association (b) followed by pair compatibility testing (c,d,e) and minimisation of the point-plane energy (f) are iterated until convergence.

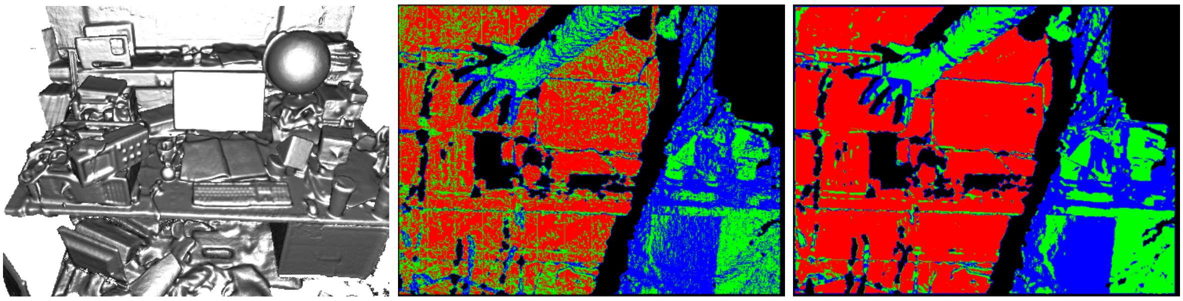


Figure 8.5: Example of point-plane outliers as person steps into a partially reconstructed scene shown rendered in (a). Outliers from compatibility checks (Equation 8.33) using a metric surface measurement from Kinect (b). $\Omega(u) \neq$ null are shown in red, $V_l(u')$ are shown in black and outliers are shown for incompatibility in normal (green) and both distance and normal (blue). In (c) we demonstrate the improved data association by performing tracking on a bilateral-filtered version of the raw depth map used in the *KinectFusion* application described in Chapter (9).

Here $v_l(u')$ is the projectively data associated live vertex at estimated pixel correspondence u' for the reference vertex $v_r(u)$ at pixel u with normal $N_r(u)$:

$$v_r(u) = K^{-1}\dot{u}\mathcal{D}_r(u) \ , \tag{8.30}$$

$$v_l(u') = \tilde{T}_{rl}K^{-1}\dot{u}'\mathcal{D}_l(u') \ , \tag{8.31}$$

$$u' = \mathbf{w}_{se3}(u, \mathbf{x}_0) = \pi(K\tilde{T}_{lr}v_r) \ . \tag{8.32}$$

Here $\mathbf{w}_{se3}$ is the same warp function as used previously in photometric tracking in Equation (8.17), evaluated at the current transform estimate $\tilde{T}_{lr}$, but where the incremental transform $\exp(\hat{\mathbf{x}})$ has been moved outside of the perspective projection. The warp function provides the *projective data association* by computing the corresponding pixel in the live depth map, pairing points between the reference and live images which currently lie on the same ray, illustrated in Figure (8.4b).

The above error function makes the assumption that all pixels in the reference frame have correspondence under the initial transform in the live depth image. This is not correct for surface elements that are occluded under the estimated transform. The use of a robust error metric in (8.28) can mitigate explicit modelling of such occlusion errors. If the depth images are metric measurements of the surface, then we can instead use an explicit rejection step using informative thresholds on the expected error of a matched pair such that correspondences should have a point-to-point distance and normal-to-normal angular difference within some thresholds, illustrated in Figure (8.4e).

To achieve this we restrict the set of vertex correspondences $\{v_l(u'), v_r(u)|\Omega(u) \neq \text{null}\}$ by testing the predicted and measured vertex and normal for compatibility: a threshold on the distance of vertices and difference in normal values suffices to reject grossly incorrect correspondences, also illustrated in Figure 8.5:

$$\Omega(u) \neq \text{null iff} \begin{cases} V_l(u') & = & 1, & \text{and} \\ \|\hat{T}_{rl} v_l(u') - v_r(u)\|_2 & \leq & \epsilon_d, & \text{and} \\ \langle \hat{R}_{rl} N_l(u'), N_r(u) \rangle & \geq & \epsilon_\theta. & \end{cases} \tag{8.33}$$

Here ϵ_d and ϵ_θ are threshold parameters of the system. The quadratic penalisation function can then be used with the remaining inlier set $\Omega(u) \neq \mathit{null}$.

Computing a Gauss-Newton update as outlined in Section (3.3) we derive $J(u)$ with the point-plane error function in Equation (8.29):

$$J(u, \mathbf{x}) = \frac{\partial e(u, \mathbf{x})}{\exp(\hat{\mathbf{x}}) \tilde{T}_{rl} v_l(\mathbf{w}(u, \mathbf{x}))} \frac{\exp(\hat{\mathbf{x}}) \tilde{T}_{rl} v_l(\mathbf{w}(u, \mathbf{x}))}{\partial \mathbf{x}} \, . \tag{8.34}$$

Defining $(x, y, z)^\top = \tilde{T}_{rl} v_l(\mathbf{w}(u, \mathbf{x}))$ the resulting 1×6 gradient vector for pixel u is computed as:

$$J(u, \mathbf{x}) = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}^\top \begin{pmatrix} 1 & 0 & 0 & 0 & z & -y \\ 0 & 1 & 0 & -z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \end{pmatrix} , \tag{8.35}$$

which together with the error function derivative evaluated at $\mathbf{x}_0 = 0$ results in a linear system which is solved for $\mathbf{x}$ and taken to an element SE_3 via the exponential map. Composition onto the initial transformation estimate computes the new solution point:

$$\tilde{T}_{lr} \leftarrow \exp(\hat{\mathbf{x}}) \tilde{T}_{lr} \, . \tag{8.36}$$

Stability and validity check for transformation update As inter-frame sensor motion increases, the assumptions made in both linearisation of the point-plane error metric and the projective data association can be broken. Also, if the currently observable surface geometry does not provide point-plane pairs that constrain the full 6DoF of the linear system then an arbitrary solution within the remaining free degrees of freedom can be obtained. We therefore perform a check on the null space of the normal equations to ensure it is adequately constrained. We also perform a simple threshold check on the magnitude of the incremental transform parameters $\mathbf{x}$, to ensure the small angle assumption was not drastically broken. If either test fails, tracking is stopped and the system must be relocalised. We describe a basic relocalisation mechanism in the final section of this chapter.

8.4 Direct Signed Distance Function Tracking

The projective data association based ICP pipeline makes use of the depth map structure available for surface measurements obtained from depth cameras and stereo systems. If, however, no such supporting data structure is available, acceleration of the closest point computation is required to reduce the $\mathcal{O}(nm)$ computational cost for two meshes with m and n points.

Fitzgibbon (2001) provided an important insight into the ICP algorithm, and reformulated ICP by replacing the explicit data correspondence followed by error minimisation steps with a direct Newton-style minimisation over the distance metric embedded in the closest point computation:

$$\underset{T_{lr} \in SE_3}{\operatorname{argmin}} \sum_{u_l \in \Omega} \min_{u_r} \left\{ \psi \left(v(u_r) - T_{lr} v(u_l) \right) \right\} \ . \tag{8.37}$$

Furthermore, he noted the importance to application in dense reconstruction, where the majority of systems that require mesh alignment to bring partial scans into a common frame of reference utilise some form of distance function approximation to achieve a global reconstruction. One example of which is the volumetric SDF fusion approach described in Chapter (6). **Mitra et al.** (2004); **Pottmann et al.** (2004) developed the same direct optimisation strategy but using efficient approximations to the full distance transform, including the octree based d^2-tree approximation to the distance field (**Leopoldseder et al.**, 2003). In either case, optimisation of Equation (8.37) can be achieved using a direct numerical gradient descent on a discretised distance transform of the reference surface.

Since a truncated signed distance field is provided in real-time at the core of the dense reconstruction mechanism detailed in the previous chapters, it is of interest to investigate the potential of using it directly for dense tracking, bypassing the need to compute a geometric prediction. In the following section we derive the Gauss-Newton step for the direct distance function tracking with small modifications to account for the truncation region in which no valid distance is available. In Section (8.4.3) we then outline a standard procedure for computing the exact Euclidean distance function but seeded by the truncated SDF. We then compare the convergence performance for the point-plane and both truncated and exact direct distance function tracking methods.

8.4.1 Distance Transform Error function

The truncated signed distance transform S has zero crossings at the reconstructed surface interface. Given a live depth map measurement $\mathcal{D}_l$ and an estimated transform into the world (model) frame $\tilde{T}_{wl}$ we can define the error induced by the incorrect sensor pose by directly summing up the magnitudes of the SDF values located at the transformed points in S.

A vertex in the live image at pixel u is incrementally transformed by $\tilde{T}_{wl}\exp(\hat{\mathbf{x}})$ to obtain the single pixel error: which can then be indexed in S:

$$e(u,\mathbf{x}) = S[c(\tilde{T}_{wl}\exp(\hat{\mathbf{x}})v_l(u))] \ , \tag{8.38}$$

$$v_l(u) = K^{-1}\dot{u}\mathcal{D}_l(u) \ , \tag{8.39}$$

$$c(v) = S_q\frac{v^\top - S_{0,0,0}}{S_r} \ . \tag{8.40}$$

$c(v)$ transforms a continuous point measurement v from reconstruction units within the reconstruction volume with origin $S_{0,0,0}$ and extent or range $S_r = S_{x,y,z} - S_{0,0,0}$ to the discretised voxel grid with resolution $S_q = (q_x, q_y, q_z)$.

An optimal pose will transform the whole surface measurement from the live to the world frame, where all indexed points in S have an absolute distance value near zero, modulo errors in S and $\mathcal{D}_l$. We therefore define the whole image energy in the live frame that we want to minimise:

$$E_{tsdf}(\mathbf{x}) = \sum_{u\in\Omega}\psi(e(u,\mathbf{x})) \ . \tag{8.41}$$

Again we can derive the Gauss-Newton update using the approximation to the second order Taylor series expansion of E_{tsdf} using Equation (3.35). The resulting minimisation requires computation of $J(u)$ for the error function in Equation (8.41):

$$J(u) = \frac{\partial e(u,\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial S[c(v)]}{\partial v}\frac{\exp(\hat{\mathbf{x}})\tilde{T}_{rl}v_l(u)}{\partial \mathbf{x}} \ . \tag{8.42}$$

The current measurement transformed into the global frame is $(x,y,z)^\top = \tilde{T}_{wl}v_l(u)$ and the resulting 1×6 gradient vector for pixel u in the live frame is computed as:

$$J(u,\mathbf{x}) = \frac{S_q}{S_r}\begin{pmatrix}\nabla_x S\\ \nabla_y S\\ \nabla_z S\end{pmatrix}^\top \begin{pmatrix}1 & 0 & 0 & 0 & z & -y\\ 0 & 1 & 0 & -z & 0 & x\\ 0 & 0 & 1 & y & -x & 0\end{pmatrix} \ . \tag{8.43}$$

The gradient ∇S is evaluated using finite differences, and since the truncated signed distance function has $\nabla S = \mathbf{0}$ within the positive and negative truncated regions and is not defined in unobserved regions care must be taken to use only valid TSDF values for the gradient computation. The scale factor $\frac{S_q}{S_r}$ transforms the signed distance function gradient from the discretised grid units into reconstruction units.

The resulting 6×6 linear system can be solved through a Cholesky decomposition:

$$\mathbf{x} = -\left(\sum_{u \in \Omega} J(u, \mathbf{x})^\top J(u, \mathbf{x})\right)^{-1} \sum_{u \in \Omega} \psi'(e(u, \mathbf{x}) J(u, \mathbf{x})) \ . \tag{8.44}$$

The solution vector $\mathbf{x}$ is taken to an element in $SE3$ via the exponential map, and composed onto the initial transformation estimate to obtain the current solution point:

$$\tilde{T}_{wl} \leftarrow \tilde{T}_{wl} \exp(\hat{\mathbf{x}}) \ . \tag{8.45}$$

We note that at convergence to the true solution there is an equivalence between the direct tracking Jacobian in Equation (8.43) and the point-plane Jacobian in Equation (8.35) since the gradient of the SDF, which is perpendicular to the isosurface, is the surface normal.

8.4.2 Full Distance Transform

The above direct TSDF tracking approach adds only a minor modification to the optimisation presented in **Fitzgibbon** (2001), ensuring only valid gradient computations are performed within the SDF region. Unfortunately, the truncation region presents a more basic problem for achieving high quality tracking using the TSDF representation from Chapter (6). In reconstruction, the truncation region must be kept small to ensure front and back surfaces do not interfere. This enables reconstruction of thinner surfaces and finer detail and serves to reduce distortion due to the projective approximation to the TSDF used when constructing the field. Since truncation of the distance field effectively provides only a thin band of usable SDF gradient values near the reconstructed surface, the available basin of convergence for the optimisation is drastically reduced. We now outline the method to construct the true Euclidean distance field from the TSDF and demonstrate the increased ability to achieve convergence from large baselines that it affords.

8.4.3 Computing the Euclidean Distance Transform

There is a vast literature on computing distance transforms, arising from three main research communities in computational physics, computer vision and computer graphics. In each area this has given rise to efficient approximations for computation of the distance transform in multiple dimensions given an initial surface boundary (**Jones et al.**, 2006).

Since we will utilise the complete distance transform for tracking purposes only we can ignore the sign of the distance that is crucial in the dense reconstruction framework since the error to be minimised is a function of the positive distances. We can define the distance transform over a discrete grid $\mathcal{G}$, computing the distance at each grid location $p \in \mathcal{G}$ to a known discretised surface $P \subset G$:

$$S(p) = \min_{q \in P}(d(p,q) + f(q)). \tag{8.46}$$

Here $d(p,q)$ defines a distance metric between the point p and q and when $f(p)$ is the indicator function $1(q)$:

$$1(q) = \begin{cases} 0 \text{ if } q \in P \\ \infty \text{ otherwise} \end{cases} \tag{8.47}$$

the result is the classic distance transform.

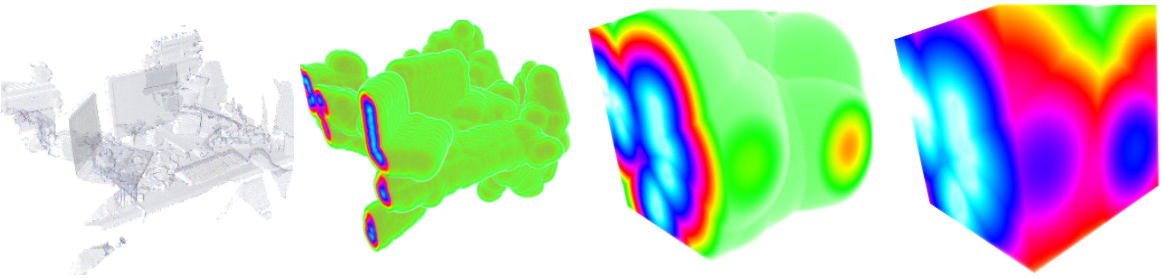
The simplest brute force approach directly evaluates all point pairs in $d(p,q)$, storing the minimum obtained for the grid point p. The complexity is $\mathcal{O}(mn)$ for m reference points on a grid with n voxels, which for the size of voxel grid and mesh vertices required to represent the surface presents an infeasibly large problem to solve in near real-time even on modern GPGPU hardware.

When the distance metric is Euclidean, $d(p,q) = \|p-q\|_2$, a classic approximation from image processing introduced by **Borgefors** (1986) propagates the minimum distance from the surface boundary conditions out, called the Chamfer distance. The resulting distance field is relatively cheap to compute with complexity $\mathcal{O}(n)$, independent from the number of points representing the surface, but results in distorted fields due to metrication errors. The distortions can be decreased by using a larger mask region, with increasing computational expense.

Felzenszwalb and Huttenlocher (2004) established the relationship between the squared Euclidean distance transform and its minimum convolution operator, leading the way to an efficient $\mathcal{O}(n)$ computation of the exact Euclidean distance transform which we utilise here. The essence of the method is the computation of several $1D$ distance transform. Letting the $3D$ grid be $\mathcal{G} = (0, ..., h-1) \times (0, ..., w-1) \times (0, ..., d-1)$ the $3D$ squared Euclidean distance transform is:

$$S_f(x,y,z) = \min_{[i,j,k] \in \mathcal{G}}((x-i)^2 + (y-j)^2 + (z-k)^2 + f(i,j,k)), \tag{8.48}$$

$$f(i,j,k) = 1(i,j,k). \tag{8.49}$$



(a) Projective TSDF, truncation $\approx 5mm$.
(b) True EDT, truncation $\approx 60mm$.
(c) True EDT, truncation $\approx 300mm$.
(d) True EDT to boundary.

Figure 8.6: An example exact Euclidean distance transform for a partial reconstruction of a desktop scene, obtained using the truncated signed distance function integration approach from Chapter (6). In each figure we volume render the distance function, using a colour mapping to highlight the shape of the field moving away from the surface interface. In (a) we show the thin band of distance values obtained from the TSDF volume, it is clear that the thin band used for the reconstruction results in vast sparsity throughout the volume (white space), where no useful gradient is available for the direct tracking approach. In (b,c,d) we render the equivalent exact distance function truncating the distance to highlight the growing convexity of field away from the interface. Note that surface features present on each level set (constant colour) become more prominent nearer to the interface. This far-field behaviour of the full distance field leads to a large basin of convergence.

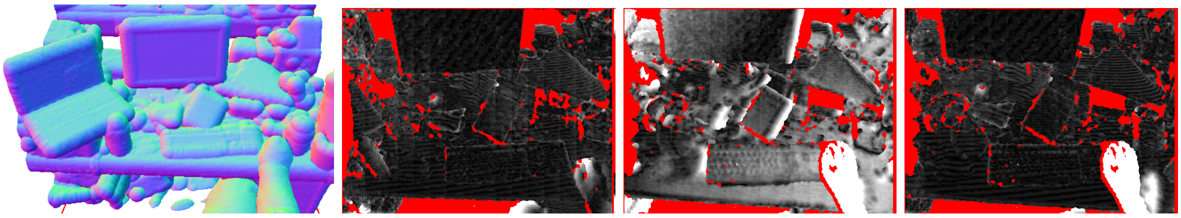
Noting that the squared Euclidean distance can be separated in each dimension, Felzenszwalb and Huttenlocher (2004) compute:

$$d_f(x,y,z) = \min_i((x-i)^2 + d_{f|i}(y,z)) \ , \tag{8.50}$$

$$d_{f|i}(y,z) = \min_j((y-j)^2 + d_{f|ij}(z)) \tag{8.51}$$

$$d_{f|ij}(z) = \min_k((z-k)^2 + f(i,j,k)) \ . \tag{8.52}$$

The resulting algorithm first computes the $1D$ distance transform for all rows of G along a chosen dimension for example along z in Equation (8.52). The resulting rows then define the new function $d_{f|ij}$ which replaces the initial indicator function f. The distance transform is then computed for all rows in a second dimension using $d_{f|ij}$, computing $d_{f|i}$ in Equation (8.51). Finally the distance transform is computed using $d_{f|i}$ on all rows of the remaining dimension in Equation (8.50). The result of this final transform is a squared Euclidean distance transform, which we convert using a square root back to the euclidean distance metric. The complexity of the algorithm which can be used for any grid with dimension k is $\mathcal{O}(kn)$ for n grid points and is trivially parallelisable since all rows are computed independently for each dimension. We seed the initial distance transform using the square of the thin band of SDF values from the TSDF removing the need to first extract and dis-



(a) EDT surface truncation.

(b) Error prior to outliers.

(c) ψ as quadratic penalty.

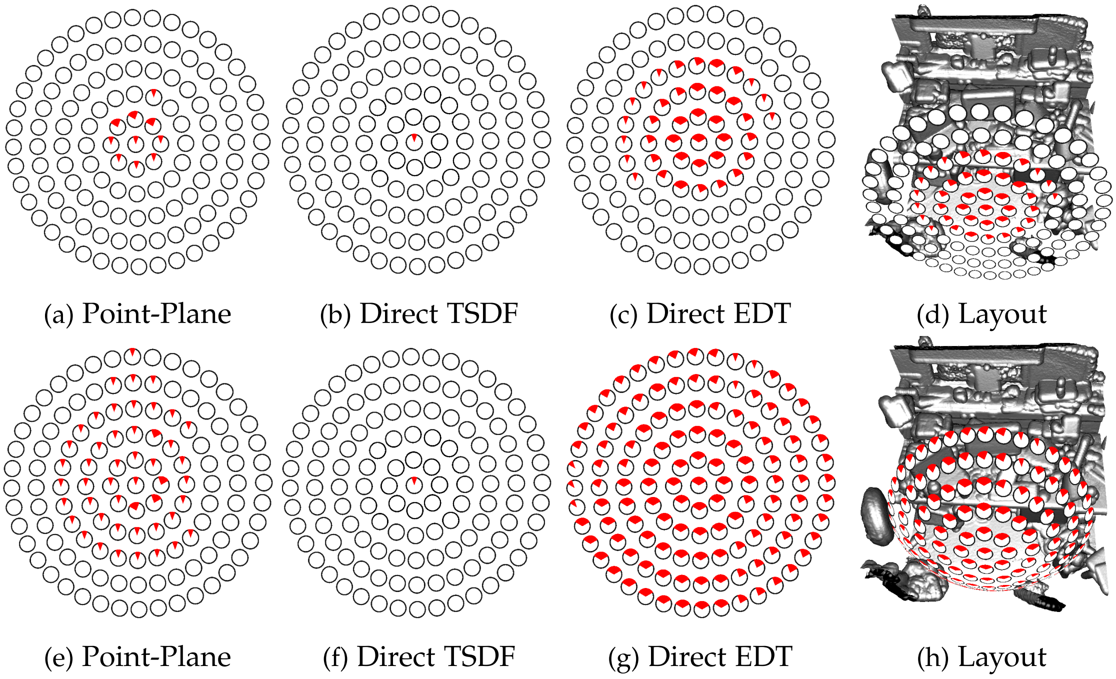
(d) ψ as Huber penalty.

Figure 8.7: Example tracking using the true distance field computed using the EDT on an office desktop scene. In (a) we render the scene showing the level set of the distance field at approximately $30mm$ truncation, resulting in the blob like appearance of the modelled objects. In (b) we show an example error image in the live frame, prior to motion of hand and arm shown in the bottom right corner of the reconstruction: red pixels indicate no valid surface measurement while the intensity of the remaining pixels indicate the absolute error computed in Equation (8.38). In (c,d) a hand in the scene moves presenting outliers to the static scene. In (c) the use of a quadratic penalty term results in a biased solution, while in (d) the Huber penalty is capable of down-weighting the outliers resulting in the correct pose estimate.

cretise the surface interface for computation of the indicator function $f(i,j,k)$ in Equation (8.52). Figure (8.6) illustrates the vast increase in available, valid, distance values obtained by computing the true distance transform for a partial scene reconstruction. In Figure (8.7) the EDT representation is used within the direct gradient descent approach from Section (8.4.1), and robustness to pose estimation in the presence of outliers is demonstrated using Huber penalisation.

8.4.4 Comparing Point-Plane, Direct TSDF and EDT Tracking

We will now compare the basin of convergence for each of the depth image tracking approaches. We note that the point-plane metric uses only the surface of a given reconstruction in the form a depth map and associated surface normal, and so together with the projective data association mechanism, correspondences can be obtained even when there is a significant error in the initial pose estimate, given a specific maximum error when using the point distance and normal compatibility tests in Equation (8.33). The direct tracking approach, does not make use of such explicit compatibility tests, using a robust penalty function to down-weight possible outliers. However, when applied to the volumetric TSDF it is clear that there is only a small region of error possible for a given initial pose estimate outside of which the the resulting transformation of the measurement vertices into the global frame will yield a non computable gradient. This problem is mitigated by computation of the full distance field.


(a) Point-Plane (b) Direct TSDF (c) Direct EDT (d) Layout

(e) Point-Plane (f) Direct TSDF (g) Direct EDT (h) Layout

Figure 8.8: Analysis of the basin of convergence for the direct depth map tracking approaches. Given the dense desktop scene reconstruction, shown in (d), and an initially known pose sampled at the center of the convergence funnel we perturb the pose to the sampled locations and run each algorithm using the measurement obtained at the initial location. We plot a red sector for successful convergence from the sample perturbation pose. In (a,b,c) we show the resulting convergence funnels for the pose sampling illustrated in (d), note the convergence funnel from (c) is rendered in the illustration. The sampling takes plots over a disc with translational increments of $100mm$ in the camera X-Y plane, and incremental rotation around the camera forward axis of 40 degrees. In (e,d,g) we plot the convergence funnel for the fixation pose layout shown in (h). Direct tracking using the true distance transform results in a vastly increased basin of convergence over the TSDF representation.

In Figure (8.8) we compare the convergence capabilities of each of the tracking approaches using a funnel of convergence analysis (**Mitra et al.**, 2004). In this analysis we reconstruct a scene (in this case an office desktop), and then given a surface measurement with known pose, we deterministically sample a range of pose perturbations that are used as the initial pose estimate in each of the iterative optimisation approaches. A larger basin of convergence, indicating increased tracking capability for large camera motion, results in the ability to converge to the correct known measurement pose from larger perturbations. We have evaluated two sampling strategies which empirically model two modes of real-time camera tracking use. In the first analysis perturbations to the initial pose are performed on a disc: with a translation in the camera $X - Y$ plane and rotation around the camera forward (Z) axis. This is similar to an exploratory motion, the sampling is shown in Figure

(8.8d). In the second analysis we perturb the camera to simulate fixating on a scene point, useful when performing augmented reality within a fixed region of the scene. At each point on the sampled motion arc we then further rotate the camera around its forward axis shown in Figure (8.8h). We use all valid measurements in each iteration of the optimisation, and allow the optimisation used in each approach a fixed maximum time within which to converge. Convergence is marked by the estimated relative pose to the ground truth being within a magnitude of $10mm$ translation and 1 degree rotation.

In summary we find that the true distance function provides a far wider basin of convergence in comparison to either the direct TSDF or dense point-plane ICP approaches. However, in a reconstruction setting, use of the true distance field would require twice the storage requirements (assuming the same resolution of volume is used). Unfortunately while direct tracking from the TSDF used in reconstruction presents an opportunity to remove that cost and gain the benefits of direct tracking, the convergence analysis shows the small basin of convergence possible when using the truncation settings for a high quality reconstruction. We therefore find in practice that the dense point-plane ICP tracking provides a useful basin of convergence without requiring extra storage. We make use of the point-plane ICP in the full dense SLAM KinectFusion system in Chapter (9).

8.5 Re-localisation

Direct approaches provide impressive tracking during rapid agile motion with concomitant resilience to measurement blur caused by motion or defocus and graceful degradation when faced with increasing homogeneously textured regions; but they can not prevent tracking failure in all scenarios. When tracking does fail visual SLAM systems that rely on incremental forms of tracking, including all of the direct approaches detailed in this chapter, must perform re-localisation of the sensor pose relative to the model.

Relocalisation is most difficult when no prior assumption about the live sensor pose is available or useful, including the possibility that live sensor measurements are outside of the predictive domain of the current model, known as the lost or kidnapped robot scenario in robotics. State of the art solutions to the general relocalisation problem, make use of efficient forms of feature based pose estimation also used in the tracking by detection pipeline discussed in this Chapter's introduction. The feature based pipelines are effective when the live measurement provide the minimally required to solve the camera pose with $n \geq \{3,4,5\}$ $2D-3D$ point correspondences (**Ess et al.**, 2007). If this is not possible, for example during exploratory motion when performing visual SLAM, an effective approach developed by **Eade and Drummond** (2008) unified loop closure and the relocalisation mechanism so that when tracking fails, a new map is initialised in which the live

Algorithm 1 Keyframe relocalisation based on (**Klein and Murray**, 2008)

1. Key-frame detection: Compute the zero mean sum of square errors (ZMSSE) between sub-sampled versions of the live image and each key-frame texture. The key-frame with the lowest image error is deemed the closest key-frame.
2. SE_2 **Direct Alignment:** An SE_2 transformation is estimated using direct ESM optimisation between the live frame and the closest key-frame.
3. SE_2 **to** SE_3 **upgrade:** Given a small selection of possible correspondences in the live frame, the estimated SE_2 transform is projected to the nearest pose in SE_3.
4. Increase $2D - 3D$ **correspondences:** Given the estimated 6DoF camera pose, the standard feature-based tracking pipeline is used to obtained further correspondences to furnish a non-linear Gauss-Newton gradient descent on the pose parameters.

pose is tracked. Adjacency of sub-maps can then be detected and jointed in a semi-offline manner by detecting shared features.

When information that restricts the likely pose of the sensor is available, the task is greatly simplified. In the particular scenario we are interested in, an assumption can be made that a human user is able to approximately reposition the sensor within some previously visited region. With basic feedback to the user regarding the success or failure of the relocalisation mechanism, a feedback loop between the user and the system can be set up to attempt to guide the user to reposition the sensor near to a previously visited region. In this interactive setting **Klein and Murray** (2008) demonstrated the effectiveness of a simple key-frame based relocalisation mechanism in their Parallel Tracking and Mapping system, which we now outline and extend.

8.5.1 Dense Surface Key-frame based relocalisation

Key-frame based relocalisation strategies use the ability to capture a distribution over likely pose parameters for the observed scene. A set of key-frames samples poses from the camera trajectory, resulting in a distribution that is application and scene dependent. Clearly a camera pose can not exist in physically implausible location, and occurs only by being moved there by the user. By storing a sampling of the trajectory poses together with a basic descriptor capturing the image data found at each pose, a lost camera can be relocalised by searching for the key-frame most similar in description to the live frame, followed by initialisation of an iterative camera tracking mechanism initialised using the pose of the key-frame, in the hope that the initial pose is close enough to live pose to enable tracking convergence. The key-frame re-localisation mechanism introduced by **Klein and Murray** (2008) and used in PTAM performs four consecutive steps outlined in Algorithm (1).

During browsing of the scene, we build the key-frame set using the same insertion mechanism described in Section (6.4.2) for use in the view dependent texturing pipeline, il-

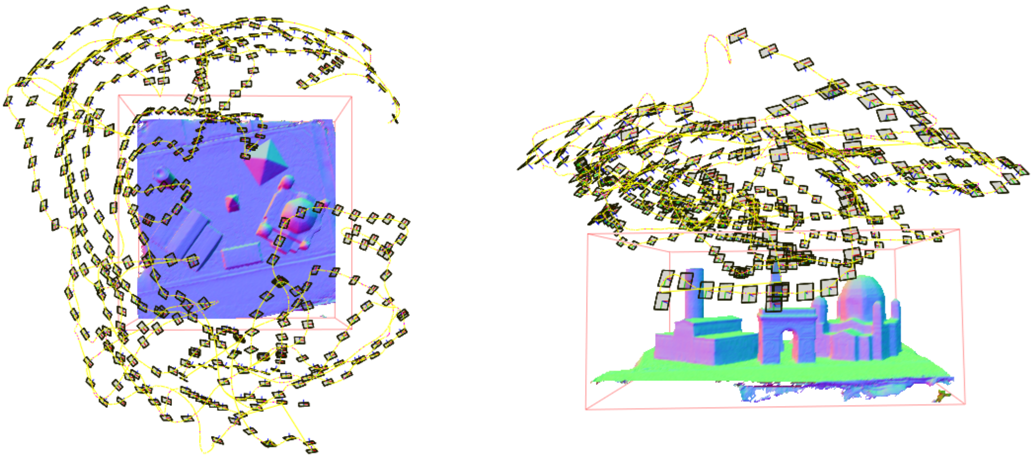


Figure 8.9: An example set of key-frames, illustrating the non-parametric distribution that it provides over the camera pose. Note there is no possibility of finding a camera pose in a location that the camera cannot have physically visited.

lustrated in Figure (8.9). Unlike the feature-based PTAM, in our system we are able to furnish each key-frame with a dense geometry prediction. We can therefore synthesise steps 2, 3 and 4 from Algorithm (1) into a *single* procedure by replacing those steps with a dense $SE3$ tracking stage given the initial pose of the key-frame. In the case where no convergence is obtained over the current key-frame set, we can then proceed to sample *synthetic key-frames* by computing a photometric and geometric prediction of the model into a sampled pose. Finally, we note that while we focus on relocalisation in the passive camera tracking scenario here, the same mechanism can be used for relocalisation in the depth camera tracking pipeline by replacing the tracking component with one of the depth camera tracking approaches from Section (8.3).

Our dense $SE3$ tracking pipeline simplifies the relocalisation mechanism, but is clearly only useful in a limited workspace. Successful relocalisation using the mechanism outlined here is achieved when the basin of convergence governed by the selected key-frames covers the workspace. This provides an insight into a potentially more efficient approach to key-frame selection specifically for relocalisation. Computational and memory costs associated with relocalisation using the approach outlined here grow linearly with the number key-frames, and even when utilising more sophisticated feature based detection mechanisms it is desirable to reduce the number of frames used in the offline feature exaction process if possible (**Dong et al.**, 2009). A sufficient number of key-frames could be defined as the set which maximises the combined basin of convergence given a direct optimisation method over a given workspace with the minimum number of key-frames. Given this, we could exploit the dense geometric and photometric predictive capabilities of the dense model to sample synthetic key-frames to achieve exactly this aim in the future.

9

Dense SLAM Systems

Contents

In this chapter we describe four dense SLAM systems built from the tracking and mapping components detailed in the previous chapters of this thesis. We begin the chapter in Section (9.1) with an outline of the *live dense reconstruction* (LDR) system (**Newcombe and Davison**, 2010) which combines real-time feature-based tracking and mapping with dense surface estimation. Given calibrated imagery we compute overlapping depth maps from dense optical flow correspondences between the frames which are stitched together into a global surface model. While this earlier system enables geometry-aware augmented reality, it lacks the ability to improve the dense reconstruction over time or to exploit the dense surface representation within the core of the visual SLAM system for pose estimation. These are the key developments of the three further systems described here.

In Section (9.2) we describe the *dense tracking and mapping* (DTAM) system (**Newcombe et al.**, 2011c) that replaces the redundant optic flow computation used in the earlier LDR system with a more efficient depth estimation described in this thesis. DTAM also replaces the sparse feature extraction and mapping visual SLAM component with direct whole-image alignment based camera pose estimation. Direct image alignment between the predicted dense surface model and the live image enables tracking throughout rapid agile camera motion in cases where motion blur artefacts lead to failure of sparse feature based tracking. The use of highly parallel general purpose GPU (GPGPU) techniques is at the core of all of our design decisions, allowing live dense surface reconstruction and dense tracking at frame-rate.

The advent of commodity depth cameras has removed the need to solve the hard depth inference problem required in many computer vision applications. In Section (9.3) we describe the *KinectFusion* pipeline (**Newcombe et al.**, 2011b), a fully dense SLAM system that jointly estimates the surface geometry and camera pose in real-time using all available measurements from a commodity depth camera. We replace the stitched depth map scene representation used in DTAM and LDR with a volumetric signed distance function enabling continuous updating of the surface topology. We exploit real-time dense surface predictions from the model to achieve robust camera tracking using direct whole depth-image alignment. We also provide experimental results that demonstrate drift-less tracking and resilience of the system to when used with reduced computing resources and measurement quality. The section closes with an overview of the extensions that have been reported in the literature, along with our own extensions to enable larger scene reconstructions using sub-mapping techniques.

We finish with Section (9.4) where we return to a single passive camera scenario. The final system builds on the dense SLAM concept developed in KinectFusion to enable volumetric SDF surface reconstruction to operate with the real-time depth maps computed using the multi-view stereo techniques developed in this thesis. We interleave surface reconstruction with the real-time camera tracking used in DTAM, directly tracking from the continuously updated geometric and photometric predictions available from the dense scene model. We demonstrate the fundamental capability of the single camera visual SLAM system, in contrast to active or fixed stereo systems: the ability to perform reconstruction across a range of scales with a single sensor.

In tables 9.1 to 9.4 we specify the main components that are used each of the systems described in this chapter. Specifically, we tabulate the use of four major algorithmic components use, that in combination, specify the main differences between the systems described here. In table 9.1 we state the type of surface representation used. Table 9.2 maps which dense depth map estimation techniques are used. Table 9.3 maps out the camera pose estimation technique; and finally 9.4 states the use of surface reconstruction techniques used in each system.

	Surface Representation, Chapter 6			
Dense SLAM Algorithm	*Sparse Point Cloud*	*Global Mesh*	*Dense Key frames*	*Global Implicit Surface*
(9.1) Live Dense Reconstruction	✓	✓	✓*	✓
(9.2) Dense Tracking and Mapping	✓	–	✓*	–
(9.3) KinectFusion	–	–	–	✓
(9.4) Passive Surface Fusion	–	✓	✓	✓*

Table 9.1: Overview of Surface Representations Used.

	Depth Map Estimation, Chapters 4 and 5				
Dense SLAM Algorithm	*Variational Optical Flow correspondences*	*Variational Stereo (full data term search)*	*Variational Stereo (depth map denoising)*	*Variational Stereo (Linearised data term)*	*Structured Light*
(9.1) Live Dense Reconstruction	✓	–	–	–	–
(9.2) Dense Tracking and Mapping	–	✓	–	–	–
(9.3) KinectFusion	–	–	–	–	✓
(9.4) Passive Surface Fusion	–	–	✓	✓	–

Table 9.2: Overview of Dense Depthmap Estimation Algorithms Used.

	Camera Pose Estimation, Chapter 8			
Dense SLAM Algorithm	*Sparse Feature tracking and keyframe BA*	*Direct frame to dense Keyframe (RGB)*	*Direct frame to model (RGB)*	*Direct frame to model Alignment (Depth)*
(9.1) Live Dense Reconstruction	✓	–	–	–
(9.2) Dense Tracking and Mapping	✓	✓*	–	–
(9.3) KinectFusion	–	–	–	✓
(9.4) Passive Surface Fusion	–	–	✓	–

Table 9.3: Overview of Camera Trajectory Estimation Algorithms Used.

	Surface Reconstruction, Chapters 6 and 7	
Dense SLAM Algorithm	*Truncated signed distance function fusion*	*Piecewise Depth map*
(9.1) Live Dense Reconstruction	–	✓
(9.2) Dense Tracking and Mapping	–	✓
(9.3) KinectFusion	✓	–
(9.4) Passive Surface Fusion	✓	–

Table 9.4: Overview of Surface Reconstruction Algorithms Used.

9.1 Live Dense Reconstruction

In this system we combine real-time single passive camera tracking and sparse feature mapping from the *parallel tracking and mapping* system developed by **Klein and Murray** (2007) with a concurrently operating dense surface estimation pipeline to enable *live dense reconstruction* of desktop scale workspaces.

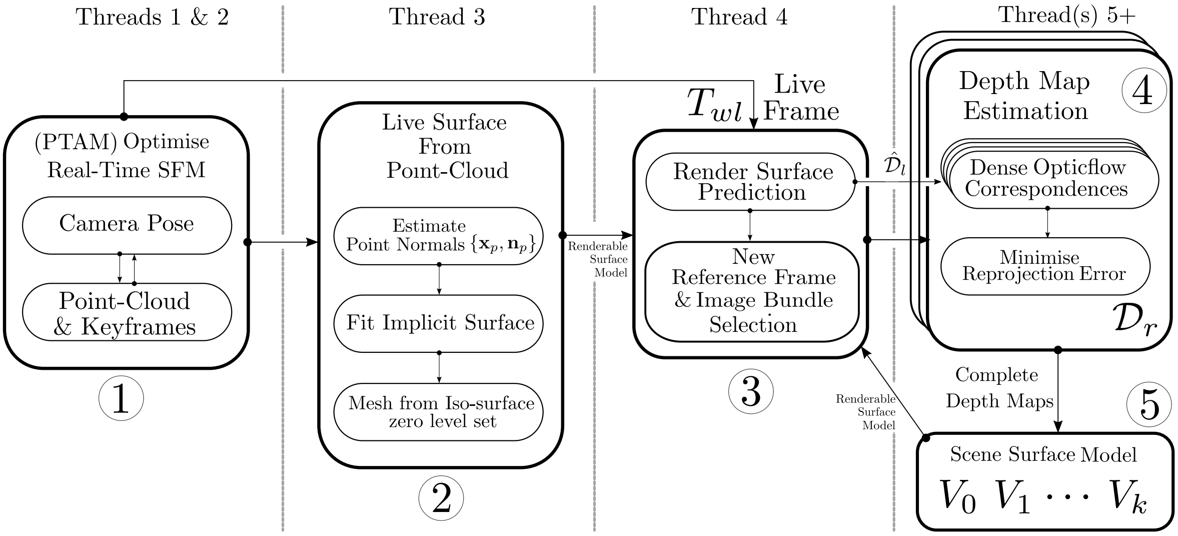


Figure 9.1: LDR system outline. Live dense reconstruction pipeline overview. (1) Online structure from motion provides real-time camera pose estimates and a point map. (2) An implicit surface is fitted to the point map, and the zero level set is polygonised into a coarse mesh. (3) Local bundles of cameras with partial visible surface overlap are selected around reference frame. (4) The base model surface is sampled at every pixel in a given reference frame, and deformed into a photo-consistent dense local model using dense correspondence measurements. (5) All local reconstructions are integrated into the global surface model and redundant vertices are trimmed.

9.1.1 Method

An overview of the LDR pipeline is illustrated in Figure (9.1). The LDR pipeline consists of five components which run in a loosely coupled concurrent form. As outlined in Figure (9.1)(1), a real-time camera pose is estimated within a sparse feature-based structure from motion pipeline, simultaneously providing a point-based map representation of the observed scene. As new points are added to the point map, a continuously updated implicit surface base model is computed and polygonised to provide a dense but approximate estimate of the scene's surfaces, providing a coarse geometry proxy in the form of a mesh, Figure (9.1)(2). Given two camera poses, this coarse but dense surface proxy enables a prediction of the image correspondences between those frames that we use in the high quality surface reconstruction pipeline.

The coarse surface reconstruction is further used to select bundles of frames which have partially overlapping observations of the scene shown in Figure (9.1)(3), each comprising a single reference frame with known pose in the world frame T_{wr} and image $\mathcal{I}_r$ together with several neighbouring frames. Each camera bundle is fed to the dense reconstruction process, Figure (9.1)(4), which produces a dense depth map in the reference view $\mathcal{D}_r$. A depth map is computed by minimising the re-projection error of dense correspondences computed using variational optical flow between the reference and neighbouring views, we initialise the variational optimisation with the predicted optical flow obtained from the coarse surface proxy. Finally, each dense depth map is then triangulated and transformed into the global frame, creating a set of overlapping meshes that form the global surface model, Figure (9.1)(5). In the following subsections we briefly describe each of the components and provide results from LDR in Section (9.1.2).

Real-time Structure from Motion

Drift-free camera tracking is provided by PTAM (**Klein and Murray**, 2007), relying on measurements of hundreds of features per frame, and interleaved with repeated bundle adjustment optimising a set of selected key-frame poses and a point-cloud scene map. Tracking and bundle adjustment operate in parallel using two CPU threads. High quality points are determined in PTAM using inlier/outlier feature re-matching counts. We collect the highest quality $3D$ points from PTAM's map obtained from features matched at the original image resolution to pass on to the dense reconstruction pipeline.

Base Surface Construction

Given the $3D$ point cloud from PTAM we further estimate the surface normal at the point using knowledge of feature co-visibility obtained from PTAM as determined by key-frames observing each point. We make a rough initial surface normal estimate for each point by averaging the optic axis directions of the key-frames in which it is visible. We now aim to estimate an initial continuous scene surface which will form the basis for dense surface reconstruction.

As discussed in Chapter (6), surface reconstruction from oriented point samples has received considerable attention in both the computer vision and graphics communities. Implicit surfaces provide topologically agnostic shape representation, represented in an embedding function $S : \mathbb{R}^3 \mapsto \mathbb{R}$ such that the reconstructed surface is represented as the (zero) level set of the function $S(\mathbf{x}) = 0$. A reconstructed mesh is extracted by polygonising the function's zero level set using a marching cubes technique (**Lorensen and Cline**, 1987) or a continuation style polygonisation method (**Bloomenthal**, 1994).

In our application, speed is crucial in obtaining an up to date base model. Globally optimal non-parametric surface fitting techniques originally suffered from high computational cost of solving large, dense systems of equations (**Turk and OBrien**, 1999; **Kazhdan et al.**, 2006). In more recent years large reductions in the computational cost of reconstruction have traded global optimality for hierarchical, coarse-to-fine solutions. In particular, radial basis functions with finite support have enabled the dense system of constraints to be made sparse. We use a state of the art multi-scale compactly supported radial basis function (MSCSRBF) technique for 3D scattered data interpolation by **Ohtake et al.** (2003) discussed in Section (6.1.3). This method combines some of the best aspects of global and local function approximation and is well suited to the sparse, coarsely oriented point cloud obtained from the PTAM point cloud, in particular retaining the ability to interpolate over large, low density regions. We extract the coarse surface mesh using the implicit surface polygonisation method from **Bloomenthal** (1994). In practice we are able to run base surface reconstruction every time a new key-frame is generated, maintaining an up-to-date base model.

High Quality View-Predictive Optical Flow

Each reference frame has a grey-scale image $\mathcal{I}_r$ and SE_3 pose matrix T_{wr}, together with $n \geq 1$ other nearby calibrated comparison frames $\mathcal{I}_{i\in(1\ldots n)}$. This set of frames constitutes a *camera bundle*. A method for automatically selecting the frames in a camera bundle from the live stream is outlined later on in this section.

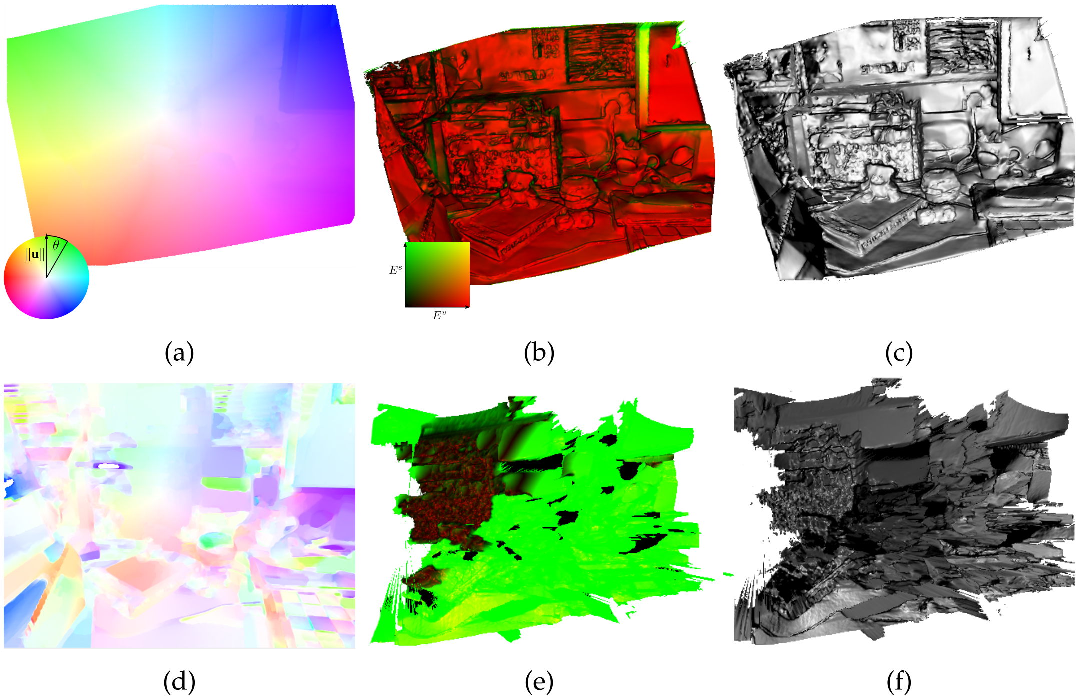
Due to the small baseline nature of neighbouring frames in video we use high accuracy, variational optimisation based optical flow algorithm to obtain dense, sub-pixel quality correspondences between the reference and comparison views. Large improvements to variational optical flow solutions have been gained by utilising an ℓ_1 data fidelity norm and a Total Variation regularisation of the solution $u : \Omega \mapsto \mathbb{R}^2$:

$$\min_u \left\{ \int_\Omega \lambda |\mathcal{I}_0(x) - \mathcal{I}_1(x + u(x))| \mathrm{d}x \ + \int_\Omega |\nabla u| \mathrm{d}x \right\} \ . \tag{9.1}$$

Details of solutions to minimising Equations (9.1) are given in **Pock** (2008) and **Zach et al.** (2007a) with further implementation details of the highly optimised GPGPU solution we use provided in **Wedel et al.** (2009). This two-view optical flow energy is the unconstrained form of the convex stereo energy described in detail in Chapter (5). Here, λ provides a user-controllable trade-off between the data fidelity and regularisation terms.

We initialise the optic flow estimation by warping the reference image via the current coarse surface estimate into the comparison frame, performed efficiently on GPU hardware using projective texturing. Dense correspondence are then obtained by applying optical flow estimation *between each synthesized image for a comparison camera and the real image capture*. Each correspondence field is thereby constructed by summing the estimated optic flow to the predicted flow field.

It is important to note that a valid spatio-temporal derivative must exist across each pixel in the coarsest level of the multi-scale optical flow optimisation for a correspondence to



(a) (b) (c)

(d) (e) (f)

Figure 9.2: Example surface from depth map reconstruction using model predictive optical flow between two images (a) in comparison to the optical flow without base mesh initialisation (d). A rotation around the optical axis induces large displacements of upto 150 pixels resulting in errors in the raw flow field shown in (d). The ego-motion induced rotational component of the flow field is eliminated using view prediction. The polygonised results are shown with associated per-vertex error measures (b,e; red indicates high photo-consistency).

be estimated there, which places a limit on the largest displacement measurable between corresponding image points. Wider baseline dense correspondences are obtained using the *model predictive optical flow* since the distance to corresponding pixels is reduced wherever the base model is approximately correct, including removal of the rotational flow components induced by a rotational difference between the reference and comparison view, increasing the opportunity for a valid spatio-temporal derivatives across the images. Figure (9.2a) illustrates the improved correspondence field produced by model predictive optical flow.

Depth Map Estimation

A reference frame depth map, $\mathcal{D}_r$, is computed by minimising the reprojection error of each reference pixel x_r and with the correspondence in each neighbouring view i obtained from the optical flow u_i:

$$\mathcal{D}_r(u_r) = \underset{d \in \mathbb{R}_+}{\operatorname{argmin}} \sum_{i=1}^{n} \psi(\pi(KT_{ri}K^{-1}\dot{x}_r \cdot d) - x_i) \tag{9.2}$$

$$x_i = x_r + u_i(x_r) \ . \tag{9.3}$$

Assuming co-observation of a surface patch projecting into a reference and neighbouring view, an optimal depth $d_0 + d*$ results in no displacement error between the predicted pixel projection, and the corresponding pixel x_i in view i. Given $n \geq 1$ correspondence field(s), we can therefore obtain the per-pixel depth estimate by iterative non-linear least squares estimation of Equation (9.2).

Stuehmer et al. (2010) pointed out that the use of $n \geq 1$ explicitly obtained optical flow fields followed by minimisation of the re-projection error described here is unnecessary for depth map estimation. Indeed, as described in detail in the multiple-view depth map estimation methods described in Chapter (5), the single degree of freedom per pixel optimisation problem presented by Equation (9.2) can be formulated to minimise the pixel value error directly, using all of the n neighbouring views within a single data-term. This is in contrast to estimation of $2n - 1$ extra variables obtained from full $2D$ correspondence across n views. It is interesting to note however that the optical-flow based depth map estimation provides a potential benefit during surface estimation in the presence of noisy pose estimation. In such a scenario the extra degrees of freedom enable correspondence to be obtained in cases where the epipolar constrained data-terms might result in incorrect matches. Ultimately the full correspondence field provides the required data for a full joint optimisation over all camera poses together with the depth map.

Iterating Model Prediction

We further utilise the model predictive optical flow by performing multiple iterations of the reconstruction algorithm. Prior to triangulation of a depth map we perform depth map denoising using the g-weighted Huber-ℓ_1 model described in Section (4.5). The denoised depth map is then triangulated and the surface normals estimated using the approach outlined in Section (6.1.1), the mesh is then transformed into the global frame replacing the original surface model and leading to an improved view prediction. The optical flow based depth map estimation is repeated on the updated model, ultimately increasing photo-consistency in the reconstructed model. Figure (9.3) illustrates the results of processing a second iteration.

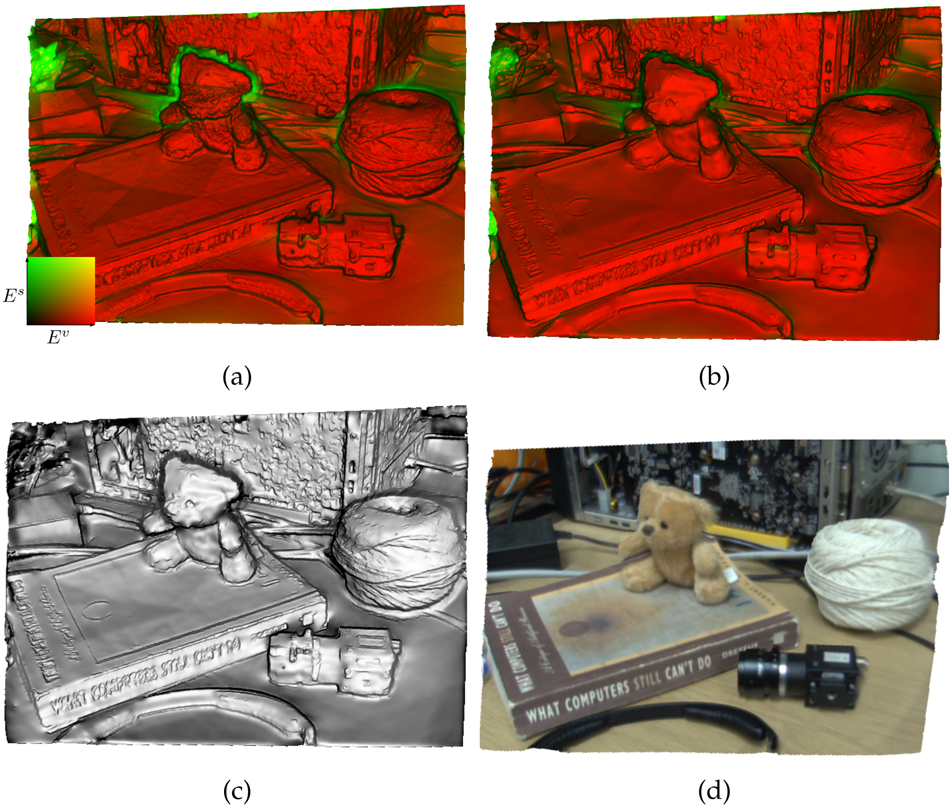


Figure 9.3: A local surface reconstruction using a total of four images of resolution 640 × 480 pixels. Reconstruction after one iterations and associated error measures (a). A second iteration results in high photo consistency (b). The resulting Phong shaded reconstruction (c) and a synthesised view using the reference camera image to texture the surface model (d).

Surface Errors

For each vertex of the triangulated depth map we assign a vector $E(x) = [E^s(x), E^v(x)]$ of measures of reconstruction fidelity. $E^s(x)$ is the per vertex mean reprojection residual resulting from minimisation of Equation (9.2), while $E^v(x) = |\langle K^{-1}\dot{x}, N(x)\rangle|$ is a measure of the visibility of the surface element in the reference view, computed by the inner product of the reference pixel ray with the surface normal computed in the reference frame.

Local Model Integration

A number of algorithms have been developed to enable the fusion of multiple depth maps as discussed in detail in Chapter (6). Here we simply overlay depth maps transformed into the global frame of reference and remove vertices from newly estimated depth maps that are found to be within a given distance threshold of an already mapped vertex. Given a newly triangulated depth map, we render the currently integrated dense reconstruction

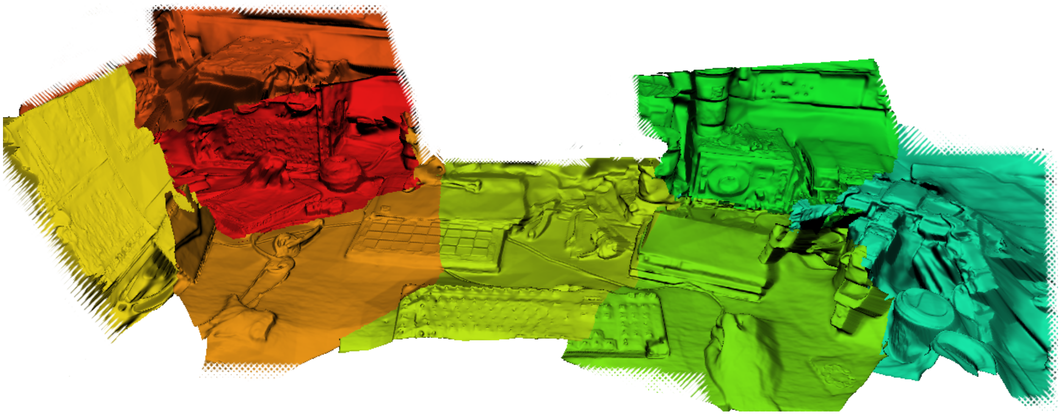


Figure 9.4: Full scene reconstruction obtained during live scene browsing. Eight camera bundles each containing four images including the reference were used for reconstruction. Each colour indicates a region reconstructed from a single camera bundle.

into the reference view and remove the vertices in the new vertex map where the distance to the current vertex is within ϵ_{dist} of the new depth value. We also remove less visible vertices with high solution error in the new mesh where $E^v(x) < 0.9$ and $E^s(x) > 1e^{-3}$. In Figure (9.4) we illustrate how sub-maps that are stitched together to form a reconstruction of a desktop scene.

Camera Bundle Selection

Each local reconstruction requires a camera bundle, consisting of a reference view and neighbouring views, and we aim to select camera bundles to span the whole scene automatically. As the camera browses the scene the integrated model is rendered into the virtual current camera, enabling the ratio of pixels in the current frame that cover the current reconstruction to be computed. We maintain a rolling buffer of the last 60 frames and camera poses from which to select bundle members. When the current reconstruct covers less than one third of the live frame image a new reference frame is initialised into which a depth map will be estimated.

Given the new reference frame pose we obtain a prediction of the surface co-visibility with each subsequent frame. The method for view selection is based on obtaining the largest coverage of different translation only predicted optical flow fields computed by analysing the histogram of a coarse flow field predicted into the neighbouring views using the base surface. The result is a set of n cameras with disparate translations that scale with the distance to the visible surface and that are distributed around the reference view. The automatically selected views increase the sampling of the spatio-temporal gradient between the predicted and real comparison views, reducing effects of the aperture problem in the optical flow computation used in the dense reconstruction.

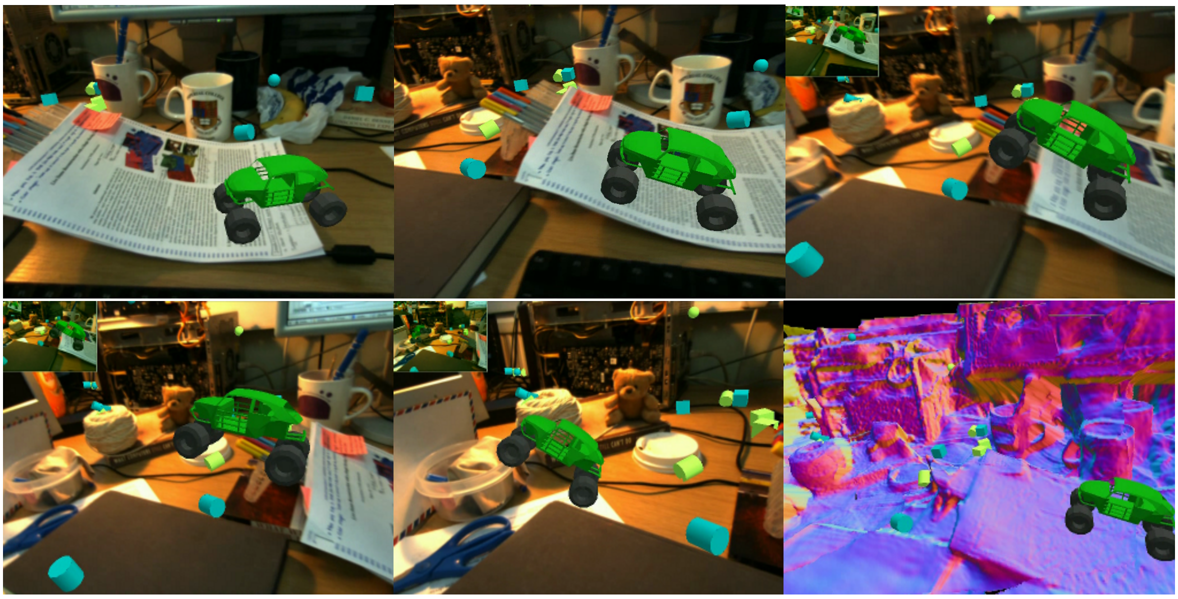


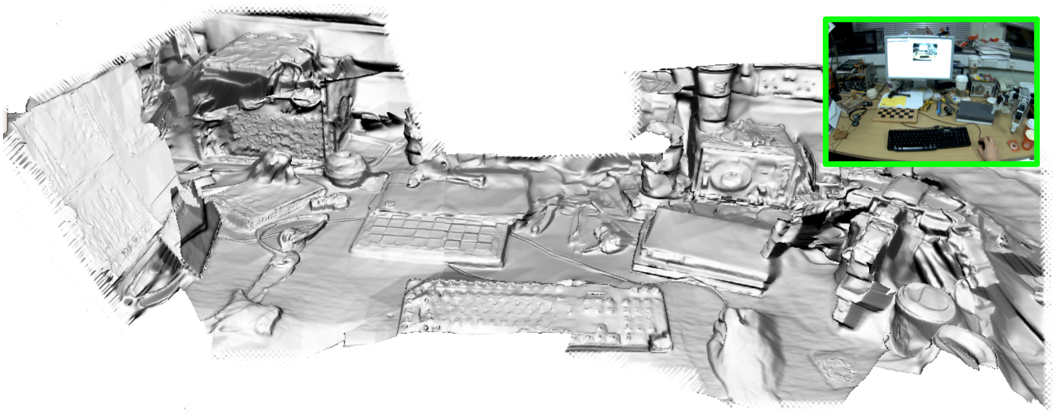
Figure 9.5: Use of the desktop reconstruction for advanced augmented reality, a car game with live physics simulation. Far right: the car is seen sitting on the reconstructed surface. The other views are stills from our video where the car is displayed with the live camera view, jumping off a makeshift ramp, interacting with other objects and exhibiting accurate occlusion clipping.

9.1.2 Results

Our results have been obtained with a hand-held Point Grey Flea2 camera capturing at 30Hz with 640 × 480 resolution and equipped with an 80° horizontal field of view lens. The camera intrinsics were calibrated using PTAM's built-in tool, which includes radial distortion modelling. All computation was performed on a Xeon quad-core PC using one dedicated GPU for variational optic flow computation, and one GPU for live rendering and storage of the reconstructions.

The results are best illustrated by the videos available online [1] which demonstrate extensive examples of the reconstruction pipeline captured live from our system. Here we present a number of figures to illustrate operation. Figure 9.4 demonstrates the sub-mapping based reconstruction, including a number of low texture objects, obtained using four comparison images per bundle from a slowly moving camera, also shown in Figure (9.6) with an insert image of scene being reconstructed. . To give an indication of scale and the reconstruction baseline used, here the camera was approximately $300mm$ to $700mm$ from the scene and the automatically selected comparison frames were all within $50mm$ of the reference frame. The reconstruction quality demonstrates that the model predictive optical flow provides

[1] http://www.doc.ic.ac.uk/~rnewcomb/CVPR2010/



(a)

Figure 9.6: Scene rendering using diffuse Phong shading with inset image over view of the scene reconstructed also shown in sub-map form in Figure (9.4).

accurate sub-pixel correspondence.

Figure (9.5) demonstrates use of the global dense surface model in a physics simulator. Our dense reconstruction permits augmented reality demonstrations far beyond those of **Klein and Murray** (2007) that are restricted to a single plane and do not provide synthetic object occlusion or interaction. The reconstruction was made in under thirty seconds as the camera naturally browsed the scene resulting in faster camera motion in this experiment slightly reduces image and therefore reconstruction quality compared to the more controlled setting used for reconstruction of the scene in Figure (9.6). However, this reconstruction is representative of live operation of the system and is highly usable for geometry aware augmented reality.

9.1.3 Summary

In this system we presented a pipeline which enables automatic live dense reconstruction in the context of live camera tracking, enabling geometry aware augmented reality with a single camera, but the system is lacking in a number of ways. Most importantly, the surface representation with the depth map estimation approach outlined here does not permit continuous updating of a previous reconstructed surface region. This results in incorrectly estimated surface topology wherever the base proxy is grossly inaccurate. Also, as previously noted, the method of solving for each depth map in the reconstruction using optical flow results in a more computationally expensive optimisation problem than is required if a static scene is assumed. Finally, as stated throughout this thesis, the availability of a dense surface model opens up the ability to perform camera tracking using a direct optimisation approach, which the LDR system here does not achieve.

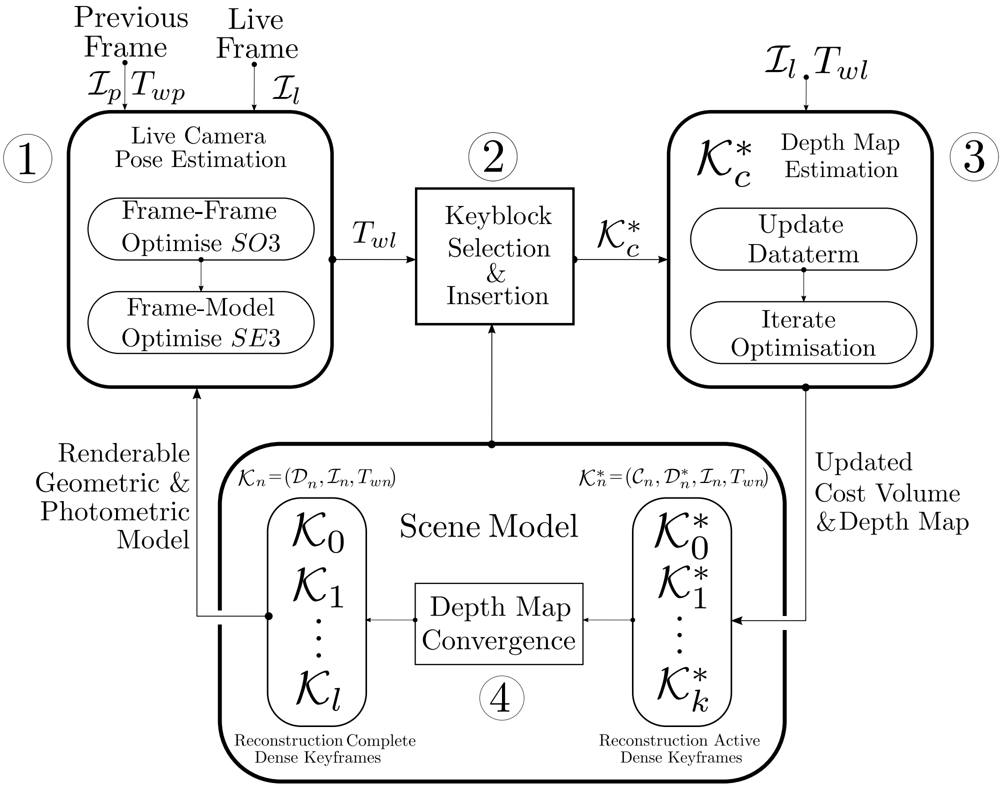

Previous Frame
Live Frame
$\mathcal{I}_p$ T_{wp}
$\mathcal{I}_l$
1
Live Camera Pose Estimation
Frame-Frame Optimise $SO3$
Frame-Model Optimise $SE3$
T_{wl}
2
Keyblock Selection & Insertion
$\mathcal{K}^*_c$
$\mathcal{I}_l$, T_{wl}
3
$\mathcal{K}^*_c$
Depth Map Estimation
Update Dataterm
Iterate Optimisation
Renderable Geometric & Photometric Model
Updated Cost Volume & Depth Map
$\mathcal{K}_n = (\mathcal{D}_n, \mathcal{I}_n, T_{wn})$
$\mathcal{K}^*_n = (\mathcal{C}_n, \mathcal{D}^*_n, \mathcal{I}_n, T_{wn})$
Scene Model
$\mathcal{K}_0$
$\mathcal{K}_1$
$\mathcal{K}_l$
$\mathcal{K}^*_0$
$\mathcal{K}^*_1$
$\mathcal{K}^*_k$
Depth Map Convergence
4
Reconstruction Complete Dense Keyframes
Reconstruction Active Dense Keyframes

Figure 9.7: DTAM system outline.

9.2 DTAM: Dense Tracking and Mapping in Real-time

We build on the LDR from Section (9.1) in two specific ways: we replace the depth map estimation pipeline that used computationally expensive view-predictive optical-flow with the more efficient multi-view depth map estimation from Section (5.3), enabling us to incorporate hundreds of video frames of live video into the computation of a single depth map. Furthermore, we fully exploit the ability to compute a geometric and photometric rendering of the scene into a live frame enabling the robust whole image alignment of the live image with the dense model to estimate the camera pose without requiring explicit feature extraction and matching.

9.2.1 Method

As presented in Figure (9.7), the overall structure of our algorithm is straightforward. Given a dense model of the scene, represented with a partially overlapping set of textured depth maps, we use direct whole image alignment against that model to track camera motion at frame-rate (9.7)(1). Tightly interleaved with this, we update and expand the model by determining scene regions which have not yet been represented in the model (9.7)(2),

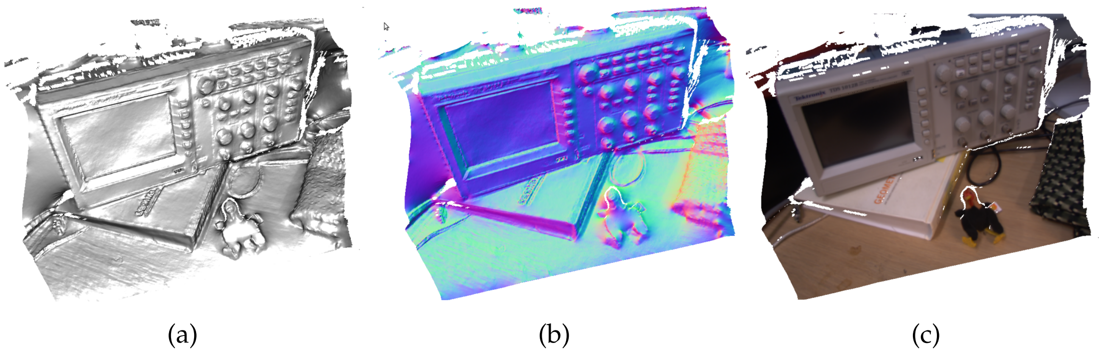


Figure 9.8: The scene is represented as an overlapping set of textured depth maps.

inserting and refining the dense textured depth maps using historically tracked camera frames (9.7)(3). Once bootstrapped, the system is fully self-supporting and no feature-based skeleton or tracking is required. In the following sections we outline the components of the DTAM pipeline.

Scene Representation

The scene model comprises two connected sets of overlapping *dense keyframes* which hold geometric and photometric representations of the scene as a collection of depth maps computed during live camera motion, depicted in Figure (9.7). Elements of the first set are called *reconstruction active* keyframes denoted as $\mathcal{K}^*_{i\in 0..k}$. Each active keyframe consists of a reference image $\mathcal{I}_r$ with known pose $\mathtt{T}_{wr}$ and a data cost volume $\mathcal{C}_r$ that stores the average photometric error. We will use this in the global optimisation based depth map estimation technique described in Chapter (5) to compute a depth map $\mathcal{D}_r$ for the keyframe using tens to hundreds of video frames collected from nearby and overlapping real-time frames. The depth map estimation technique enables a depth to be computed incrementally denoted $\mathcal{D}^*_r$ during estimation, interleaving the update of the associated cost volume with optimisation to find the optimal depth map. Once a depth map has been fully estimated we convert the active keyframe into a *reconstruction complete* dense keyframe denoted $\mathcal{K}_{i\in 0..l}$, containing only the depth map $\mathcal{D}_r$, reference image and camera transform. The memory used in the cost volume storage are then released to be used in a new active keyframe. A sample keyframe is shown in Figure (9.8). We provide more details on keyframe insertion and management at the end of this Section.

Dense Tracking

Given a dense model consisting of one or more keyframes, we can synthesise novel photo-consistent views over wide baselines by projecting the entire model into a virtual camera. Since such a model is maintained live, we benefit from a fully predictive surface represen-

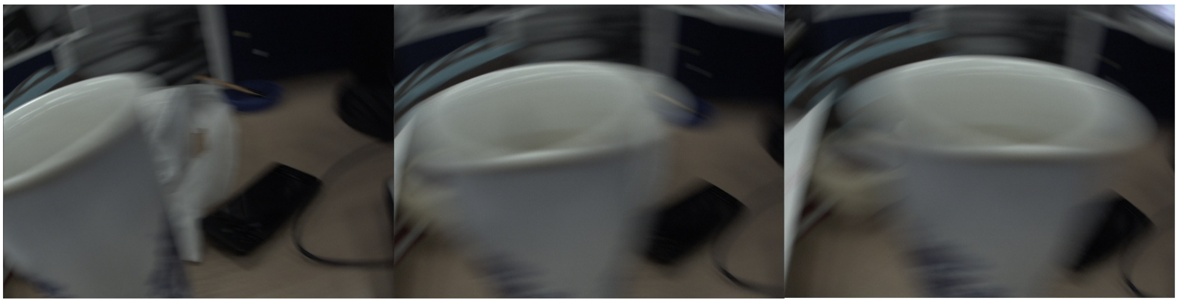


Figure 9.9: Motion blur in a scene with few trackable features. A dense scene model enables whole image alignment to continue tracking despite motion blur.

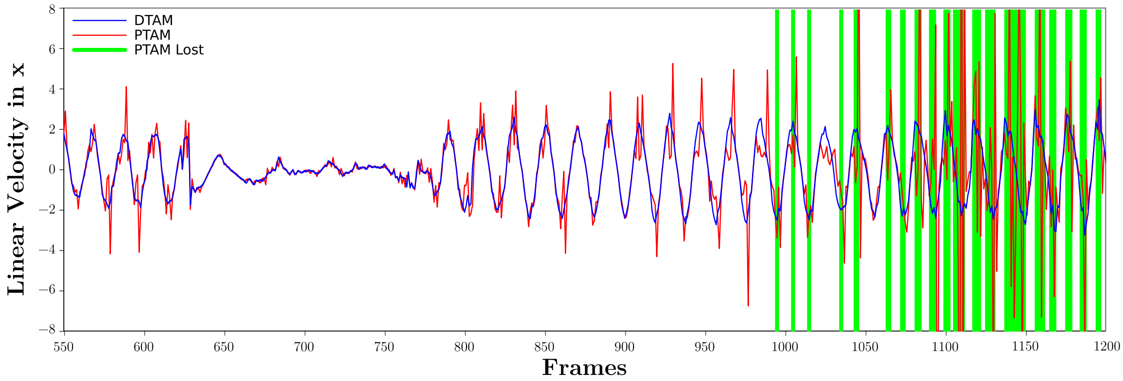
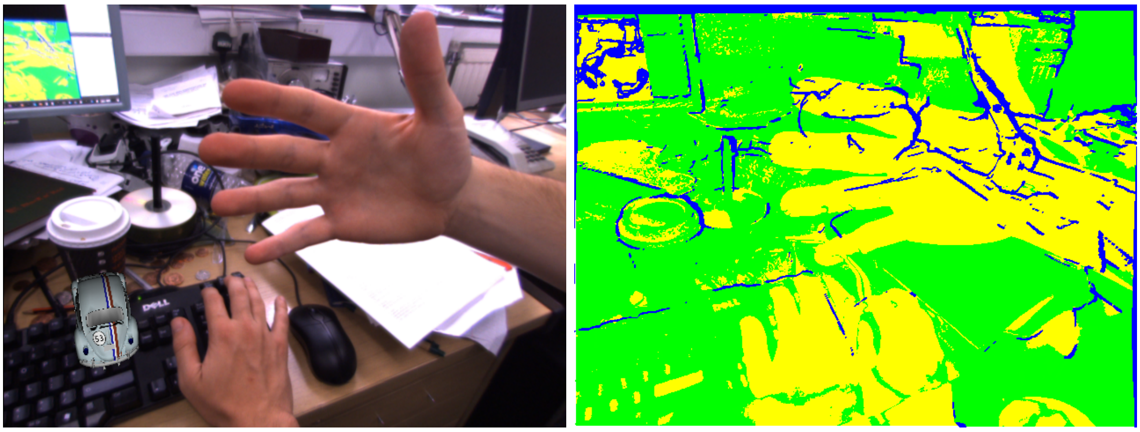


Figure 9.10: Linear velocities for DTAM (blue) and PTAM (red) over a challenging high acceleration back-and-forth trajectory close to a cup. Areas where PTAM lost tracking and resorted to relocalisation are shown in green. In comparison, DTAM's relocaliser was disabled. Notice that DTAM's linear velocity plot reflects smoother motion estimation.

tation, handling occluded regions and back faces naturally. We estimate the pose of a live camera by finding the parameters of motion which generate a synthetic view which best matches the live video image using the two stage pipeline detailed in Section (8.2.6). The pipeline first estimates a constrained inter-frame rotation estimation using the direct alignment method of **Lovegrove and Davison** (2010). From lower levels within an image pyramid computed on the live and previous frames we obtain the rotational odometry, offering resilience to motion blur since consecutive images will have common artefacts including motion blur which are not modelled explicitly. This lower dimensional optimisation is also more stable than 6DoF estimation when the number of pixels considered is low, helping to converge for large pixel motions, even when the true rotation is not strictly rotational. Second, we initialise 6DoF full pose refinement against the model as detailed in Section (8.2.2). In Figures (9.9) and (9.10) we provide an illustration of the improved resilience that dense tracking provides against motion blur in scenes with low texture regions. Tracking throughout camera defocus is demonstrated in Figure (9.15).



(a) (b)

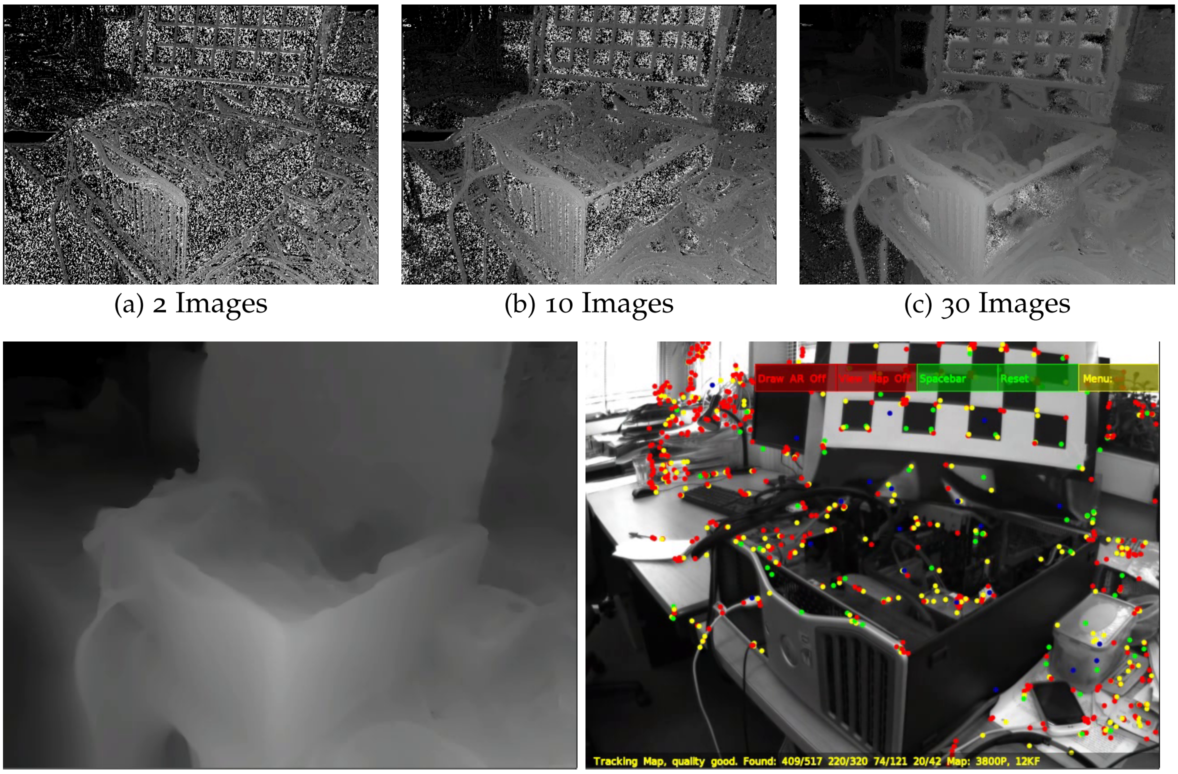
Figure 9.11: Excerpt from video of the live running DTAM system, tracking robustness to unmodelled dynamic scene motion in the live image. An augmented reality car appears fixed rigidly to the world as an unmodelled hand is waved in front of the camera (a). Pixels in **green** are used for tracking whilst **blue** correspond to unmoddled scene regions and **yellow** are rejected (hand / monitor / shadow) using a Tukey penalty term.

In Section (8.2), we detail the forward-composition based whole image alignment method that tracks the live image $\mathcal{I}_l$ by minimising the pixel error induced from a current model. We robustify the optimisation against outliers in both geometric (depth map) and photometric predictions using the Tukey penalty term, modelling the pixel error distribution as a corrupted Gaussian with uniform outlier distribution. We reject pixels with image error magnitude above a user defined threshold, altering the variance of the inlier distribution to ensure that gross outliers from moving objects and variations between the photometric prediction and live frame caused by non-Lambertian surface reflectance and lighting fluctuations are rejected from the error function, shown in Figure (9.11).

Depth Map Estimation

The reconstruction framework is targeted at a live setting, where hundreds of narrow-baseline video frames are the input to each depth map. We gather photometric information sequentially in a cost volume, and incrementally solve for regularised depth maps via a novel non-convex optimisation framework with elements including accelerated exact exhaustive search to avoid coarse-to-fine warping that can result in convergence of poor quality local minima. Full details of the depth map estimation process are provided in Chapter (5), Section (5.3).

One of the main modes of use of the DTAM system is in geometry aware augmented reality, in which a live dense reconstruction of a workspace scene is first reconstructed to a user defined level of completeness. The scene model is then used in a dense tracking


Draw AR Off
Map Off
Spacebar
Reset
Menu:
Tracking Map, quality good. Found: 409/517 220/320 74/121 20/42 Map: 3800P, 12KF

(a) 2 Images (b) 10 Images (c) 30 Images

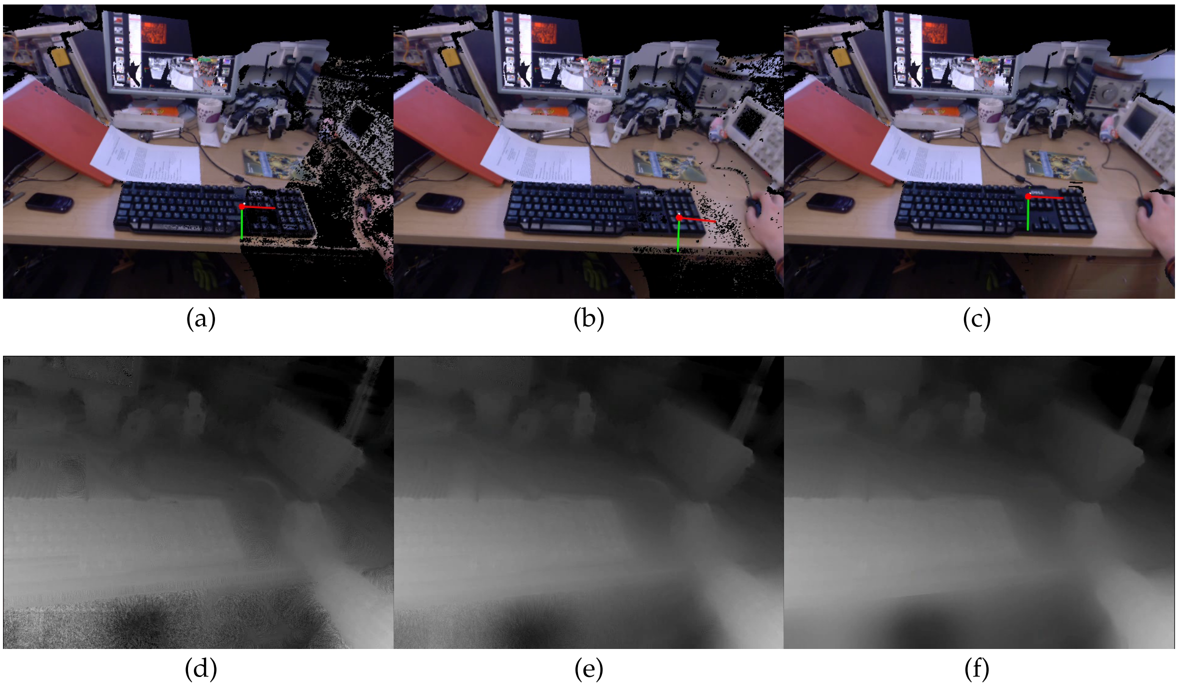
(d) Regularised dense depth map. (e) Feature-based PTAM map.

Figure 9.12: The data term is incrementally constructed using consecutive frames from the live image stream. In (a,b,c) we show the depth map data term minimum for an increasing number of images, while in (d) and (e) we compare the partial scene model estimated for the reference frame with the sparser point map from PTAM, illustrating the increase in data available in a dense-key.

only mode to provide occlusion masking in mixed reality graphics pipelines and sufficient surface representation of the scene for physical simulation of basic interactions with the static surfaces. In this setting we harness the ability of a system user to gather a high quality data term by manoeuvring the camera whilst using the feedback provided from the interleaved depth map optimisation process showing the convergence process of the depth map estimation, as illustrated in Figure (9.12) for a single depth map. An example of the interleaved cost volume update and optimisation process is shown in Figure (9.13).

Dense Keyframe Selection and Insertion

Depth map estimation is performed on the currently selected *active* key-frame $\mathcal{K}_c^*$ which is either selected from a current active set or inserted into the set by reasoning about the visibility of the current scene reconstruction given the newly estimated live camera pose. Given the current scene model comprising semi and fully converged triangulated depth



(a) (b) (c)

(d) (e) (f)

Figure 9.13: Interleaved depth map cost volume acquisition with optimisation. Given a partial reconstruction of the scene a dense keyframe is added. Depth map estimation for the keyframe then proceeds by interleaving the addition each subsequent video frame into the depth map cost volume with a fixed set of optimisation iterations. In (a,b,c) we show the increasing completeness of the textured scene model as more data is added to the cost volume and optimisation proceeds. In (d,e,f) we show the corresponding regularised dense key-frame depth map $\mathcal{D}_r^*$ with the final solution shown in (f). We note that the textured scene model can make use of the incomplete depth map after only a small number of data-terms have been added into the cost volume by thresholding away regions which are yet to converge.

maps, together with the live camera pose, we compute a live prediction of the currently modelled geometry. A new key-frame is added when the number of pixels in the previous predicted image without visible surface information falls below some threshold. This key-frame insertion mechanism is arguably better founded than heuristics used in feature based systems which can not reason about the visibility of the current model in the live frame and instead rely on inter key-frame distance based insertion metrics. We also utilise the predicted minimum and maximum scene depth from the current model to set the depth range of newly inserted key-frame cost-volumes, enabling efficient quantisation of the inverse depth parametrised data-term.

If there is already sufficient coverage from the scene model and there are remaining active key-frames, the system must choose which of the key-frames should be selected to continue depth map estimation. For each active key-frame we compute an average of co-visible points between the key-frame and live frame, and transform the average into the live frame. The active key-frame is chosen based on the average point that projects closest to the live frame image center. Since each data-term cost volume requires a significant amount of memory we store only a limited number of the most recently used active key-frames in GPU memory. We update the GPU memory from host-CPU memory resident active key-frames, swapping between the memories as required to enable many more active frames to be used in the system.

The annealing procedure used in the alternating depth map optimisation of Section (5.3), requires a fixed number of iterations to achieve a suitable level of depth map convergence. We deterministically interleave the data-term update of the active cost volume with a fixed set of 30 iterations of the primal-dual optimisation procedure, converting the key-frame into the reconstruction's complete form upon convergence with a user defined number of frames used in the data-term. We set the number of input frames between 30 and 200 images per depth map but note that a second DTAM mode enables prolonged data-term collection, by enabling the user to specify where a new key-frame is inserted and when to stop optimisation. We found that this semi-automatic reconstruction process allows a user to achieve a higher quality scene reconstruction by using a slower camera motion to reduce motion blur artefacts in the images while also increasing camera pose accuracy.

Model Initialisation

The system is initialised using calibrated imagery obtained from PTAM (**Klein and Murray**, 2007), continuing to track the live camera pose and update an initial keyframe which is added at a user-defined time. DTAM is then switched to the fully dense tracking and mapping pipeline. Initialisation need not be performed at frame-rate, but only within a reasonable time frame for the one-time key-frame initialisation in a live setting. Therefore a joint-optimisation that estimates both an initial set of camera poses and a single dense depth map could replace the current sparse-feature based bootstrapping (**Szeliski and Kang**, 1993), potentially improving the ability to initialise the system in feature poor environments.

9.2.2 Results

We have evaluated DTAM in the same desktop setting where PTAM has been successful. In all experiments, we have used a Point Grey Flea2 camera, operating at 30Hz with 640×480 resolution and 24bit RGB colour. The camera has pre-calibrated intrinsics. We run on a commodity system consisting of an NVIDIA GTX 480 GPU hosted by an i7 quad-core CPU. We provide an illustration of the live running system demonstrating the ability of DTAM,

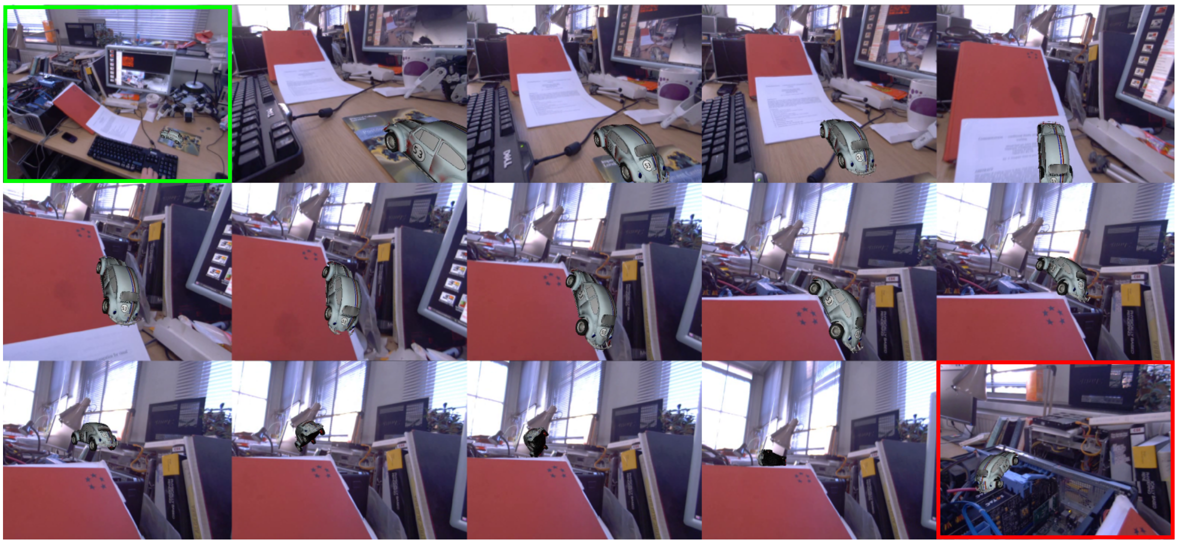


Figure 9.14: Example geometry aware augmented reality with a car. Completing the scene reconstruction shown in Figure (9.13), the dense surface model is used in an augmented reality game enabling the desktop to be driven over by a virtual car with simulated physical interaction over the static scene. We show an overview of the scene that is reconstructed in the top left corner of the figure (green box), and illustrate the large scale over which dense tracking with the model is performed: following the virtual car with the live camera as it drives up a ramp constructed from a book, and coming to stop atop of the graphics card currently running the DTAM computations (bottom right, red box).

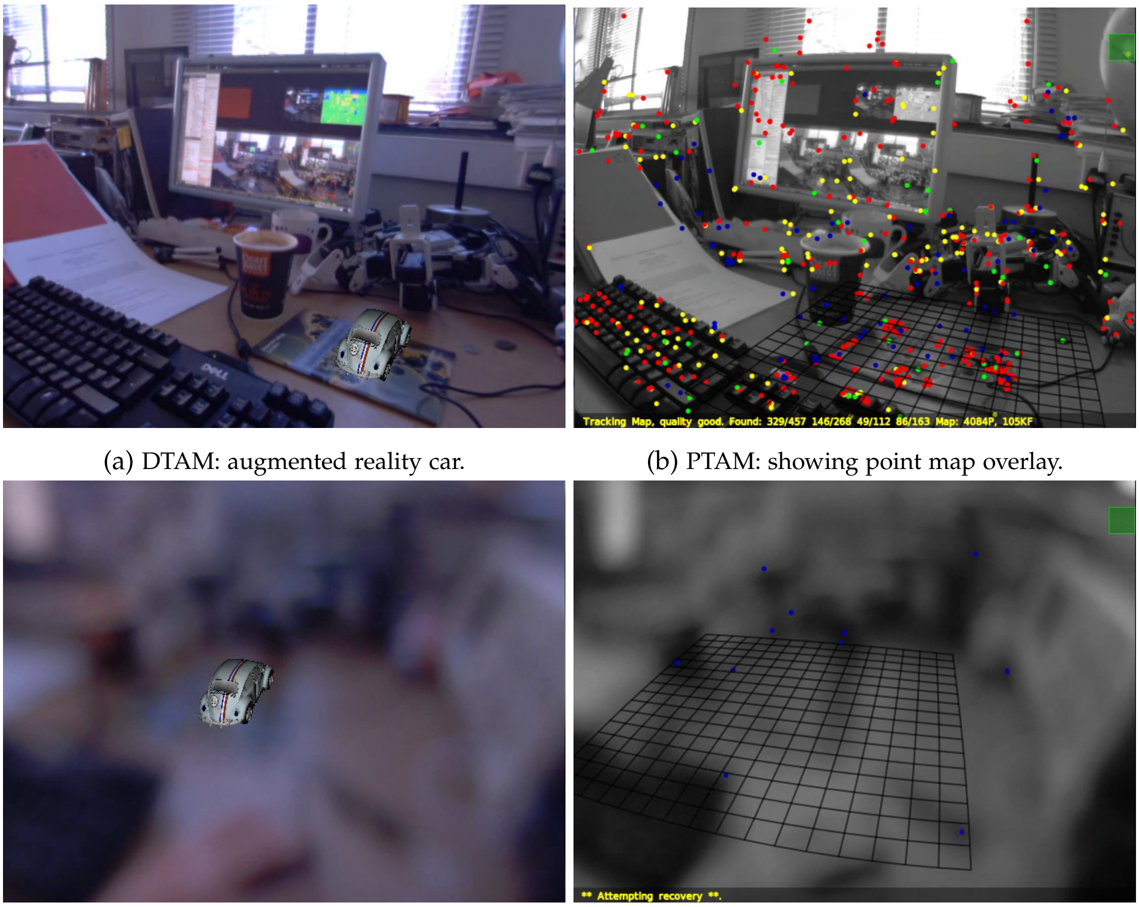
with particular interest for augmented reality use where it is desirable to maintain quality tracking throughout high speed agile motion (Figure 9.9), altering camera focus (Figure 9.15) and dynamic scene interaction (Figure 9.11).

Geometry Aware Augmented Reality

We present a qualitative comparison of the live running system including extensive tracking comparisons with PTAM and augmented reality demonstrations in an accompanying video `http://youtu.be/Df9WhgibCQA`, with extracted stills of an physical car simulation driving over a reconstructed desktop scene shown in Figure (9.14).

9.2.3 Summary

While the ability to render a dense surface prediction provides numerous benefits over a sparse feature-based scene representation, in particular for robust camera tracking, the system described in this section lacks the ability to continuously update regions of the scene which have previously been represented with a single depth map. While overlapping depth maps can represent any surface topology in principle, the fixed nature of each depth map results in errors in the reconstruction which cannot be updated to incorporate newer


Tracking Map, quality good. Found: 329/457 146/268 49/112 86/163 Map: 4084P, 105KF
** Attempting recovery **.

(a) DTAM: augmented reality car.

(b) PTAM: showing point map overlay.

(c) DTAM: dense tracking succeeds throughout camera defocus.

(d) PTAM: detected features during defocus prior to complete tracking failure.

Figure 9.15: Tracking benefits from the predictive capabilities of a dense model with regard to occlusion handling and multi-scale operation, making it much more robust and at least as accurate as any feature-based method; in particular, performance degrades remarkably gracefully in reaction to motion blur or camera defocus.

measurements, for example when the camera is brought closer to the scene, or if the scene structure alters over time. Furthermore, each depth map is anchored to the reference frame pose used to create it which is fixed after initial estimation, hence although tracking is driftless relative to a fixed model, the system has no way to prevent the build up of camera drift that inevitably accumulates when jointly mapping and tracking extended scenes.

9.3 KinectFusion: Dense SLAM with a Depth Camera

The advent of commodity depth sensors has ushered in a new era in the application of computer vision in the real-world. Affordable depth cameras based on structured light stereo estimation such as the Primesense design used in the Microsoft Kinect and Asus Xtion cameras removes one of the computationally demanding components of low level vision required in many higher level vision applications, and in particular the single camera LDR and visual SLAM systems previously discussed in Sections (9.1) and (9.2). In this section we exploit this new capability to focus on an investigation in scene representation: lifting the limitations imposed by the explicit mesh based scene representations used in those single camera systems that prevented a truly incremental and continuous reconstruction of the scene.

The system described in this section, KinectFusion, is a dense SLAM pipeline which permits real-time, dense volumetric reconstruction of complex room-sized scenes using a hand-held commodity Kinect sensor. The core components of the system are the volumetric truncated signed distance function surface reconstruction and the dense whole-depth image pose estimation frameworks described in this thesis. While the depth measurements used in this system are provided by an active depth camera, the pipeline that was developed provided insights into dense surface scene representation, reconstruction and tracking in general, and led to both the LDR system described in Chapter (7) that performs incremental surface reconstruction from video, and finally to the single camera dense visual SLAM system introduced in Section (9.4).

With KinectFusion, users can simply pick up and move the depth camera to generate a continuously updating, smooth, fully fused 3D surface reconstruction. Using only depth data, the system continuously tracks the 6DoF pose of the sensor *using all of the live data available* from the Kinect sensor rather than an abstracted feature subset, and integrates depth measurements into a global dense volumetric model. A key novelty is that tracking, performed at 30Hz frame-rate, is always relative to the *fully up-to-date fused dense model*, and we demonstrate the advantages this offers.

9.3.1 Method

In Figure (9.16), we outline the KinectFusion pipeline that combines the *volumetric truncated signed distance function* (TSDF) integration approach to dense surface reconstruction described in Chapter (6), together with the dense ICP depth map *frame to model* tracking described in detail in Chapter (8). Unlike the previously described dense visual SLAM systems, KinectFusion performs real-time continuous surface reconstruction, enabling the

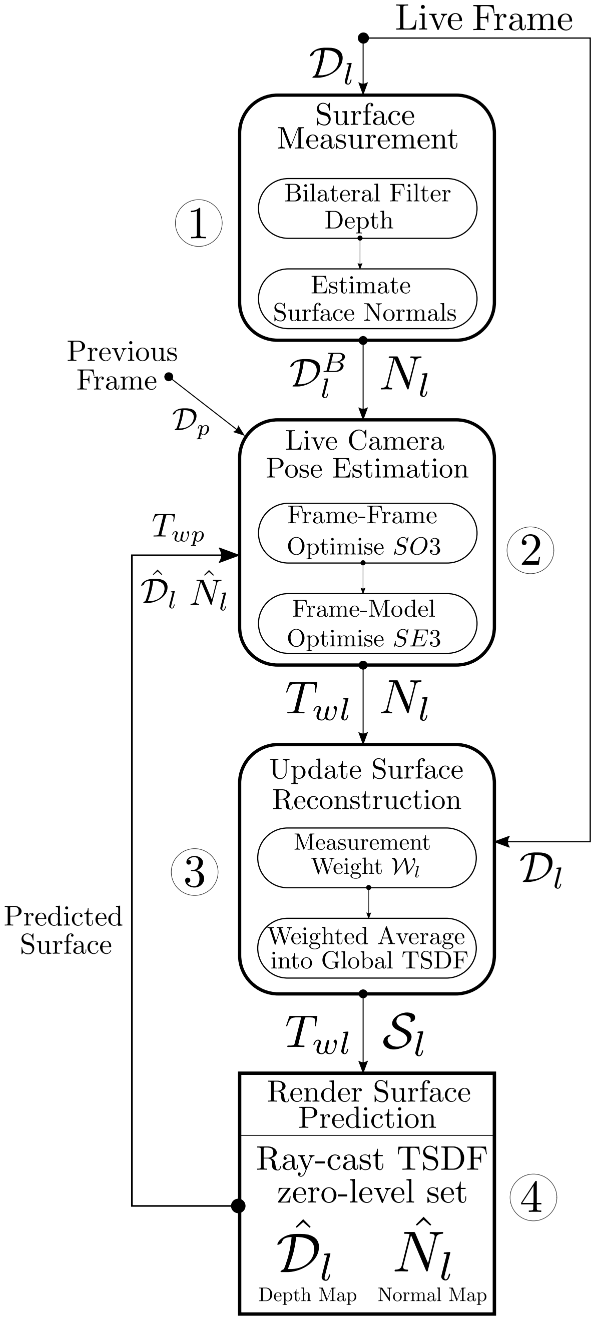
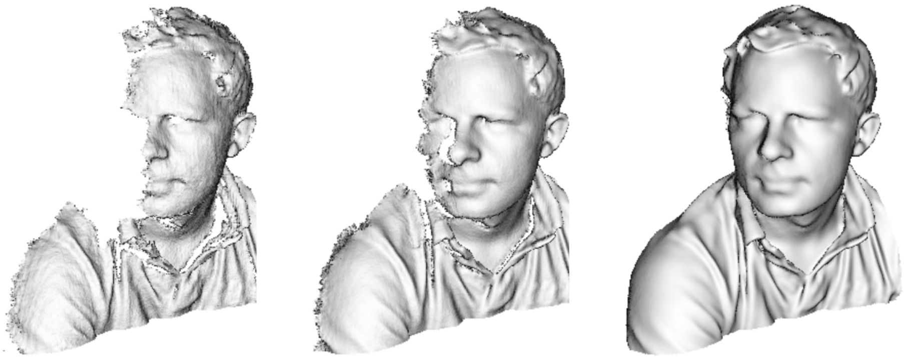


Figure 9.16: KinectFusion Pipeline Overview.



(a) 2 Frames (b) 30 Frames (c) 100 Frames

Figure 9.17: KinectFusion reconstructing a person. In (a,b,c) we show the partial reconstructions after integrating 2, 30 and 100 frames into the model, also illustrating the state of the available model with which frame-to-model tracking is performed.

correct topology of the surfaces to be refined over time from continuous observations of the surface. The four components of the KinectKusion have been designed to take advantage of GPGPU hardware, enabling all depth measurements to be used both the dense reconstruction and tracking at frame-rate. In Figure (9.16)(1) we pre-process the raw live depth map which is then tracked against the most up to date surface prediction (9.16)(2), using a point-plane metric based whole image ICP with projective data association. Given the live frame pose estimate we fuse the surface measurement into a global truncated signed distance function representation of the scene, Figure (9.16)(3). We then render a prediction of the current surface estimate that will be used in tracking the next sensor frame, Figure (9.16)(4). In Figure (9.17) we illustrate the incremental reconstruction process which we now describe in more detail.

Surface Measurement

A live surface measurement comprises a raw depth map $\mathcal{D}_l$ obtained from the commodity structured light sensor, providing calibrated depth measurements $\mathcal{D}_l(u) \in \mathbb{R}_+$ at each image pixel $u \in \Omega$. We apply a bilateral filter (**Tomasi and Manduchi**, 1998) to the raw depth map to obtain a discontinuity preserved depth map with reduced noise $\mathcal{D}_l^B$, which is used in estimation of the the surface normals using the method described in Section (6.1.1), yielding a live measurement normal map estimate N_l.

Surface Reconstruction Update

Each consecutive depth frame, with an associated live camera pose estimate, is fused incrementally into one single 3D reconstruction using the volumetric truncated signed distance function (TSDF) described in Chapter (6). In a true signed distance function, the value corresponds to the signed distance to the closest zero crossing (the surface interface), taking on positive and increasing values moving from the visible surface into free space, and negative and decreasing values on the non-visible side. The result of averaging the SDF's of multiple 3D point clouds (or surface measurements) that are aligned into a global frame is a global surface fusion. We use the trivially parallelisable projective *truncated* signed distance function (TSDF) fusion described in detail in Section (6.2).

Surface Prediction

With the most up-to-date reconstruction available comes the ability to compute a dense surface prediction by rendering the surface encoded in the zero level set of the TSDF into a virtual camera. As detailed in Section (6.3), rendering of a predicted surface depth and normal map can be achieved efficiently and directly on GPGPU hardware either for a specific view through iso-surface raycasting or using full iso-surfce extraction via marching cubes, followed by rendering of the view with a rasterising graphics pipeline.

Live Sensor Pose Update

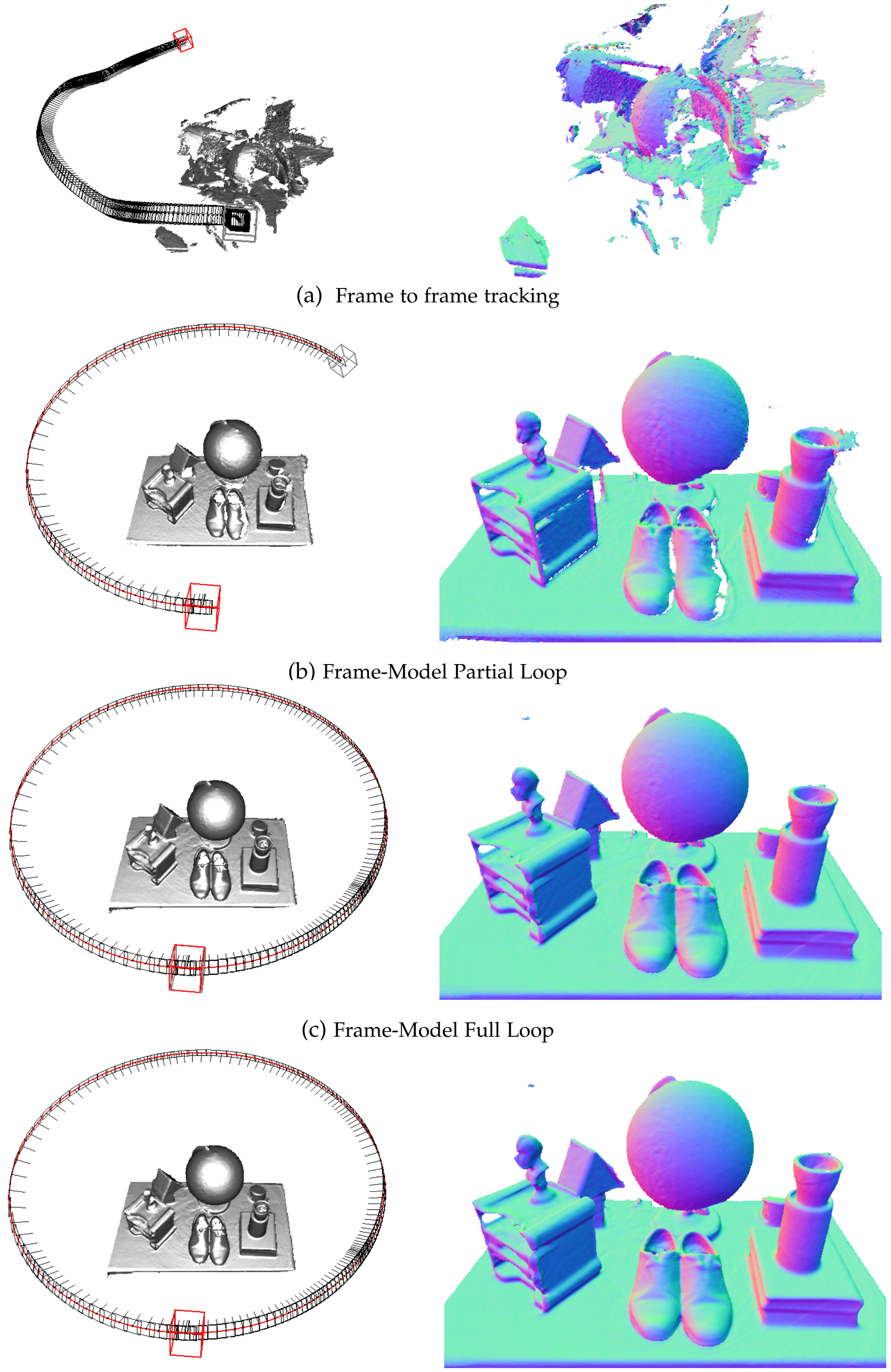
We utilise the full predictive capabilities of the up to date surface model for real-time sensor pose estimation using a direct whole depth image alignment. Rendering the updated model into a predicted live sensor frame, simply taken here as the previous frame pose estimate (constant position motion model), we therefore predict the depth and normal map $\hat{\mathcal{D}}_l$ and $\hat{N}_l$ and obtain the current live pose estimate using the bilaterally filtered depth and normal map surface measurement with the dense point-plane based ICP described in Section (8.3). In contrast to previous depth map based SLAM (scan-matching) systems, we take advantage of full GPGPU acceleration to enable use of *all* of the data in a depth image without needing to sub-sample or sparsify the data. The resulting fully dense system scales gracefully with higher and lower resolution input data, as demonstrated in the following experimental section.

9.3.2 Experiments

We have conducted a number of experiments to investigate the performance of our system. These and other aspects, such as the system's ability to keep track during very rapid motion, are illustrated extensively in an accompanying video `http://youtu.be/quGhaggn3cQ`.

Metrically Consistent Reconstruction

Our tracking and mapping system provides a constant time algorithm for a given area of reconstruction, and we are interested in investigating its ability to form metrically consistent models from trajectories containing local loop closures without requiring explicit joint



(a) Frame to frame tracking

(b) Frame-Model Partial Loop

(c) Frame-Model Full Loop

(d) Full Loop Duplicated M times

Figure 9.18: Circular motion experiment to highlight the SLAM characteristics of our system as the sensor orbits a table. For each tracking mode we show the estimated sensor trajectory (every 4th of N frames is shown) (left row), together with a closer view highlighting reconstruction quality with normal map rendering. Details of the experiment are given in the main text.

optimisation over the camera pose and scene structure state. We are also interested in the ability of the system to scale gracefully with different processing and memory resources.

To investigate these properties we conducted the following experiment. The Kinect sensor was placed in a fixed location observing a tabletop scene mounted on a turntable. The turntable was then spun through a full rotation as depth data was captured over ≈ 19 seconds, resulting in $N = 560$ frames. For the purposes of our system, if the reconstruction volume is set to span solely the region of the rotating scene. The resulting depth image sequence obtained is obviously equivalent to the Kinect having been moved on a precise circular track around a static table, and this allows us to easily evaluate the quality of tracking. All parameters of the system are kept constant, using a reconstruction resolution of 256^3 voxels unless stated otherwise.

The N frames of depth data captured were then processed in each of the following ways:

1. Frames $1 \ldots N$ were fused together within the TSDF using sensor pose estimates obtained with our frame-to-frame *only* ICP implementation.

2. Frames $1 \ldots L$, $L < N$ were fed through our standard tracking and mapping pipeline, forming an incomplete loop closure. Here, sensor pose estimates are obtained by the full frame-model ICP method.

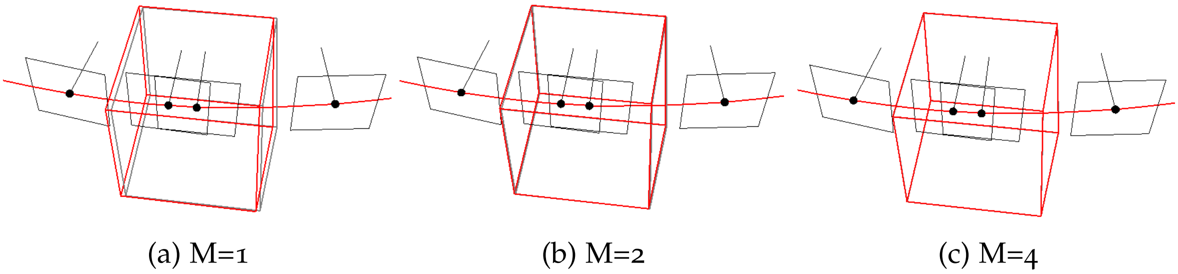
3. Frames $1 \ldots N$ were fed through our standard tracking and mapping pipeline resulting in a complete loop closure around the table. Again, sensor pose estimates are obtained by frame-model ICP.

4. Frames $1 \ldots N$ were fed not just once but repeatedly for $M = 4$ loops to the standard tracking and mapping pipeline. This was possible because the sensor motion was such that frame 1 and frame N were captured from almost the same place.

5. Finally, for comparison, a new longer dataset of MN frames was processed, where a user moved the sensor over the scene without precise repetition.

Our main motivation in performing experiments $2, 3$ and 4 is to investigate the convergence properties of the tracking and mapping scheme, as no explicit joint optimisation is performed.

Figures illustrating the resulting sensor trajectories and reconstructions are given in Figure (9.18). In Figure (9.18a) we demonstrate the result of frame-to-frame depth map tracking, where the pose of each new frame is estimated by registration against just the last frame. Rapid accumulation of errors results in the non-circular trajectory and poor reconstruction is apparent (though see later Figure 9.21a where frame-skipping is shown to improve this).


(a) M=1

(b) M=2

(c) M=4

Figure 9.19: Close up view of loop closing frames in circular experiment as the data from a single loop is repeatedly fed to our system. We see (a) initially good alignment after one pass improving through (b) two passes to finally (c) the frames are extremely closely registered after four passes.

In Figures (9.18b-9.18d) we use the full frame-to-model tracking approach following the settings described above.

In Figure (9.18b) processing is halted with the loop two-thirds complete. Figure (9.18c) shows loop closure, where the last frame processed is a duplication of the first frame and should have an identical ground truth location. We highlight these two frames, and they are seen almost overlapping (red and black) alongside excellent trajectory (see also Figure 9.19) and scene reconstruction quality. Some small artefacts in the reconstruction induced by loop closure can be seen (the diagonal slash across the books in the bottom-right). In Figure (9.18d) we have taken the same data from (9.18c) and fed it repeatedly ($M = 4$ times) to the algorithm to investigate the convergence properties of our system. We now see even better alignment between the loop closing frames, and reconstruction artefacts reduced. Note that this can be compared with the reconstruction from the same number of MN different frames of the same scene obtained from hand-held sensor motion in Figure 9.20. By pushing a single loop sequence of depth maps through the pipeline several times we are able to inspect further the quality of trajectory convergence, since after loop closure the the ground truth poses for the first frame of the sequence are identical. In Figure (9.19) we render the initial frame after $M = 1$, $M = 2$ and $m = 4$ loops of the sequence demonstrating the converging pose estimation.

While the turntable experiments demonstrate interesting convergence of the system without an explicit joint optimisation of the parameters, the real power in integrating every frame of data is the ability to rapidly assimilate as many measurements of the surfaces as are possible, (experiment 5). Figure (9.20) shows the surface reconstruction where $NM = 560 \times 4$ different frames were acquired from a free moving Kinect sensor. While the same algorithmic parameters were used, including reconstruction volume, the increased range of viewpoints result in a reconstruction quality superior to the turntable sequence.

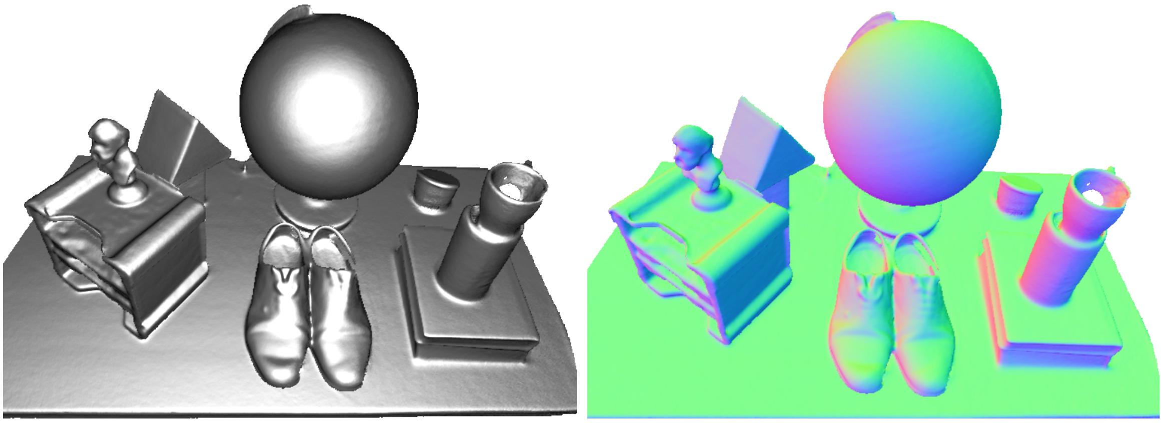


Figure 9.20: Agile sensor motion based reconstruction of the same scene, with the same reconstruction volume but MN different images. Here we see better reconstruction quality due to each depth map offering independent data and a greater range of viewpoints. Model rendered with (a) Phong shading and (b) normal map shading.

A natural extension to a frame-to-frame (scan matching) ICP-based SLAM system is to drop keyframes and perform tracking relative to the keyframe. Using such *anchor* scans reduces drift. This is clearly demonstrated in Figure (9.21a) where we sub-sample the N frames to use every 8^{th} frame only. While the drift is drastically reduced in comparison to Figure (9.18a), the frame-to-model tracking approach presents a drift free result with the same input data as illustrated in Figure (9.21b). Our frame-to-model alignment mitigates a number of hard problems that arise in a fully fledged keyframing system, including deciding where to drop keyframes, and how to detect which keyframe(s) to track from.

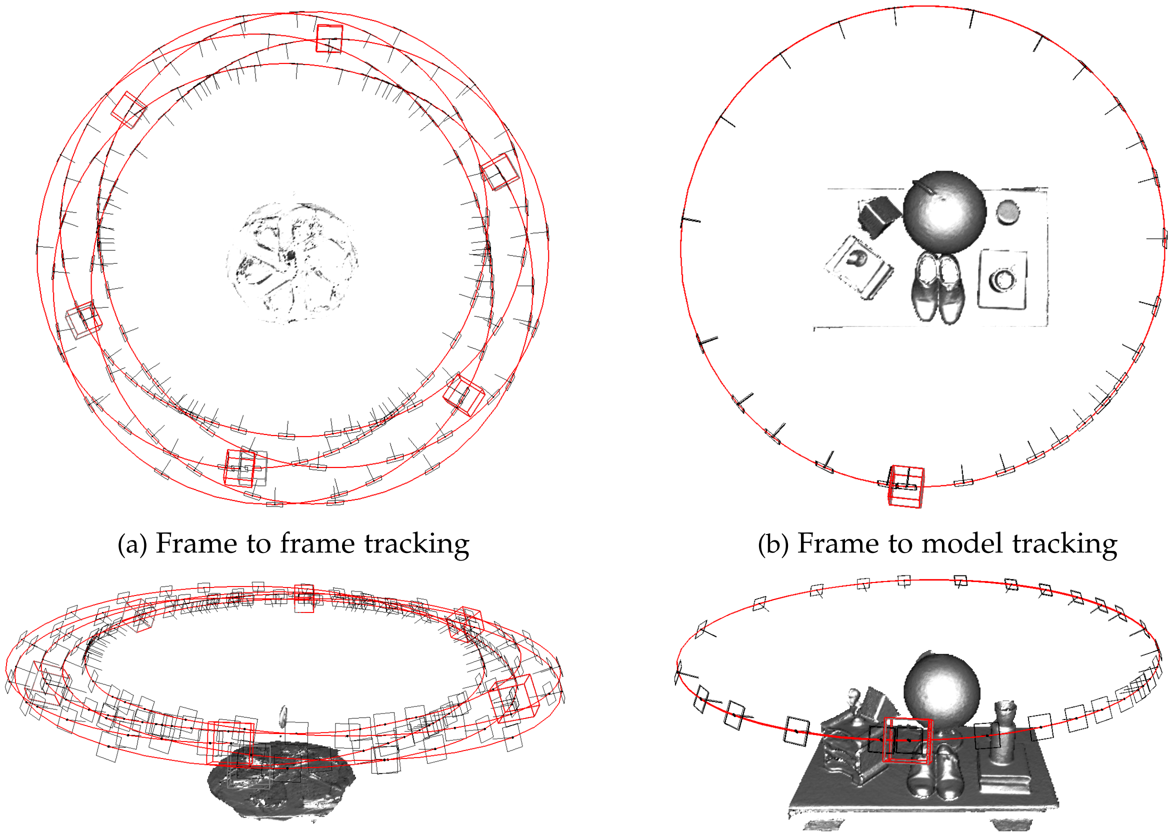
An important aspect of a useful system is its ability to scale with available GPU memory and processing resources. Figure (9.22) shows the reconstruction result where the the N frames are sub-sampled in time to use every 6^{th} frame, and 64 times less GPU memory is used by reducing the reconstruction resolution to 64^3.

Processing Time

Figure (9.23) shows results from an experiment where timings were taken of the main system components and the reconstruction voxel resolution was increased in steps. We note the constant time operation of tracking and mapping for a given voxel resolution independent of scene surface complexity.

Observations and Failure Modes

Our system is robust to a wide range of practical conditions in terms of scene structure and camera motion. Most evidently, by using only depth data it is completely robust to indoor


(a) Frame to frame tracking

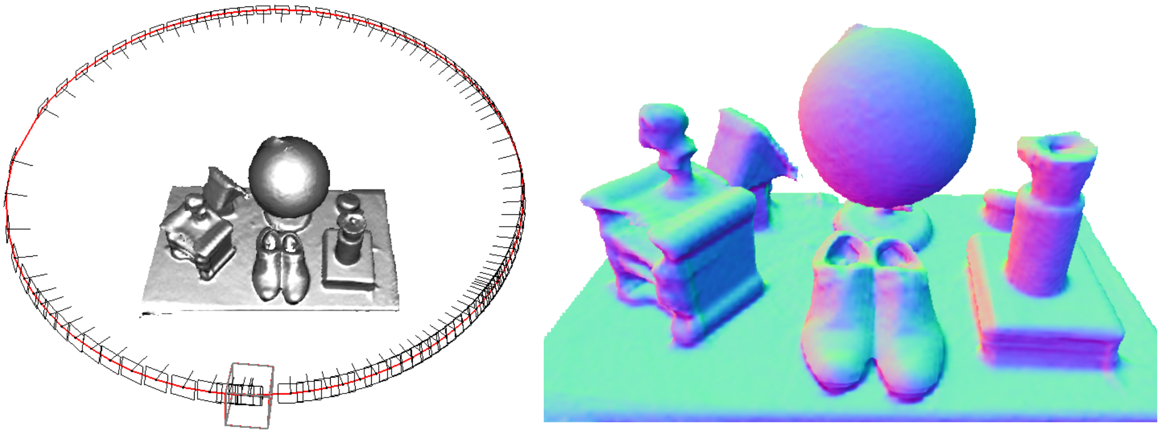
(b) Frame to model tracking

Figure 9.21: (a) Frame to frame vs. (b) frame to model tracking, both using every 8^{th} frame. There is a drastic reduction in drift compared to Figure (9.18a) where all frames are used. But the frame to model tracking results in drift-free operation without explicit global optimisation.

lighting scenarios. The accompanying video of the system in live operation demonstrates a variety of agile motion, with tracking successful through rapid motion. The main failure case in standard indoor scenes is when the sensor is faced by a large planar scene which fills most of its field of view. A planar scene leaves three of the sensor's 6DoF motion unconstrained in the point-plane linear systems null space, resulting in tracking drifting or failure.

9.3.3 Geometry aware AR

The dense accurate models obtained in real-time open up many new possibilities for AR, human-computer-interaction and robotics. For example, the ability to reason about changes in the scene, utilising outliers from ICP data association (see Figure 8.5), allows for new object segmentation methods. These segmented objects can be tracked independently using other instances of ICP allowing piece-wise rigid tracking techniques; and physics can



(a) (b)

Figure 9.22: A reconstruction result using $\frac{1}{64}$ the memory (64^3 voxels) of the previous figures, and using only every 6th sensor frame, demonstrating graceful degradation with drastic reductions in memory and processing requirements.

be simulated in real-time on acquired models directly from the TSDF volumetric representation (see Figure 9.24 and accompanying video). For AR, the dense model also provides an ability to handle truer occlusion boundaries between real and virtual for rendering. We investigated each of these application possibilities in **Izadi et al.** (2011).

9.3.4 Spatially Extended *KinectFusion*

The current system works well for mapping medium sized rooms, objects and workspaces, with volumes of $\leq 7m^3$. However, the reconstruction of large-scale models such as the interior of a whole building raises a number of additional challenges. Firstly, the current dense volumetric representation requires too much memory to enable real-time operation on commodity hardware. More importantly, very large exploratory sequences would lead to reconstructions with inevitable drift which would be apparent in the form of misalignments upon trajectory loop closures. These are classic problems in SLAM with good solutions for sparse representations, but which require new thinking for fully dense modelling. In this subsection we overview new state-of-the-art work which builds on the KinectFusion results discussed in this section.

Sub-Mapped KinectFusion

We have examined the use of basic sub-mapping techniques within the KinectFusion pipeline extending the system to achieve out-of-GPU-core reconstruction capabilities for mapping of small offices and rooms with reduced GPGPU hardware, or extending the size of reconstructions possible on state of the art platforms. The system is limited in practice by the

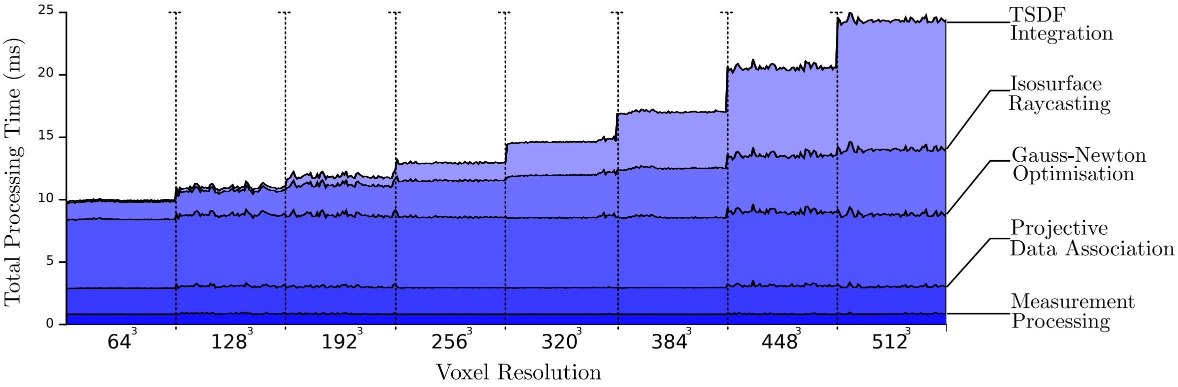


Figure 9.23: Real-time cumulative timing results of system components, evaluated over a range of resolutions (from 64^3 to 512^3 voxels) as the sensor reconstructs inside a volume of $3m^3$. Timings are shown (from bottom to top of the plot) for: pre-processing raw data; multi-scale data-associations; multi-scale pose optimisations; raycasting the surface prediction and finally surface measurement integration.

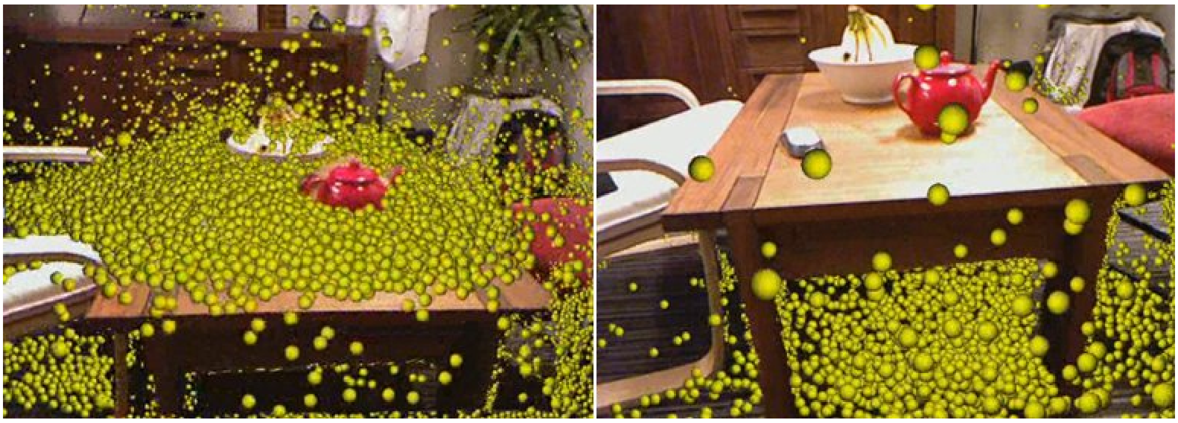


Figure 9.24: Thousands of particles interact live with surfaces as they are reconstructed. Notice how fine-grained occlusions are handled between the real and virtual. Simulation works on the TSDF volumetric representation, and runs on the GPU alongside tracking and mapping, all in real-time.

available storage capacity of the computer main memory hosting the GPGPU device. We focus here on the sub-mapping representation which enables a pose graph optimisation solution to correct for camera drift, which we do not describe here. Example reconstructions are illustrated in Figures (9.25) and (9.26) obtained *without* global pose-graph optimisation.

Hybrid surface representation: Our hybrid scene representation augments an active set of tiled KinectFusion volumes into which new depth maps are fused. Each volume is currently fixed in both spatial resolution and extent at the beginning of the reconstruction. These volumes are combined in the hybrid representation with inactive volumes which are stored in CPU memory and are represented for surface rendering and camera tracking

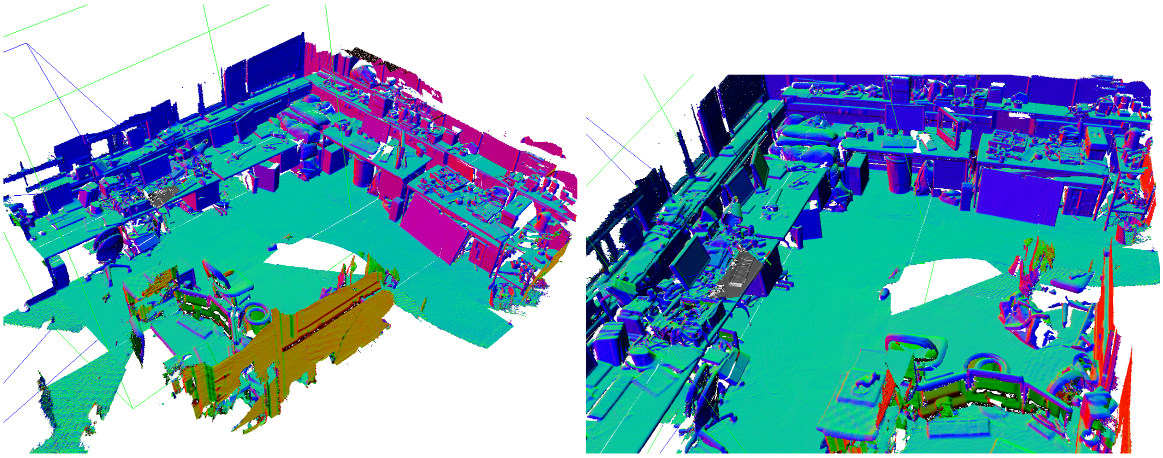


Figure 9.25: Two views from a partially complete reconstruction obtained using our sub-blocking extension to KinectFusion. The live camera is shown with a blue camera frustum together with a surface measurement in silver. The room measures approximately $10 \times 7 \times 3$ meters, and is captured here with a virtual reconstruction volume of 2048^3 voxels.

purposes on the GPU using a triangle mesh of the current level set. Real-time tracking is performed against the complete hybrid surface prediction combining the surface interface extracted from each of the active sub-blocks' zero-crossings together with the surface mesh from inactive blocks extracted using marching cubes.

Sub-block insertion, selection and updating: We initialise a new KinectFusion block by analysing the ratio of data in the new sensor frame not intersecting the current set of blocks. This is efficiently performed by computing the bounding box of the input data and rounding up to the nearest quantised block, then binning the data into sub-blocks and initialising a new sub-block at the the bounds of the block which contains surface measurements projecting to the largest area in the image space. We choose current sub-blocks using the same criteria applied to surface measurements. In practice we can insert and keep active a number of sub-blocks filling the available memory of the GPGPU device. Active sub-blocks are then updated using the standard KinectFusion mapping method. Prior to swapping any currently active sub-block out of GPU memory, we extract the zero-level set for use in the hybrid surface representation. The TSDF volume is then copied into Host CPU memory. Importantly, our dual representation enables revisiting previous sub-blocks, which if activated are copied from host to GPU memory where surface fusion can continue on the TSDF volume.

Towards Scalable Dense SLAM: The sub-block volume and resolution can be altered prior to mapping according to user needs, trading off reconstruction accuracy, real-time tracking performance and reconstruction scale, limited only by the rate of drift in the system. In the near future we plan to utilise a pose-graph based optimisation on sub-block transforma-

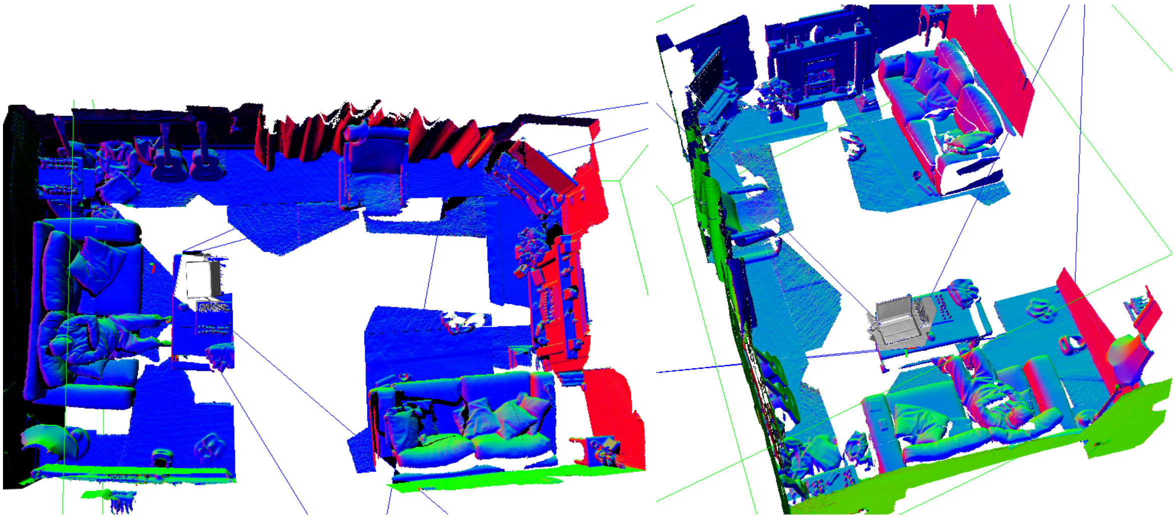


Figure 9.26: Living Room Reconstruction using the key-block extension. Note the partially complete wall shows drift that occurred over the course of the camera trajectory.

tions to achieve large scale globally consistent dense SLAM.

Further Extensions

A number of other researchers have also extended *KinectFusion* to enable larger scale mapping. **Zeng et al.** (2012) exploit the sparsity of the truncated SDF representation, developing a real-time GPU implemented octree formulation and octree raycasting to replace the regular grid approach to storing and rendering the implicit surface. They are able to double the resolution of a given reconstructed volume using the approach, also achieving double the frame-rate for real-time raycasting required for frame-model tracking. However, they do not tackle the important problem of drift and loop closure for larger scale reconstructions possible with the system.

Whelan et al. (2012b) have developed a spatially extended version of KinectFusion, *Kintinuous*. In contrast to the original approach they add a mechanism to move the center of the TSDF volume to ensure reconstruction of unmapped space as the camera frustum moves away from the original working volume. If the camera pose translation exceeds a specified threshold they translate the working volume holding the TSDF in a cyclic buffer. Represented surface regions at the volume boundary which fall outside of the re-centred TSDF are extracted and represented as a point-cloud. They have employed a pose-graph representation of the extracted point-cloud, which together with a loop closure detection mechanism enables correction of drift. Finally, while their system does not represent the surface outside of the current working volume with a TSDF, they have experimented with reintegrating the extracted point-cloud back into the working volume. **Whelan et al.** (2012a) further develop the tracking used in the system, augmenting the depth only ICP tracking, which can fail in areas of few geometric features. They include a vision based direct whole image alignment using techniques similar to those developed in Chapter (8) and used in DTAM.

Roth and Vona (2012) also developed a moving volume KinectFusion system, but focus on the improved visual odomoetry ability provided by the frame to model tracking. Given an updated camera pose relative to the current volume they integrate the new depth map and then remap the entire volume using a trilinear interpolation into the new camera pose frame of reference. Therefore, the working volume remains fixed to the live camera frame of reference.

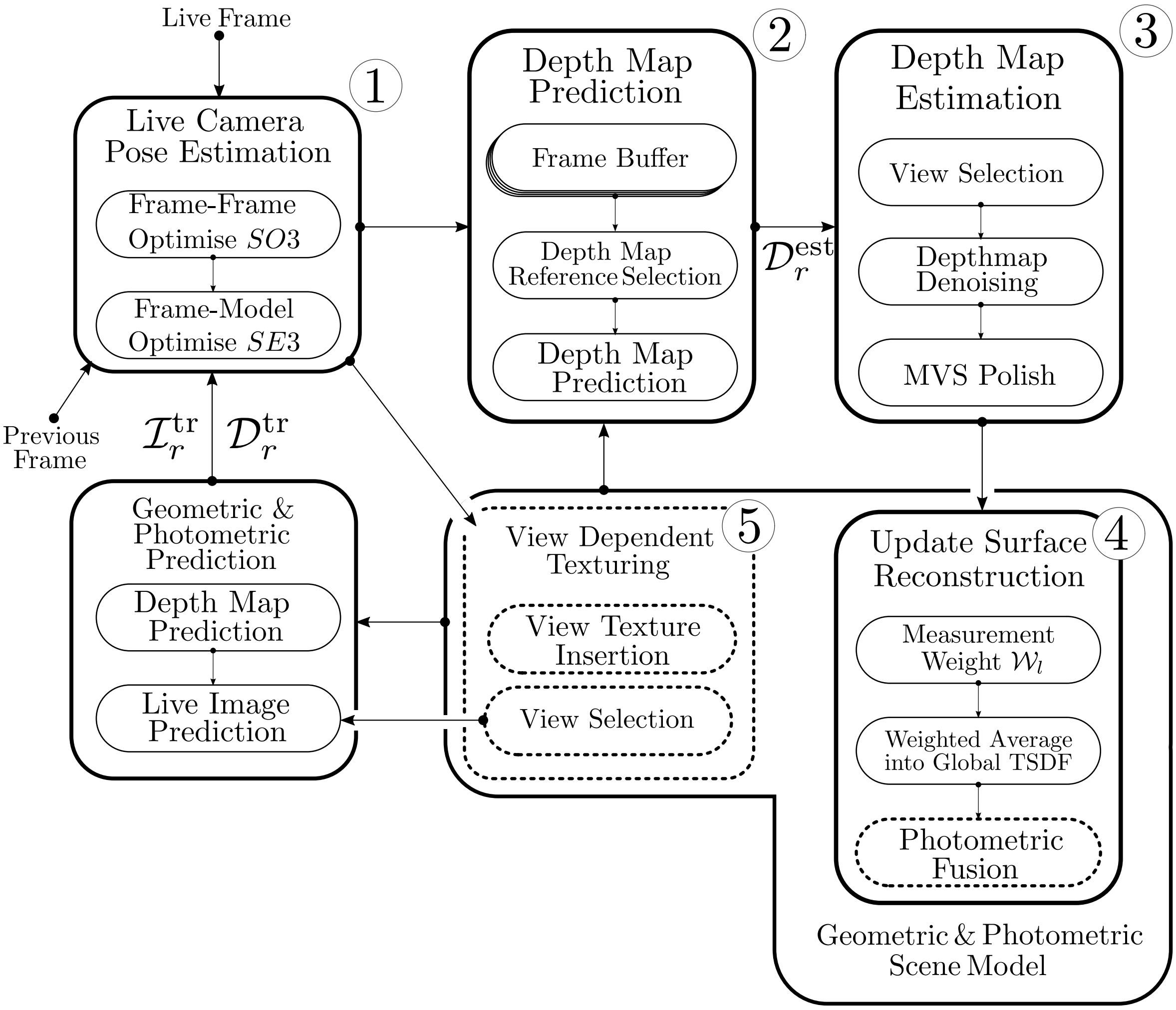

Live Frame
1
Live Camera Pose Estimation
Frame-Frame Optimise $SO3$
Frame-Model Optimise $SE3$
Previous Frame
$\mathcal{I}_r^{\mathrm{tr}}$
$\mathcal{D}_r^{\mathrm{tr}}$
2
Depth Map Prediction
Frame Buffer
Depth Map Reference Selection
Depth Map Prediction
$\mathcal{D}_r^{\mathrm{est}}$
3
Depth Map Estimation
View Selection
Depthmap Denoising
MVS Polish
Geometric & Photometric Prediction
Depth Map Prediction
Live Image Prediction
5
View Dependent Texturing
View Texture Insertion
View Selection
4
Update Surface Reconstruction
Measurement Weight $\mathcal{W}_l$
Weighted Average into Global TSDF
Photometric Fusion
Geometric & Photometric Scene Model

Figure 9.27: Passive Fusion Pipeline Overview.

9.4 Surface Fusion and Tracking from Real-time Video

We now return to real-time dense reconstruction from live video obtained from a single moving camera. The principal advantage of a completely passive single camera pipeline is an ability to work across reconstruction scales. This is in contrast to active depth cameras and fixed multi-camera devices, which depending on the technology used, have limitations on both the short and long range capabilities of depth measurement. In Chapter (7) we demonstrated that accurate *dense* reconstructions can be achieved on commodity hardware using a single live moving camera as the only measurement device. In this last chapter we provide preliminary results for a system that combines the real-time multi-view stereo pipeline with direct whole image alignment based camera tracking. The resulting system is similar in principle to KinectFusion, where all available image data is used in a feature-less manner for both dense reconstruction and tracking.

9.4.1 Method

As outlined in Figure (9.27), the dense surface fusion and tracking from real-time video pipeline combines the depth map fusion system detailed in Chapter (7) with the direct model to whole image alignment approach developed in Chapter (8). As with KinectFusion, the passive image pipeline uses the geometric prediction available from the current dense reconstruction, but also requires a dense photometric prediction which we obtain here using either the photometric fusion or view dependent texture mapping (VDTM) approach detailed in Chapter (6).

Tracking the Live Image

The dense tracking component, Figure (9.27)(1), uses the current dense model with both a photometric and geometric prediction following the direct method described in Section (8.2) using the frame-frame SO3 and frame-model SE3 estimation methods. Recalling the use of a conservative surface estimation from the current implicit surface described in Section (7.4.2), where we threshold the surface prediction using the per voxel weight, we further re-weight the per-pixel SE3 direct alignment error function using the current surface model confidence extracted at the surface interface:

$$e(\mathbf{x}, u) = \mathcal{W}_r^{tr}(u)(\mathcal{I}_l(\mathbf{w}_{SE_3}(u, \mathbf{x})) - \mathcal{I}_r^{tr}(u)) \ . \tag{9.4}$$

Here $\mathcal{W}_r^{tr}(u)$ is the predicted confidence of the surface element rendered into the tracking reference frame pixel u, and minimisation of the whole image error proceeds under a chosen penalty function for parameters $\hat{\mathbf{x}} \in \mathfrak{se}_3$. Failure to achieve tracking convergence leads to initialisation of the keyframe based re-localisation mechanism with the last known pose. Normal tracking resumes after convergence of the pose estimate from a keyframe as described in Section (8.5).

Photometric Prediction

Photometric prediction is achieved either using photometric fusion or VDTM. When using VDTM we use the titled marching cubes extraction on every other frame to obtain a surface model that is efficiently rendered into the set of selected texturing frames, ensuring we keep the predicted geometry associated with the texture up-to-date. Given the live pose we then decide whether to add the frame into the VDTM key-frame set using the insertion mechanism, Figure (9.27)(5), described in Section (6.4.2).

Predictive Depth Map Fusion

Successfully tracked frames are added into the dense reconstruction image buffer, which together with a predicted depth map rendered into the new reference frame establishes the required input for the multi-view predictive depth map fusion pipeline components shown in Figure (9.27(2-4)) and detailed in Chapter (7) which uses the predictive multi-view stereo methods from Chapters (4) and (5). If using the photometric fusion mechanism for texture modelling, we further integrate the associated depth map reference image into the appearance component of the volumetric reconstruction at this time.

System Initialisation

The fully dense tracking and mapping pipeline requires an estimation of the camera pose for at least two frames from which an initial depth map can be estimated and fused into the volumetric surface representation, along with providing a calibrated texture frame to enable direct whole image alignment. In practice, as in the DTAM system (Section (9.2), we make use of the high quality feature-based pose estimation from the PTAM system. After 2 to 3 seconds of calibrated image input to the depth map fusion pipeline we then switch to the fully dense pipeline.

9.4.2 Preliminary Results

We provide preliminary results captured from the live operating system. In all cases, video input was set to a resolution of 640×480 with a frame rate of $30Hz$ from a point-grey flea2 device. We use automatic exposure control, and fix all other settings of the camera including gain and colour balance, photometrically calibrating the camera with the fixed settings using the technique outlined in Section (3.2). Each video image is therefore photometrically normalised enabling the photo-consistent reconstructions using either the photometric fusion or view dependent texture mapping pipelines, used in direct tracking, to work with the varying exposure of each frame. Specific details and comments on each of the reconstructions obtained are provided demonstrating operation on a range of indoor office scenes including a cluttered desktop in Figure (9.29d), close up reconstruction of a toy pangolin (9.28), the inside of a computer with numerous cables (9.30) and a larger scale office view in Figure (9.31).

The figures illustrate the typical quality of reconstructions possible, ranging in reconstructions with bounding volumes from $0.2 \times 0.1 \times 0.1m^3$ to $3 \times 3 \times 2.5m^3$ which we further demonstrate in the accompanying videos. We note that we provide these reconstructions to demonstrate not only the modelling capabilities of the surface representation used, but also by demonstrating the consistent, drift free, which result from the *combination* of the

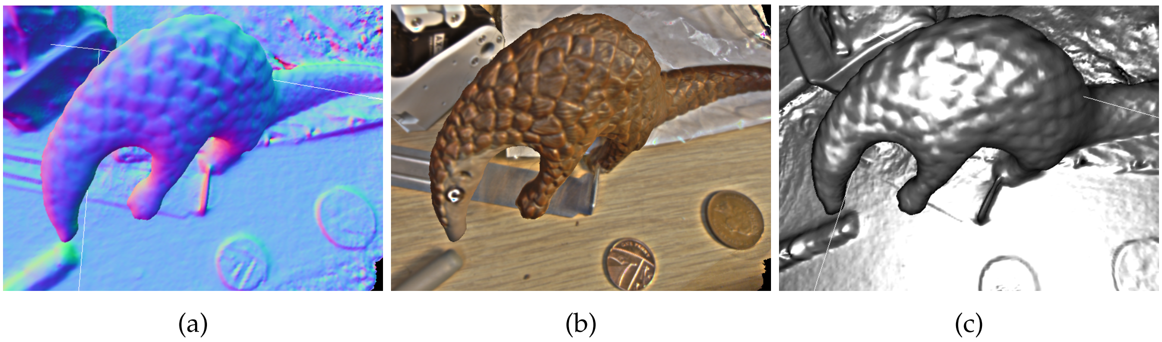


Figure 9.28: Reconstruction of a toy pangolin on desktop. Demonstrating the small scale, high detail reconstruction possible with a single camera. The reconstruction bounds were approximately $0.2 \times 0.1 \times 0.1m$ which was represented using a volume with a resolution of $480 \times 256 \times 256$ voxels. We show the normal map and Phong shaded rendering of the reconstructed geometry in (a,c), together with the photometric prediction for a virtual in (c). Here we have made use of the view dependent texture map based photometric modelling. The scene contains a number of challenging surfaces for passive dense reconstruction. Notably reconstruction of the desktop surface and mixed paper-cellophane bag (back of the pangolin) and back-plate of a computer (beneath the pangolin) have been reconstructed, despite their secularity; this is a consequence of using very short baseline imagery from the live video in the multi-view stereo pipeline, which can make use of the weak data-terms by using multiple nearby frames. The reconstruction required approximately 1 minute of real-time operation.

dense tracking and reconstruction components, we illustrate the system performing dense visual SLAM with a single camera.

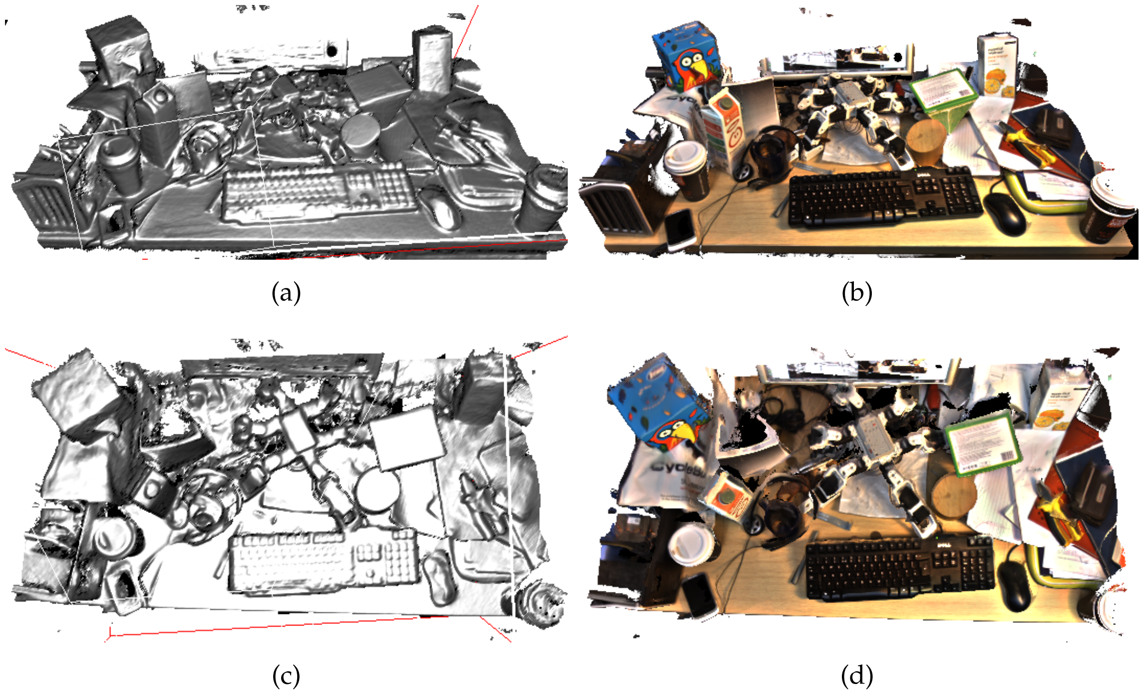
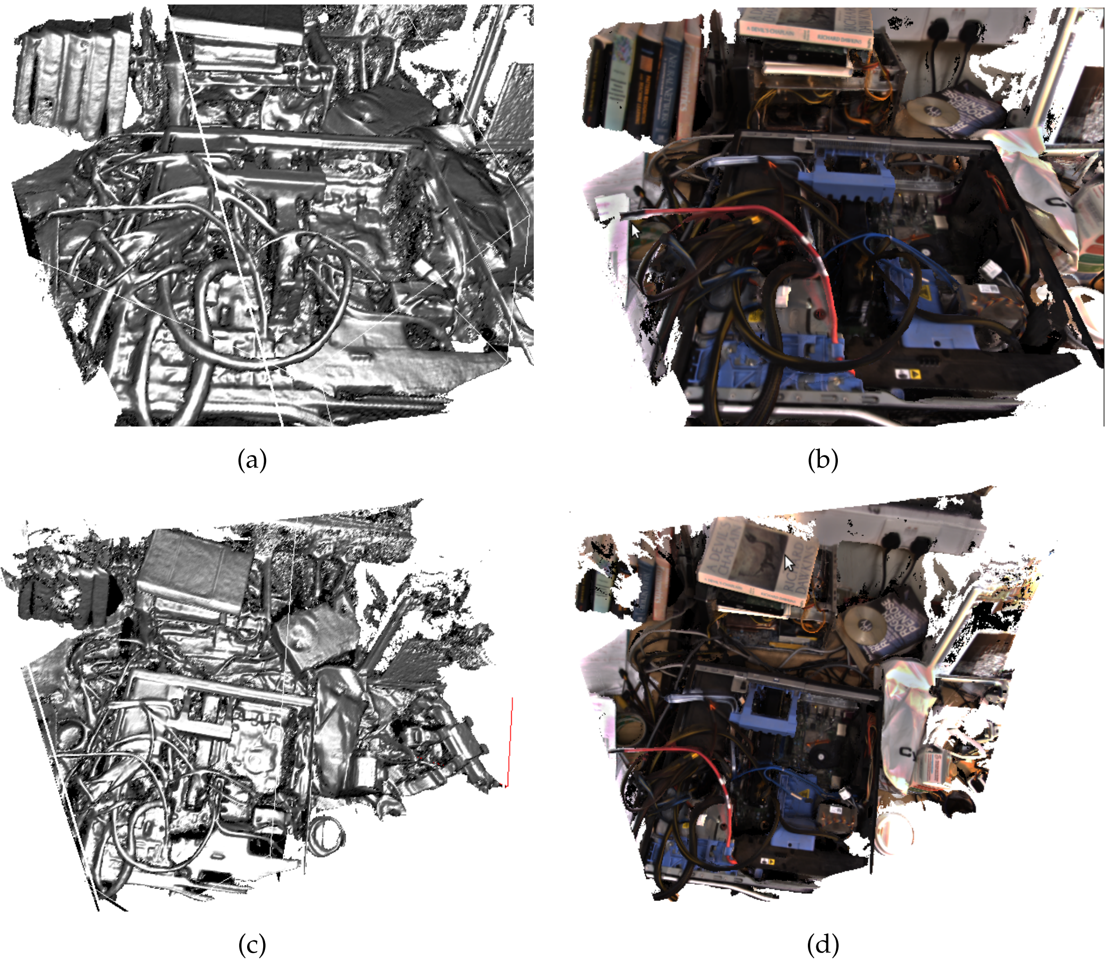


Figure 9.29: Reconstruction of a cluttered desktop scene. Providing a basic comparison point with the previously described live dense reconstruction systems. A resolution of $480 \times 384 \times 384$ voxels was used to represent the scene shown with a bounding volume of approximately $1.5 \times 1 \times 1m^3$, we note regions below the desktop have been clipped for the illustration. Here we demonstrate photometric fusion for tracking, showing the geometry with Phong shading in (a,c) and corresponding photometric predictions for the virtual views in (b,d). This example demonstrates the pipeline operating to reconstruct surface regions varying texture characteristics, requiring approximately 1 minute reconstruction time, with a loopy motion, repeatedly visiting areas in the scene. Notably, the desktop surface texture is clearly non discriminative (also shown in a close up reconstruction in Figure (9.28)); while numerous objects with homogeneous texture on the desktop include white paper and a purpose printed gradient image, a reference wooden cylinder with a slanted top and dark paper coffee cups; all of which are successfully reconstructed despite having few discernible features. This ability stems from the use of ultra-small baseline imagery used in the multi-view depth map estimation on a scene with constant global illumination. In this setting the convex depth estimation methods are able to use small variations in the pixel intensity. We note that the the highly specular surfaces in the reconstruction: a mobile phone screen (bottom left) and the keyboard and mouse, do show reduced reconstruction quality.



(a) (b)

(c) (d)

Figure 9.30: Partial reconstruction of a computer with many small cables. This scene presents a number of challenges for passive dense reconstruction, and demonstrates well the power the implicit surface representation used, which effortlessly handles the intricate topology of the scene geometry. The scene volume of approximately 0.7 × 0.7 × 0.5 was captured with a resolution of 384 × 384 × 256 voxels, requiring approximately 30 seconds real-time operation. This reconstruction again demonstrates in several reconstructed regions of the scene that dense multi-view stereo using the ultra-small baseline imagery provided by video rate data enables reconstruction of very low texture objects. In this scene, we see a mixture of cables as well as near-texture-less black and blue plastics, correctly reconstructed.

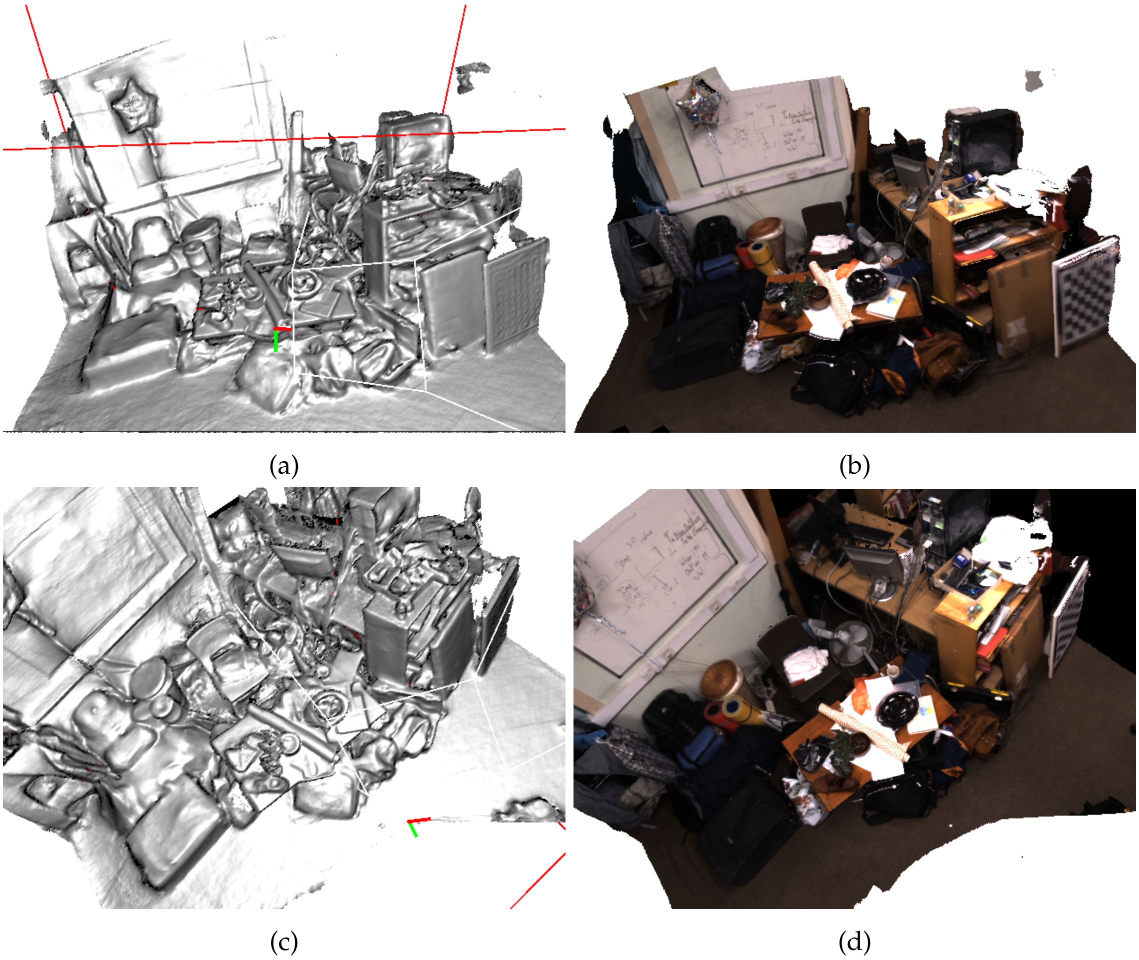
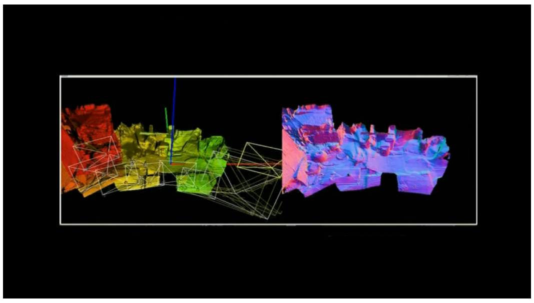


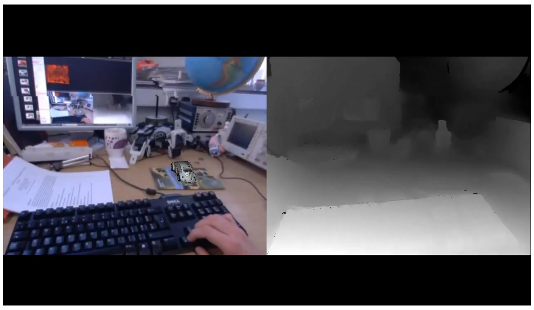
Figure 9.31: Reconstruction of an office scene. We demonstrate a larger scale reconstruction in this example, noting the different range of reconstruction scale captured between this scene and coins on the desktop in Figure (9.28). Here the reconstruction captures a volume of approximately $3 \times 3 \times 2.5m^3$ using a resolution of 480^3 voxels. We note a number of useful results in the reconstruction. There is further demonstration of successful reconstruction in homogeneously textured regions such as the floor; chair; table and objects; black suitcase on the floor; and cardboard boxes; but also illustration of reconstruction capability in repeated textured regions shown for the chequerboard pattern, although the embossing artefacts in the reconstructed surface of that region shows the active use of the image weighting in the multi-view stereo pipeline. The browsing camera remained at least 1 meter from the far wall during reconstruction resulting in the lower quality reconstruction of the white board and wall there, but it is not clear that a high quality of reconstruction and tracking would be obtained by moving closer to that region, since the board presents a large specular surface. Reconstruction has not been successful for the highly specular computer case (top right), although the photometric prediction of the region (b,d) still demonstrates high photo-consistency. This reconstruction required approximately 2 minutes of scene browsing.

9.5 Video Appendix

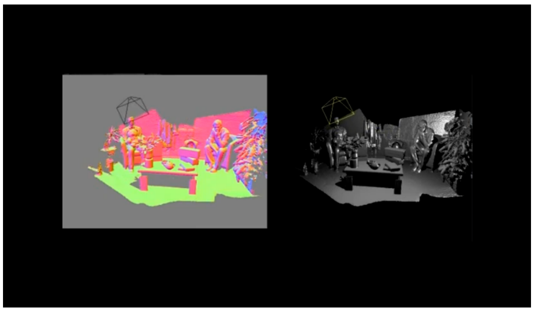
This appendix lists the accompanying videos for demonstrations from systems in this chapter and also for the demonstration of a number of working components from chapters throughout this thesis.



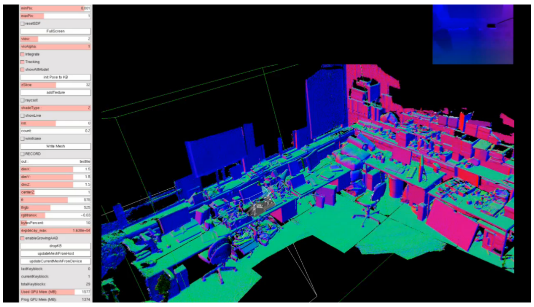
Live Dense Reconstruction with a Single Moving Camera. *CVPR 2010*, **Newcombe and Davison** (2010).
http://youtu.be/CZiSK7OMANw



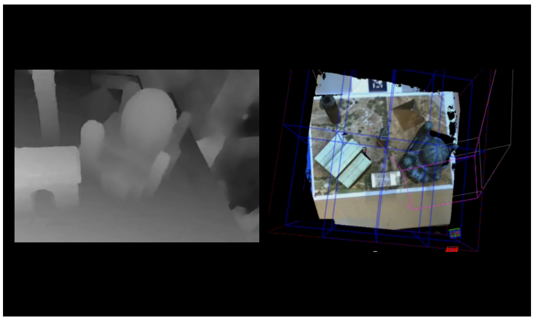
DTAM: Dense Tracking and Mapping in Real-time. *ICCV 2011*, **Newcombe, Lovegrove, and Davison** (2011c).
http://youtu.be/Df9WhgibCQA?hd=1



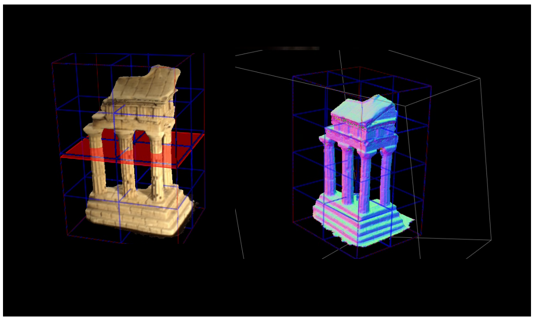
KinectFusion: Real-time Dense Surface Mapping and Tracking. *ISMAR 2011*, **Newcombe, Izadi, Hilliges, Molyneaux, Kim, Davison, Kohli, Shotton, Hodges, and Fitzgibbon** (2011b) and KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. *UIST 2011*, **Izadi, Kim, Hilliges, Molyneaux, Newcombe, Kohli, Shotton, Hodges, Freeman, Davison, and Fitzgibbon** (2011),
http://youtu.be/q8jfgTilFmo



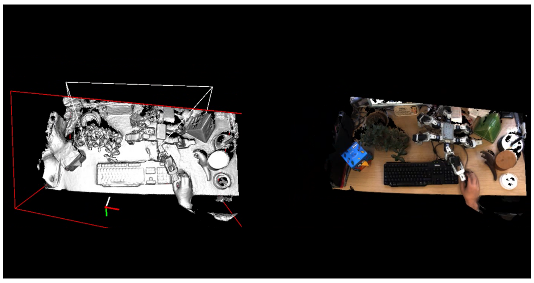
Key-Block extension to KinectFusion outlined in Section (9.3.4)
http://youtu.be/InYNITya7zg?hd=1



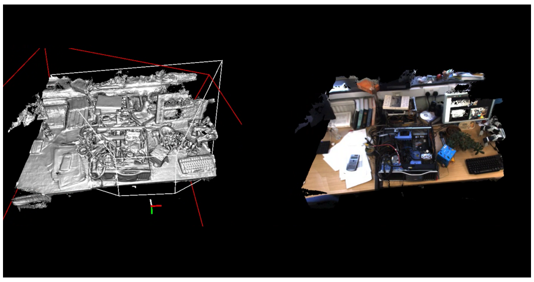
Incremental reconstruction of the Graz City of Sights, using the video rate incremental LDR detailed in Chapter (7), with view dependent texture mapping from Section (6.4.2).
http://youtu.be/eWoHMfRjo-M?hd=1



Incremental reconstruction of the Middlebury MVS Temple dataset. Also using the video rate incremental LDR detailed in Chapter (7), with view dependent texture mapping from Section (6.4.2).
http://youtu.be/6fiGePRp6HU?hd=1



Incremental desktop reconstruction using the *Surface Fusion and Tracking* pipeline from real-time video outlined in Section (9.4)
http://youtu.be/qBEI0yKih18?hd=1



Incremental desktop with computer reconstruction using *Surface Fusion and Tracking* pipeline from real-video pipeline outlined in Section (9.4)
http://youtu.be/FKtJ-5yuGxs?hd=1

10

Conclusions

In this thesis we have brought together several branches of research from the field of computer vision to develop systems capable of dense visual SLAM. Dense tracking and mapping components naturally arise from the representation of a scenes *surfaces*, while feature-based methods more naturally arise from an abstraction of the surface to a point cloud model. There has long been interest in the relative merits of feature-based and dense (direct) methods (**Irani and Anandan**, 1999; **Torr and Zisserman**, 1999; **Triggs et al.**, 1999), but unlike feature based techniques, the use of dense methods within visual SLAM has been limited. The restricted processing resources imposed by real-time operation seemed to preclude dense methods in previous monocular SLAM systems, and indeed the recent availability of powerful commodity GPGPU processors is a major enabler of our approach in both the reconstruction and tracking components. While the benefits of a dense surface representation for applications which require surface estimates is clear enough, we believe that there has been less understanding of the advantages dense models can provide within online visual SLAM. We have seen that the availability of a dense scene model, all the time, enables many simplifications of issues with point-based systems, for instance removing the need for wide baseline feature matching; enabling reconstruction of surfaces in low texture regions; naturally handling occlusions in tracking; and providing graceful degradation for tracking when presented with image blur due to rapid motion.

10.1 Contributions to Dense SLAM

Dense Surface Measurements

In Chapters (4) and (5) we developed techniques to estimate a real-time depth map from a single moving camera. The multi-view stereo depth maps are computed by exploiting the massive amounts of information available in calibrated short-baseline video together with a prior assumption about the resulting smoothness of the depth map. Specifically we used a variational framework to enable depth maps to be obtained as the solution of an optimisation process, combining data and regularisation terms, (**Pock**, 2008). This has allowed us to incorporate our prior assumptions about the smoothness of physically realistic depth maps; to resolve structure information in the video images that would otherwise be ambiguous or noisy. In particular we have built on solutions using continuous convex optimisation based formulations, using primal-dual models that combine a data and regularisation term that provide a trade-off between efficient computability and model accuracy.

Within this framework we presented three convex optimisation based multi-view stereo depth map estimation methods that take as input calibrated video. Our depth map denoising (DMD) approach explicitly extracts and denoises a data term only depth map. The data term depth map is efficient to compute and can leverage explicit occlusion handling over the small-baseline video rate data. We then looked at the minimisation of the full multi-view stereo energy functional. We used a linearisation of the multi-view stereo error function and presented a modern primal-dual multi-image solution to the classical variational stereo problem. We made use of the DMD result to initialise the solution in place of the traditional coarse-to-fine optimisation strategy. Finally, we investigated an alternative solution to the full MVS depth map energy functional that replaces the linearisation of the error function with an exact search over the solution space by taking advantage of the ability to solve two constrained optimisation problems in alternation. In each case, the abstraction from the image to the geometric *surface* measurement, in the form of a dense depth map, demonstrated that far more structure was available from a real-time single moving camera than is so often extracted in feature based visual SLAM systems.

Non-Parametric Surface Models

In Chapter (6) we built on the incremental surface reconstruction and prediction capabilities of the volumetric signed distance function range fusion approach by **Curless and Levoy** (1996). This reconstruction technique has been used to great effect in offline active measurement and multi-view stereo reconstruction methods, where the implicit surface enables the capture of arbitrary surface topology. This power to represent intricate real-world surfaces, is accompanied with the ability to perform incremental surface fusion of new measurements into the reconstruction, making use of the spatial structure and free-space information available in projectively acquired depth map measurements. Furthermore, the implicit surface representation enables both high quality representation and fast prediction of the surface in an any-time manner through raycasting the iso-surface, enabling synthetic views of a partial surface reconstruction to be rendered into a given camera frame. This combined ability of surface representation, integration and prediction, all available at a constant time cost for a given volume – independent of the complexity of the model within the volume, provides the measurement prediction and map updating properties required for a dense SLAM pipeline. In that chapter we also developed photometric prediction mechanisms that augment the geometric model to provide photo-consistent predictions of the scene into novel views.

In Chapter (7) we combined the video rate depth map estimation from Chapters (4) and (5) with the incremental surface reconstruction and demonstrated incremental live dense reconstruction. The ability to produce a prediction of surface geometry from the current reconstruction allows us to do more than simply passively integrate the measurements. Instead we demonstrate the use of the implicitly captured uncertainty in the surface representation; providing a bound on the data term search used in the multi-view stereo estimation process. This results in a reduced computational cost of the dense depth map estimation, but also reduces ambiguity in the data term enabling reconstruction of less highly textured surfaces. The resulting live dense reconstruction system made use of calibrated video input, with camera pose estimates provided by PTAM (**Klein and Murray**, 2008). We point out that such systems will require new methods of evaluation for real-world use over comparison with the traditional offline statically acquired multi-view stereo datasets. The availability of real-time feedback from the reconstruction process results in a dynamic operation with the user attempting to fill-in the reconstruction depending on the needs of the application and properties of the scene being modelled, e.g. regions of less texture or fine structure require different camera operation to those in more highly textured or geometrically homogeneous areas.

Insights from Short-Baseline Stereo and Live Dense Reconstruction

We have found that the combination of video rate short-baseline stereo estimation, together with continuous surface fusion, results in a non obvious capability to improve reconstruction of scene regions that can posses smooth or repeating textured surfaces; surfaces with reflections beyond pure diffuse; and containing wiry or otherwise self occluding geometry.

Specifically, each of these produce problems for MVS systems due to the difficulty of obtaining or resolving correct correspondence between image regions across multiple views. Clearly, changes in lighting in the scene caused by shadowing, other active lighting sources, or when the surface reflectance is not only Lambertian, cause photometric changes in image appearance. In attempting to achieve robustness to such variations, descriptors trade-off against a reduction in the discriminability of an image region. Robustness to such variation is often achieved by removing low frequency information in a region, however when the image region is only weakly textured or smooth (and the signal is contained in the low frequency components), the signal to noise ratio of the descriptors decreases. The size of descriptor can also be increased, but this often encodes an assumption about the shape of the surface in the region, and reduces the spatial precision with which correspondence can be attained.

Fortunately at video frame rate or more generally when capturing with a very short-baselines, image variations due to illumination changes are greatly reduced between views. In this scenario even what appears to be homogeneously textured surfaces may in fact have useful low frequency gradient information caused by shadowing, global lighting or surface texture. Such information is informative for forming a consistent mode in the stereo photometric likelihood When comparing image regions over short baselines using absolute differences with a single pixel, or otherwise with a very small image patch, and weak illumination invariance.

Short-baseline stereo observations also result in reduced geometric distortion of observed surfaces between views and increase the co-observability of surfaces. This results in an increased signal to noise ratio when using the simple single pixel or small patch data terms, enabling smaller structures and fine details or wiry objects to be brought into correspondence. The ability to integrate multiple observations from many different angles with a short-baseline within a live dense reconstruction frame provides a means for users to further reduce any aperture problems. Given real-time feedback users can mitigate the issue by moving the camera to obtain any usefully constraining overlapping views.

Integration of depth maps across multiple short-baselines can further enable estimation of surfaces with specular reflections as long as there is a diffuse component to the surface reflection. As more views are integrated, the number of multiple views that consistently observe the diffuse reflection in the same (correct) location produces a minima in the photometric cost function which (ignoring saturated pixel values) corresponds to a surface point that has greater photoconsistency than any of the minima caused by specularities, since the diffuse component colours are consistent across views while specularities can change colour (**Lee and Bajcsy**, 1992; **Lin et al.**, 2002).

Whole Frame Tracking

In Chapter (8) we developed dense tracking pipelines that use the ability to predict the geometry and appearance of a surface model into every sensor pixel. The result is the replacement of the feature extracting, matching and tracking pipeline of the sparse visual SLAM systems with the optimisation of the camera pose using a direct formulation of photometric image error. Our passive camera tracking approach builds on the 2.5 model alignment work from (**Baker et al.**, 2004b) to develop the full 6*DoF* model-to-frame tracking. We also developed the dense depth map tracking pipeline in the context of tracking a commodity depth camera, also formulating the frame-to-model dense tracking approach, showing the relationship between the direct passive methods and dense iterated closest point methods.

Real-time Dense SLAM Systems

In Chapter (9) we demonstrated the use of these components in new dense SLAM systems. We first presented our earlier live dense reconstruction system that extended PTAM, making use of the video rate pose estimation and sparse point based scene model from which an overlapping set of depth maps were attached. We then presented three new dense SLAM systems that show an incremental progression towards a fully dense visual SLAM system. In DTAM, we brought dense tracking into the loop, making use of the dense geometric and photometric predictions available from textured depth maps. In KinectFusion we explored our fully dense, incrementally updatable SLAM pipeline exploiting the availability of commodity depth cameras to provide the depth measurements. We moved beyond the patchwork of depth map scene representation to combine the volumetric SDF fusion and prediction framework with dense ICP based depth map tracking and demonstrated that it is possible to obtain a consistent constant time dense SLAM result, without any form of explicit feature extraction and matching. Finally we demonstrated that the same approach can successfully be used in a single moving camera setting combining the real-time predictive multi-view stereo, surface integration, prediction and tracking to produce a dense visual SLAM pipeline.

All of the dense SLAM components have been developed in the context of utilising the massive compute capability of commodity general purpose parallel hardware that developed from the graphics card processing pipeline (**Nvidia**, 2008). Specifically, each of the components described in this thesis has a large amount of fine grained parallelism. Such computations require only local interaction of the solution values obtained in simple static homogeneously laid out memory structures. This is the case for the data term computation and optimisation of the variational multi-view stereo methods; the surface integration and prediction mechanisms; and also the whole-image error computations required in the dense tracking components. These dense SLAM pipelines will continue to benefit from the advances made in massively parallel hardware over the coming years.

10.2 Future Work

There are number of exciting research questions and engineering directions we are interested in taking with this work which we now outline.

10.2.1 Formulating Online Dense SLAM

Looking at both the KinectFusion system from Section (9.3) and the equivalent single passive camera system presented in Section (9.4), we note the alternating form of optimisation over the dense surface model and camera parameters that it performs: given the current estimate of the dense surface the camera parameters are estimated and fixed; then, given a previously fixed set of camera poses the surface model is updated. This form of SLAM with structure and motion estimation factorised into independent optimisations is clearly only optimal under the assumption that each optimisation does yield an optimal estimate. However, since the system clearly demonstrates the ability to operate drift free under the various office environments demonstrated more work must be done to understand under what conditions such a simple optimisation can achieve consistent mapping, not least because such a factorised approach to tracking and mapping is very efficient.

We must therefore develop formulations for the complete dense SLAM problem, reflecting the bundle adjustment formulation for the feature based joint optimisation approach. The formulation must express the full explicit joint optimisation for a consistent and global dense surface model together with parameters of a camera trajectory. Given formulations of the full dense SLAM problem, we can begin to understand how we might achieve an online optimisation for dense SLAM in a principled way; whether through a form of filtering, keyframe representation or lower level sparsification of the resulting optimisation as is often performed in SLAM (**Dellaert and Kaess**, 2006; **Thrun et al.**, 2005). What is made clearer by the results from this thesis is that a solution to online dense SLAM does exist. Furthermore, given the results of the systems developed in this thesis that achieve consistent mapping and tracking using the most basic SLAM factorisation possible, we should be optimistic of the potential benefits that a more principled formulation might bring.

10.2.2 Getting More from Each Pixel

Beyond developing a principled dense SLAM formulation, we also look toward incorporating lower level physical priors into a dense visual SLAM system. A single moving camera is an incredibly rich source of information and the dense visual SLAM components developed in this thesis only make use of some of the information available in a pixel. Specifically an assumption of Lambertian surfaces is pervasive throughout the work, and yet the dense visual SLAM results from Section (9.4) demonstrate that it is possible, by exploiting the video rate data, to perform dense surface reconstruction even when there is a large specular component in the image of the non-Lambertian regions. One possible direction is therefore to exploit the ability to reconstruct these semi-specular surfaces and to use the surfaces themselves as a further means of observing structure in the scene which is not directly observable. For example, surface light fields (**Wood et al.**, 2000), which require an estimated surface geometry proxy, enable efficient representation of all the surface-light structure interactions. Given such a surface light field capture, it is possible to further decompose the surface reflectance into a diffuse and specular component and use the specular component to infer the positions and properties of the non directly observed lighting structures in the scene. Such a rich reconstruction of the scene would provide further contextualisation of the dense data terms used throughout the components in this thesis.

10.2.3 Sparsifying Dense SLAM: Pixels, Surfaces and Objects

A number of interesting avenues of research are open in understanding how we might make better use of the available data in all components of the dense SLAM pipeline. Within the dense tracking pipelines, there is an opportunity to reduce the massive amounts of redundancy in the normal equations, by deciding on a frame by frame basis which pixels would provide maximum information for robust and accurate solutions (**Dellaert and Collins**, 1999).

Within dense mapping we previously noted the potential for using more efficient representations of the signed distance function, exploiting the compressibility of the implicit surface representation to scalable mapping. Such extensions were discussed in Section (9.3.4), noting recent extensions to KinectFusion including the signed distance function octree work by **Zeng et al.** (2012). However, representing the dense surface in such a global form presents a serious challenge in handling drift that can be baked into the map. Issues regarding drift in global representations of the dense reconstruction must be resolved to make such a scalable representation truly useful.

Perhaps the most exciting direction of research, building on the ability to obtain high quality dense reconstructions in real-time, is the possibility to begin to extract and exploit higher level structures from the non-parametric representation (**Pauly et al.**, 2008; **Cohen et al.**, 2012). By detecting possible symmetries and repetition in the scene it is possible to introduce, when supported by the data, parametric abstractions of the scene even up to the level of whole objects, rooms and buildings. For instance, having reconstructed a chair in the scene it would be very useful if later observations of a similar chair type elsewhere could provide a strong prior on reconstruction, or enable tracking capabilities on objects which are too far from the sensor to provide useful data for live dense reconstruction from which the camera can be tracked. Such a system would require two new components in the dense SLAM pipeline: one which actively looked to compress the current scene reconstruction, resulting in a higher level basis representation or segmentation of those representable regions of the scene; and second a mechanism that enabled detection of the objects in the scene that be used to contextualise the reconstruction to make use of the extraction structures. Dynamically building an object hierarchy from surface elements to object primitives on-line, presents enormous opportunities to compress the dense scene reconstruction, handle moving objects in the scene, enable reconstruction of object which have only few or low quality observations, and enable scalable life long mapping and tracking capabilities.

.

Bibliography

M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point Set Surfaces. In *Proceedings of IEEE Visualization* (2001). 6.1.2

L. Álvarez, R. Deriche, J. Sánchez, and J. Weickert. Dense Disparity Map Estimation Respecting Image Discontinuities: A PDE and Scale-Space BasedApproach. In *Proceedings of the International Conference on Machine Vision Applications* (2000). 2.3.2

C. Audras, A.I. Comport, M. Meilland, and P. Rives. Real-time dense RGB-D localisation and mapping. In *Australian Conference on Robotics and Automation* (2011). 2.5

S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework: Part 3. Technical report, Carnegie Mellon University (2003a). 8.1.2

S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework: Part 4. Technical report, Carnegie Mellon University (2004a). 8.1.2

S. Baker, R. Gross, I. Matthews, and T. Ishikawa. Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical report, Carnegie Mellon University (2003b). 8.1.2

S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision (IJCV)*, 56(3):221–255 (2004a). 8.1.2

S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 1. *International Journal of Computer Vision (IJCV)*, 56(3):221–255 (2004b). 8.1.2, 8.2.1

S. Baker, R. Patil, K. M. Cheung, and I. Matthews. Lucas-Kanade 20 Years On: Part 5. Technical report, Robotics Institute, Carnegie Mellon University (2004b). Technical Report CMU-RI-TR-04-64. 2.4, 8.2.2, 10.1

S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision (IJCV)* (2011). 5.8

Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press (1988). 8.1.1

S. T. Barnard. Stochastic stereo matching over scale. *International Journal of Computer Vision (IJCV)*, 3(1):17–32 (1989). 2.3.1

M. Beljan, R. Klowsky, and M. Goesele. A Multi-View Stereo Implementation on Massively Parallel Hardware. Technical report, Technische Universitat Darmstadt (2011). 7.2

R. Ben-Ari and N. A. Sochen. Variational Stereo Vision with Sharp Discontinuities and Occlusion Handling. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2007). 2.3.2

J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (1992). 8.1.2, 8.2.3

P. Besl and N. McKay. A method for Registration of 3D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256 (1992). 8.3.1

D. N. Bhat and S. K. Nayar. Ordinal Measures for Visual Correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1996). 4.2.2

M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proceedings of the International Conference on Computer Vision (ICCV)* (1993). 4.2.2

G. Blais and M. D. Levine. Registering Multiview Range Data to Create 3D Computer Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8):820–824 (1995). 8.3.1

A. Blake and A. Zisserman. *Visual reconstruction*. MIT Press, Cambridge, MA, USA (1987). 2.3

M. Bleyer, C. Rhemann, and C. Rother. PatchMatch Stereo — Stereo Matching with Slanted Support Windows. In *Proceedings of the British Machine Vision Conference (BMVC)* (2011). 4.2.2

Jules Bloomenthal. An Implicit Surface Polygonizer. In *Graphics Gems IV*, pages 324–349. Academic Press (1994). 6.1.3, 9.1.1

M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe. Parallel Poisson surface reconstruction. In *Proceedings of the International Symposium on Visual Computing* (2009). 6.1.3

M. Bolitho, M. M. Kazhdan, R. C. Burns, and H. Hoppe. Multilevel streaming for out-of-core surface reconstruction. In *Proceedings of the Symposium on Geometry Processing* (2007). 6.1.3

G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371 (1986). 8.4.3

S. Boyd and L. Vandenberghe. *Convex Optimization*. Camebridge University Press (2004). 5.3

Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239 (2001). 2.3.1

K. Bredies, K. Kunisch, and T. Pock. Total Generalized Variation. *SIAM Journal of Imaging Sciences*, 3(3):492–526 (2010). 4.5.2, 5.1

X. Bresson, J. P. Thiran, and S. Osher. Global minimizers of the active contour/snake model. In *In Free Boundary Problems (FBP): Theory and Applications* (2005). 4.5.3

T. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-D Motion Estimation from a Monocular Image Sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26:639–656 (1990). 2.1.1

M. Z. Brown, D. Burschka, and G. D. Hager. Advances in Computational Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(8):993–1008 (2003). 4.1.2

T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2004). 2.3.2

N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Using Multiple Hypotheses to Improve Depth-Maps for Multi-View Stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2008). 7.2

A. Chambolle and T. Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145 (2011). 3.4.2, 3.4.4, 4.5.4, 5.1, 5.3

J. Y. Chang, H. Park, I. K. Park, K. M. Lee, and S. U. Lee. GPU-friendly multi-view stereo reconstruction using surfel representation and graph cuts. *Computer Vision and Image Understanding (CVIU)*, 115(5):620–634 (2011). 7.2

D. Chekhlov, A.P. Gee, A. Calway, and W. Mayol-Cuevas. Ninja on a Plane: Automatic Discovery of Physical Planes for Augmented Reality Using Visual SLAM. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)* (2007). 2.2.1

Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing (IVC)*, 10(3):145–155 (1992). 7.5, 8.3.1

A. Chiuso, P. Favaro, H. Jin, and S. Soatto. "MFm": 3-D Motion From 2-D Motion Causally Integrated Over Time. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2000). 2.1.1

C.-S. Chua and R. Jarvis. Point Signatures: A New Representation for 3D Object Recognition. *International Journal of Computer Vision (IJCV)*, 25(1):63–85 (1997). 8.3

Andrea Cohen, Christopher Zach, Sudipta N. Sinha, and Marc Pollefeys. Discovering and exploiting 3D symmetries in structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012). 10.2.3

R. T. Collins. A Space-Sweep Approach to True Multi-Image Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1996). 2.2.3, 4.2, 7.3.1

A. I. Comport, E. Malis, and P. Rives. Accurate Quadri-focal Tracking for Robust 3D Visual Odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2007). 2.4

A. I. Comport, M. Meilland, and P. Rives. An asymmetric real-time dense visual localisation and mapping system. In *Workshop on Live Dense Reconstruction from Moving Cameras at ICCV* (2011). 2.5, 8.2.6

N. Cornelis and L. J. Van Gool. Real-Time Connectivity Constrained Depth Map Computation Using Programmable Graphics Hardware. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005). 2.2.3

B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH* (1996). 2.2.3, 2.5, 6.1.4, 6.2.1, 6.2.2, 7.3.1, 7.3.2, 10.1

B. L. Curless. *New Methods for Surface Reconstruction from Range Images*. Ph.D. thesis, Stanford University (1997). 6.1.4, 7.3.2

T. A: Davis. *Direct Methods for Sparse Linear Systems*. SIAM (2006). 2.2.2

A. J. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2003). 2.1.1

A. J. Davison, N. D. Molton, I. Reid, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067 (2007). 2.1.2, 2.2.2, 2.2.3

A. J. Davison and D. W. Murray. Mobile Robot Localisation Using Active Vision. In *Proceedings of the European Conference on Computer Vision (ECCV)* (1998). 2.1.1, 8.1.1

P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH* (1997). 3.2.2

P. Debevec, Y. Yu, and G. Boshokov. Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. Technical report, University of California at Berkeley (1998). 6.4.2

P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of SIGGRAPH* (1996). 6.4, 6.4.2, 6.4.2

F. Dellaert and R. Collins. Fast image-based tracking by selective pixel integration. In *Proceedings of the ICCV Workshop on Frame-Rate Vision* (1999). 10.2.3

F. Dellaert and M. Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *International Journal of Robotics Research (IJRR)*, 25:1181–1203 (2006). 1.2, 10.2.1

F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13:14–24 (2001). 3.2.1, 3.2.1

U.R. Dhond and J.K Aggarwal. Structure from stereo - a review. *Proceedings of the International Conference on Systems, Man and Cybernetics, (SMC)*, 19(6):1489–1510 (1989). 4.1.2

Z. Dong, G. Zhang, J. Jia, and H. Bao. Keyframe-based real-time camera tracking. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2009). 8.5.1

H. Durrant-Whyte and T. Bailey. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. *IEEE Robotics and Automation Magazine*, 13(2):99–110 (2006). 2.1

E. Eade and T. Drummond. Edge landmarks in monocular SLAM. In *Proceedings of the British Machine Vision Conference (BMVC)* (2006a). 2.2.1

E. Eade and T. Drummond. Scalable Monocular SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2006b). 2.1.1

E. Eade and T. Drummond. Unified Loop Closing and Recovery for Real Time Monocular SLAM. In *Proceedings of the British Machine Vision Conference (BMVC)* (2008). 2.1.1, 8.5

D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Journal of Machine Vision and Applications*, 9(5–6):272–290 (1997). 8.3.1

A. Elfes and L. Matthies. Sensor integration for robot navigation: combining sonar and range data in a grid-based representation. In *Proceedings of the IEEE Conference on Decision and Control* (1987). 6.1.4

C. Engels, H Stewénius, and D. Nistér. Bundle Adjustment Rules. In *Proceedings of Photogrammetric Computer Vision* (2006). 2.1.2, 2.2.3

A. Ess, A. Neubeck, and L. J. Van Gool. Generalised Linear Pose Estimation. In *Proceedings of the British Machine Vision Conference (BMVC)* (2007). 8.5

O. Faugeras and R. Keriven. Variational principles, surface evolution, PDEs, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344 (1998). 2.3.2

O. Faugeras, T. Viéville, E. Theron, J. Vuillemin, B. Hotz, Z. Zhang, L. Moll, P. Bertin, H. Mathieu, P. Fua, G. Berry, and C. Proy. Real-time correlation-based stereo: algorithm, implementations and applications. Technical Report RR-2013, INRIA (1993). 4.2.2

O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press (1993). 2.1

O. D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1986). 2.1

P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science (2004). 8.4.3, 8.4.3

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Belief Propagation for Early Vision. *International Journal of Computer Vision (IJCV)*, 70(1):41–54 (2006). 2.3.1

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395 (1981). 1.5, 2.1.2, 8.3

A. W. Fitzgibbon. Robust Registration of 2D and 3D Point Sets. In *Proceedings of the British Machine Vision Conference (BMVC)* (2001). 8.4, 8.4.2

Andrew W. Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-Based Rendering Using Image-Based Priors. *International Journal of Computer Vision*, 63(2):141–151 (2005). 5.3

A. Flint, D. W. Murray, and I. Reid. Manhattan Scene Understanding Using Monocular, Stereo, and 3D Features. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2011). 6.1.2

P. Fua. Combining Stereo and Monocular Information to Compute Dense Depth Maps that Preserve Depth Discontinuities. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (1991). 4.2.2

S. Fuhrmann and M. Goesele. Fusion of depth maps with multiple scales. In *SIGGRAPH Asia* (2011). 7.6

Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009). 6.1.2

Y. Furukawa and J. Ponce. Accurate, Dense, and Robust Multi-View Stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2007). 6.1.2, 7.2, 7.4.2

D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2008). 2.2.3, 7.4.2, 7.4.2

D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2007). 2.2.3, 4.2.2, 4.4.1, 5.4

D. Gallup, J.-M. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010a). 6.1.2

D. Gallup, J.-M. Frahm, M. Pollefeys, and E. Zuerich. A Heightmap Model for Efficient 3D Reconstruction from Street-Level Video. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2010b). 6.1.4

N. Gelfand, S. Rusinkiewicz, L. Ikemoto, and M. Levoy. Geometrically Stable Sampling for the ICP Algorithm. In *Proceedings of the IEEE International Workshop on 3D Digital Imaging and Modeling (3DIM)* (2003). 8.3.1

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(6):721–741 (1984). 2.3

S. Geman and D. McClure. Bayesian images analysis: An application to single photon emission tomography. In *Proceedings of Statistical Computing* (1985). 2.3.2

S. F. F. Gibson. Using distance maps for accurate surface representation in sampled volumes. In *Proceedings of the IEEE Symposium on Volume Visualization* (1998). 6.1.4

S. F. Frisken Gibson, R. N. Perry, A. P. Rockwood, and T. R. Jones. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proceedings of SIGGRAPH* (2000). 6.1.4, 6.2.2

M. Goesele, B. Curless, and S. M. Seitz. Multi-View Stereo Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2006). 7.2, 7.3.1, 7.2

M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-View Stereo for Community Photo Collections. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2007). 7.2

M. Gong, R. Yang, L. Wang, and M. Gong. A Performance Study on Different Cost Aggregation Approaches Used in Real-Time Stereo Matching. *International Journal of Computer Vision (IJCV)*, 75(2):283–296 (2007). 4.2.2

G. Graber, T. Pock, and H. Bischof. Online 3D reconstruction using convex optimization. In *Workshop on Live Dense Reconstruction from Moving Cameras at ICCV* (2011). 2.3.3, 4.5.8, 7.1, 7.3.1, 7.3.2, 7.17, 7.5, 7.18, 7.19b, 7.19

L. Gruber, S. Gauglitz, J. Ventura, S. Zollmann, M. Huber, M. Schlegel, G. Klinker, D. Schmalstieg, and T. Höllerer. The City of Sights: Design, construction, and measurement of an Augmented Reality stage set. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)* (2010). 4.22, 7.5, 7.18, 7.19, 8.2.6

A. Gruen and D. Akca. Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):151–174 (2005). 8.3

A. W. Gruen. Adaptive Least Squares Correlation: A Powerful Image Matching Technique. *South African Journal of Photogrammetry, Remote Sensing, and Cartography*, 14 (1985). 8.3

M. Habbecke and L. Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2007). 6.1.2, 7.2, 7.4.2

A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison. Applications of the Legendre-Fenchel transformation to computer vision problems. Technical Report DTR11-7, Imperial College London (2011). 3.4.4

A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison. Real-Time Camera Tracking: When is High Frame-Rate Best? In *Proceedings of the European Conference on Computer Vision (ECCV)* (2012). 3.2.2, 8.1.1

M. J. Hannah. *Computer matching of areas in stereo images*. Ph.D. thesis, Stanford University, Stanford, CA, USA (1974). 4.2.2

C. Harris and C. Stennett. RAPID: A video rate object tracker. In *Proceedings of the British Machine Vision Conference (BMVC)* (1990). 8.1.1

C. G. Harris and J. M. Pike. 3D Positional Integration from Image Sequences. In *Proceedings of the Alvey Vision Conference*, pages 233–236 (1987). 2.1.1

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition (2004). 1.2

K. He, J. Sun, and X. Tang. Guided Image Filtering. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2010). 4.2.2

B. Heigl, J. Denzler, and H. Niemann. Combining computer graphics and computer vision for probabilistic visual robot navigation. In *Proceedings of SPIE Enhanced and Synthetic Vision* (2000). 8.2.6

B. Heigl, R. Koch, M. Pollefeys, J. Denzler, and L. J. Van Gool. Plenoptic Modeling and Rendering from Image Sequences Taken by Hand-Held Camera. In *Proceedings of the DAGM Symposium on Pattern Recognition* (1999). 8.2.6

P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. In *Proceedings of the International Symposium on Experimental Robotics (ISER)* (2010). 2.5

C. Hernández and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding (CVIU)*, 96(3):367–392 (2004). 7.2, 7.3.1, 7.4.2, 7.17

C. Hernández, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2007). 7.2

A. Hilton. Scene modelling from sparse 3D data. *Image and Vision Computing*, 23(10):900–920 (2005). 2.2.2, 7.2

A. Hilton, A.J. Stoddart, J. Illingworth, and T. Windeatt. Reliable Surface Reconstruction From Multiple Range Images. *Proceedings of the European Conference on Computer Vision (ECCV)* (1996). 6.1.1, 6.1.4, 7.3.1

H. Hirschmüller. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005). 2.3.1, 4.3.2

C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, and G. Reitmayr. Online Feedback for Structure-from-Motion Image Acquisition. In *Proceedings of the British Machine Vision Conference (BMVC)* (2012). 7.4.2

H. Hoppe, T. DeRose, T. Duchamp, J. A. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of SIGGRAPH* (1992). 6.1.3

B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203 (1981). 2.3.2

B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642 (1987). ISSN 1084-7529. 8.3

B.K.P. Horn and J. G. Harris. Rigid body motion from range image sequences. *CVGIP: Image Understanding*, 53(1):1–13 (1991). 8.3

A. Hornung, B. Zeng, and L. Kobbelt. Image selection for improved Multi-View Stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2008). 7.4.2

A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)* (2009). 4.2.2

A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, and C. Rother. Real-time local stereo matching using guided image filtering. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)* (2011). 4.2.2

D. F. Huber and M. Hebert. Fully automatic registration of multiple 3D data sets. *Image and Vision Computing (IVC)*, 21(7):637–650 (2003). 8.3

P. J. Huber. *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley-Interscience (1981). 3.4.1

M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze. A fast stereo matching algorithm suitable for embedded real-time systems. *Computer Vision and Image Understanding (CVIU)*, 114(11):1180–1202 (2010). 5.4

M. Irani and P. Anandan. Robust Multi-Sensor Image Alignment. In *Proceedings of the International Conference on Computer Vision (ICCV)* (1998). 8.2.5

M. Irani and P. Anandan. All About Direct Methods. In *Proceedings of the International Workshop on Vision Algorithms, in association with ICCV* (1999). 8.1.2, 10

M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision (IJCV)*, 12(1):5–16 (1994). 8.1.2

M. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *Proceedings of the European Conference on Computer Vision (ECCV)* (1996). 8.2.6

S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. A. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. J. Davison, and A. Fitzgibbon. KinectFusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proceedings of ACM Symposium on User Interface Software and Technolog (UIST)* (2011). 1.5, 9.3.3, 9.5

T. Jebara, A. Azarbayejani, and A. Pentland. 3D Structure from 2D Motion. *IEEE Signal Processing Magazine*, 16:66–84 (1999). 2.1

H. Jin, P. Favaro, and S. Soatto. Real-time 3-d motion and structure of point features: Front-end system for vision-based control and interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2000). 2.1.1

A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(1):433–449 (1999). 8.3

B. R. Jones, R. Sodhi, R. H. Campbell, G. Garnett, and B. P. Bailey. Build your world and play in it: Interacting with surface particles on complex objects. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)* (2010). 8.3

M. W. Jones, J. A. Bærentzen, and M. Srámek. 3D Distance Fields: A Survey of Techniques and Applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):581–599 (2006). 8.4.3

T. Kanade and M. Okutomi. A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 16(9):920–932 (1994). 4.2.2

S. B. Kang, R. Szeliski, and J. Chai. Handling Occlusions in Dense Multi-view Stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2001). 2.2.3, 4.2.2, 4.4.3, 7.3.1

M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Eurographics Symposium on Geometry Processing* (2006). 5.3.1, 6.1.3, 9.1.1

H. Kim and K. Sohn. Hierarchical disparity estimation with energy-based regularization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)* (2003). 2.3.2

S. J. Kim, D. Gallup, J. -m. Frahm, A. Akbarzadeh, Q. Yang, R. Yang, D. Nistér, and M. Pollefeys. Gain adaptive real-time stereo streaming. In *Proceedings of the International Conference on Vision Systems* (2007). 2.2.3

G. Klein and D. W. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)* (2007). 2.1.2, 2.2.2, 2.2.3, 5.5, 6.4.2, 7.3.2, 9.1, 9.1.1, 9.1.2, 9.2.1

G. Klein and D. W. Murray. Improving the Agility of Keyframe-based SLAM. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2008). 2.1.2, 2.2.1, 2.1a, 2.4, 4.5.8, 7.4.1, 7.5, 8.2.6, 1, 8.5, 8.5.1, 10.1

M. Klodt, T. Schoenemann, K. Kolev, M. Schikora, and D. Cremers. An Experimental Comparison of Discrete and Continuous Shape Optimization Methods. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2008). 2.3.1

L. Kobbelt and M. Botsch. A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(6):801–814 (2004). 6.1.2

R. Koch, M. Pollefeys, and L. J. Van Gool. Multi Viewpoint Stereo from Uncalibrated Video Sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)* (1998). 4.3.2, 7.2, 7.3.1

K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous Global Optimization in Multiview 3D Reconstruction. *International Journal of Computer Vision (IJCV)*, 84(1):80–96 (2009). 7.2

S. Kosov, T. Thormählen, and H.-P. Seidel. Accurate Real-Time Disparity Estimation with Variational Methods. In *Proceedings of the International Symposium on Visual Computing (ISVC)* (2009). 2.3.2

K. N. Kutulakos and S. M. Seitz. A Theory of Shape by Space Carving. *International Journal of Computer Vision (IJCV)*, 38(3):199–218 (2000). 7.2

P. Labatut, R. Keriven, and J.-P. Pons. A GPU Implementation of Level Set Multiview Stereo. In *Proceedings of the International Conference on Computational Science* (2006a). 7.2

P. Labatut, R. Keriven, and J.-P. Pons. Fast Level Set Multi-View Stereo on Graphics Hardware. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2006b). 7.2

P. Labatut, J.-P. Pons, and R. Keriven. Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2007). 6.1.2

Sang Wook Lee and Ruzena Bajcsy. Detection of Specularity Using Color and Multiple Views. In **Giulio Sandini**, editor, *ECCV*, volume 588 of *Lecture Notes in Computer Science*, pages 99–114. Springer (1992). ISBN 3-540-55426-2. 10.1

S. Leopoldseder, H. Pottmann, and H. Zhao. The d^2Tree: A Hierarchical Representation of the Squared Distance Function. Technical report, Vienna University of Technology (2003). 8.4

V. Lepetit and P. Fua. Monocular Model-Based 3D Tracking of Rigid Objects: A Survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89 (2005). 8.1.1

V. Lepetit and P. Fua. Keypoint Recognition using Randomized Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(9):1465–1479 (2006). 8.1.1

V. Lepetit, P. Lagger, and P. Fua. Randomized Trees for Real-Time Keypoint Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005). 8.1.1

D. Levin. Mesh-Independent Surface Interpolation. *Advances in Computational Mathematics* (1999). 6.1.2

M. Levoy and P. Hanrahan. Light Field Rendering. In *Proceedings of SIGGRAPH* (1996). 6.4

M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. E. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital Michelangelo Project: 3D scanning of large statues. In *Proceedings of SIGGRAPH* (2000). 8.3

M. Lhuillier and L. Quan. A Quasi-Dense Approach to Surface Reconstruction from Uncalibrated Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(3):418–433 (2005). 6.1.2

Stephen Lin, Yuanzhen Li, Sing Bing Kang, Xin Tong, and Heung-Yeung Shum. Diffuse-Specular Separation and Depth Recovery from Image Sequences. In **Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen**, editors, *ECCV (3)*, volume 2352 of *Lecture Notes in Computer Science*, pages 210–224. Springer (2002). 10.1

Y. Liu, X. Cao, Q. Dai, and W. Xu. Continuous depth estimation for multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009). 2.3.2

W. E. Lorensen and H. E. Cline. Marching Cubes: A high resolution 3D surface construction algorithm. In *Proceedings of SIGGRAPH* (1987). 2.2.3, 6.3.2, 7.3.1, 9.1.1

S. J. Lovegrove. *Parametric Dense Visual SLAM*. Ph.D. thesis, Imperial College London (2011). 2.2.2, 2.4

S. J. Lovegrove and A. J. Davison. Real-Time Spherical Mosaicing using Whole Image Alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2010). 8.2.1, 9.2.1

S. J. Lovegrove, A. J. Davison, and J. Ibanez-Guzmán. Accurate Visual Odometry from a Rear Parking Camera. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)* (2011). 8.2.1

D. Lovi, N. Birkbeck, D. Cobzas, and M. Jagersand. Incremental Free-Space Carving for Real-Time 3D Reconstruction. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2010). 2.2.2, 7.3

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110 (2004). 1.3, 2, 1.5

B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (1981). 2.4, 3.3, 3.3.1, 4.2.2, 8.1.2, 8.2.3, 8.2.4, 8.2.5

Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag (2003). 8.2

E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2004). 2.4, 8.2.1

D. Marr. *Vision*. MIT Press, Cambridge MA (1982). 2.3

L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision (IJCV)*, 3(3):209–238 (1989). 2.2.3, 4.4.2

P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nistér, and M. Pollefeys. Real-Time Visibility-Based Fusion of Depth Maps. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2007). 2.2.3, 4.3.2, 6.1.2, 7.3, 7.17

N. J. Mitra, N. Gelfand, H. Pottmann, and L. J. Guibas. Registration of Point Cloud Data from a Geometric Optimization Perspective. In *Proceedings of the Symposium on Geometry Processing* (2004). 8.4, 8.4.4

N. D. Molton, A. J. Davison, and I. Reid. Locally Planar Patch Features for Real-Time Structure from Motion. In *Proceedings of the British Machine Vision Conference (BMVC)* (2004). 2.2.1

J. M. M. Montiel, J. Civera, and A. J. Davison. Unified Inverse Depth Parametrization for Monocular SLAM. In *Proceedings of Robotics: Science and Systems (RSS)* (2006). 2.1.1

H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *A.I. Magazine*, 9(2):61–74 (1988). 6.1.4

E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real-Time Localization and 3D Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2006). 2.1.2

D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685 (1989). 2.3.2, 4.3.1

H.-H. Nagel and W. Enkelmann. An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(5):565–593 (1986). 2.3.2, 4.5.3, 5.1

P. J. Narayanan, P. Rander, and T. Kanade. Constructing Virtual Worlds Using Dense Stereo. In *Proceedings of the International Conference on Computer Vision (ICCV)* (1998). 7.2, 7.3.1

P. J. Neugebauer. Geometrical cloning of 3D objects via simultaneous registration of multiple range images. In *Proceedings of the 1997 International Conference on Shape Modeling and Applications* (1997). 8.3.1

R. A. Newcombe and A. J. Davison. Live Dense Reconstruction with a Single Moving Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010). 1.5, 2.1b, 2.3.3, 9, 9.5

R. A. Newcombe, A. J. Davison, and G. Vogiatzis. Workshop on Live Dense Reconstruction from Moving Cameras at ICCV (2011a). 7.3.2, 7.5, 7.18

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)* (2011b). 1.5, 2.5, 9, 9.5

R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: Dense Tracking and Mapping in Real-Time. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2011c). 1.5, 2.4, 9, 9.5

G. M. Nielson and B. Hamann. The asymptotic decider: resolving the ambiguity in marching cubes. In *Proceedings of the Conference on Visualization* (1991). 6.3.2

D. Nistér, O. Naroditsky, and J. Bergen. Visual Odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2004). 2.1.2

D. Nistér, O. Naroditsky, and J. Bergen. Visual Odometry for Ground Vehicle Applications. *Journal of Field Robotics*, 23(1):– (2006). 2.2.3

D. Nister and H. Stewenius. Scalable Recognition with a Vocabulary Tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2006). 8.1.1

Nvidia. *Compute Unified Device Architecture-Programming Guide Version 2.0*. NVIDIA Corporation (2008). 3.5, 10.1

Y. Ohtake, A. Belyaev, and H.-P. Seidel. A Multi-scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions. In *Proceedings of Shape Modeling International* (2003). 6.1.3, 9.1.1

M. Okutomi and T. Kanade. A Multiple-Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(4):353–363 (1993). 7.3.1

S.J. Osher and R.P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer (2002). 6.1.3, 6.1.4

G. P. Otto and T. K. W. Chau. Region-growing algorithm for matching of terrain images. *Image and Vision Computing (IVC)*, 7(2):83–94 (1989). 7.2

Q. Pan, G. Reitmayr, and T. Drummond. ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition. In *Proceedings of the British Machine Vision Conference (BMVC)* (2009). 2.2.2, 7.2, 7.3

S. Parker, P. Shirley, Y. Livnat, C. Hansen, and P. Sloan. Interactive Ray Tracing for Isosurface Rendering. In *Proceedings of Visualization* (1998). 6.1.4, 6.3.1, 6.3.1

Mark Pauly, Niloy J. Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J. Guibas. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics*, 27(3):43:1–43:11 (2008). 10.2.3

H. Pfister, M. Zwicker, J. van Baar, and M. H. Gross. Surfels: surface elements as rendering primitives. In *Proceedings of SIGGRAPH* (2000). 2.5, 6.1.2

Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2004). 6.4

T. Pock. *Fast Total Variation for Computer Vision*. Ph.D. thesis, Graz University of Technology (2008). 2.3.1, 2.3.3, 4.5.1, 5.1, 9.1.1, 10.1

T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2011). 5.1, 7.3.2

T. Pock, M. Grabner, and H. Bischof. Real-time Computation of Variational Methods on Graphics Hardware. In *Proceedings of the Computer Vision Winter Workshop* (2007a). 2.3.3

T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A Convex Formulation of Continuous Multi-Label Problems. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2008a). 2.3.1

T. Pock, M. Unger, D. Cremers, and H. Bischof. Fast and exact solution of Total Variation models on the GPU. In *Proceedings of the CVPR Workshop on Visual Computer Vision on GPU's* (2008b). 4.3.2

T. Pock, C. Zach, and H. Bischof. Mumford-Shah Meets Stereo: Integration of Weak Depth Hypotheses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2007b). 4.3.1

T. Pock, L. Zebedin, and H. Bischof. TGV-Fusion. In *Rainbow of Computer Science* (2011). 4.3.2, 4.5.2, 4.5.7, 4.5.7, 5.2.3, 5.2.4

T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(26):314–319 (1985). 2.3

M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual Modeling with a Hand-Held Camera. *International Journal of Computer Vision (IJCV)*, 59:207–232 (2004). 2.2.3, 7.2, 7.3.1

M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Hand-held acquisition of 3D models with a video camera. In *Proceedings of the IEEE International Workshop on 3D Digital Imaging and Modeling (3DIM)* (1999). 1.1, 2.2.3, 5.4

M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed Real-Time Urban 3D Reconstruction from Video. *International Journal of Computer Vision (IJCV)*, 78(2-3):143–167 (2008). 2.2.3, 7.3

J.-P. Pons, R. Keriven, and O. D. Faugeras. Modelling Dynamic Scenes by Registering Multi-View Image Sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005). 7.2

D. Porquet, J.-M. Dischler, and D. Ghazanfarpour. Real-time high-quality View-Dependent Texture Mapping using per-pixel visibility. In *GRAPHITE* (2005). 6.4.2

H. Pottmann, S. Leopoldseder, and M. Hofer. Registration without ICP. *Computer Vision and Image Understanding (CVIU)*, 95(1):54–71 (2004). 8.4

M. Pupilli and A. Calway. Real-time Visual SLAM with Resilience to Erratic Motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2006). 2.1.1

R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the limits of stereo using variational stereo estimation. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)* (2012). 5.1

G. Reitmayr and T. W. Drummond. Going out: Robust modelbased tracking for outdoor augmented reality. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)* (2006). 2.1.2

L. Robert and R. Deriche. Dense Depth Map Reconstruction: A Minimization and Regularization Approach which Preserves Discontinuities. In *Proceedings of the European Conference on Computer Vision (ECCV)* (1996). 2.3.2

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2006). 1.3

H. Roth and M. Vona. Moving Volume KinectFusion. In *Proceedings of the British Machine Vision Conference (BMVC)* (2012). 9.3.4

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268 (1992). 2.3.2, 3.4.2, 3.4.2, 4.5.1

S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-Time 3D Model Acquisition. In *Proceedings of SIGGRAPH* (2002). 2.5, 8.3.1

S. Rusinkiewicz and M. Levoy. QSplat: a multiresolution point rendering system for large meshes. In *Proceedings of SIGGRAPH* (2000). 2.5, 6.1.2

S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In *Proceedings of the IEEE International Workshop on 3D Digital Imaging and Modeling (3DIM)* (2001). 8.3.1, 8.3.1

G. Sansoni, M. Trebeschi, and F. Docchio. State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation. *Sensors*, 9(1):568–601 (2009). 8.3

D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision (IJCV)*, 47:7–42 (2001). 2.3.3, 4.1, 4.1.2, 4.2.2, 4.2.2, 4.3.1, 4.5.8

C. Schroers, H. Zimmer, L. Valgaerts, A. Bruhn, O. Demetz, and J. Weickert. Anisotropic Range Image Integration. In *Proceedings of the DAGM Symposium on Pattern Recognition* (2012). 7.3.2

W. R. Scott, G. Roth, and J.-F. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys*, 35(1):64–96 (2003). 7.4.2

A. Segal, D. Hähnel, and S. Thrun. Generalized ICP. In *Proceedings of Robotics: Science and Systems (RSS)* (2009). 8.3.1

S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2006). 2.2.3, 2.3.3, 4.5.8, 7.2, 7.2.1, 7.1, 7.3.1, 7.5

S. M. Seitz and C. R. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. *International Journal of Computer Vision (IJCV)*, 35(2):151–173 (1999). 7.2

J. Shah. A nonlinear diffusion model for discontinuous disparity and half-occlusions in stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1993). 2.3.2

J. Shi and C. Tomasi. Good Features to Track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1994). 1.4.1

G. Silveira, E. Malis, and P. Rives. An Efficient Direct Approach to Visual SLAM. *IEEE Transactions on Robotics (T-RO)*, 24(5):969–979 (2008). 2.4

S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2009). 6.1.2

J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2003). 8.1.1

N. Slesareva, A. Bruhn, and J. Weickert. Optic Flow Goes Stereo: A Variational Method for Estimating Discontinuity-Preserving Dense Disparity Maps. In *Proceedings of the DAGM Symposium on Pattern Recognition* (2005). 2.3.2

N. Slesareva, T. Bühler, K. U. Hagenburg, J. Weickert, A. Bruhn, Z. Karni, and H.-P. Seidel. Robust Variational Reconstruction from Multiple Views. In *Proceedings of the Scandinavian Conference on Image Analysis (SCIA)* (2007). 2.3.2

P. Smith, I. Reid, and A. J. Davison. Real-Time Single-Camera SLAM with Straight Lines. In *Proceedings of the British Machine Vision Conference (BMVC)* (2006). 2.2.1

S. Soatto. 3-D structure from visual motion : modeling, representation and observability. *Automatica*, 33:1287–1312 (1997). 2.1.1

S. Stegmaier, M. Strengert, T. Klein, and T. Ertl. A simple and flexible volume rendering framework for graphics-hardware-based raycasting. In *Proceedings of the International Workshop on Volume Graphics* (2005). 7.3.1

F. Steinbrucker, T. Pock, and D. Cremers. Large Displacement Optical Flow Computation without Warping. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2009). 5.3, 5.3, 5.3

F. Steinbrucker, J. Sturm, and D. Cremers. Real-Time Visual Odometry from Dense RGB-D Images. In *Workshop on Live Dense Reconstruction from Moving Cameras at ICCV* (2011). 2.5

H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige. Double Window Optimisation for Constant Time Visual SLAM. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2011). 2.5

H. Strasdat, J. M. M. Montiel, and A. J. Davison. Real-Time Monocular SLAM: Why Filter? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2010). 1.2, 2.1.2

C. Strecha, T. Tuytelaars, and L. J. Van Gool. Dense Matching of Multiple Wide-baseline Views. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2003). 2.3.2

C Strecha, W von Hansen, L. J. Van Gool, P Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2008). 4.1, 4.2, 4.4.3, 4.5.6, 4.5.8, 5.3.1, 7.2.1, 7.5

J. Stuehmer, S. Gumhold, and D. Cremers. Real-Time Dense Geometry from a Handheld Camera. In *Proceedings of the DAGM Symposium on Pattern Recognition* (2010). 2.3.3, 5.1, 5.1, 5.2.4, 5.3, 5.4, 9.1.1

J. Sun, N. Zheng, and H.-Y. Shum. Stereo Matching Using Belief Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(7):787–800 (2003). 2.3.1

R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision (IJCV)*, 5(3):271–301 (1991). 2.3

R. Szeliski. Rapid Octree Construction from Image Sequences. *Computer Vision, Graphics, and Image Processing*, 58(1):23–32 (1993). 6.1.4

R. Szeliski. Image Alignment and Stitching: A Tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104 (2006). 8.1.2

R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag, New York, NY, USA (2010). 4.2.2, 6.1.2, 6.4

R. Szeliski and S. B. Kang. Recovering 3D Shape and Motion from Image Streams Using Non-Linear Least Squares. Technical report, Robotics Institute (1993). 9.2.1

R. Szeliski and D. Scharstein. Sampling the disparity space image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26:419–425 (2004). 2.2.3, 7.2.1

R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. In *Proceedings of SIGGRAPH* (1992). 6.1.2

R. Szeliski and R. Zabih. An Experimental Comparison of Stereo Algorithms. In *Workshop on Vision Algorithms* (1999). 4.2.2

R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(6):1068–1080 (2008). 2.3.1

H. Tao, H. S. Sawhney, and R. Kumar. A Global Matching Framework for Stereo Computation. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2001). 4.2.2

M. F. Tappen and W. T. Freeman. Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2003). 2.3.1

Camillo J. Taylor and Arvind Bhusnurmath. Solving Image Registration Problems Using Interior Point Methods. In **David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman**, editors, *ECCV (4)*, volume 5305 of *Lecture Notes in Computer Science*, pages 638–651. Springer (2008). ISBN 978-3-540-88692-1. 5.3

S. Thrun and A. Bücken. Integrating Grid-Based and Topological Maps for Mobile Robot Navigation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (1996). 6.1.4

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Cambridge: MIT Press (2005). 1.5, 2.1, 6.1.4, 8.3, 10.2.1

C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *Proceedings of the International Conference on Computer Vision (ICCV)* (1998). 4.3.1, 9.3.1

P. H. S. Torr and A. Zisserman. Feature Based Methods for Structure and Motion Estimation. In *Proceedings of the International Workshop on Vision Algorithms, in association with ICCV* (1999). 8.1.1, 10

Hugo Touchette. Legendre-Fenchel transforms in a nutshell. Technical report, School of Mathematical Sciences, Queen Mary, University of London, (2005). 3.3, 3.4.3

B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle Adjustment — A Modern Synthesis. In *Proceedings of the International Workshop on Vision Algorithms, in association with ICCV* (1999). 10

J. Tukey. A survey of sampling from contaminated distributions. *Contributions to Probability and Statistics.* (1960). 8.2.4

G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of SIGGRAPH* (1994). 6.1.2, 8.3.1

G. Turk and J. F. OBrien. Variational Implicit Surfaces. Technical Report GIT-GVU-99-15, Georgia Institute of Technology (1999). 6.1.3, 9.1.1

T. Tykkala and A. I. Comport. A dense structure model for image based stereo SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2011). 2.4

M. Unger, T. Pock, and H. Bischof. Continuous Globally Optimal Image Segmentation with Local Constraints. In *Proceedings of the Computer Vision Winter Workshop* (2008). 2.3.3

O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. Ph.D. thesis, Cornell University (1999). 2.3.1

C. R. Vogel and M. E. Oman. Iterative Methods For Total Variation Denoising. *SIAM Journal of Scientific Computing*, 17:227–238 (1996). 3.4.2

G. Vogiatzis and C. Hernández. Video-based, real-time multi-view stereo. *Image and Vision Computing (IVC)*, 29(7):434–441 (2011). 2.2.3, 7.3

G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-View Stereo via Volumetric Graph-Cuts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005). 7.2

H.-H. Vu, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009). 7.2

A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An Improved Algorithm for TV-L1 Optical Flow. In *Proceedings of the Dagstuhl Seminar on Statistical and Geometrical Approaches to Visual Motion Analysis* (2009). 8.2.5, 9.1.1

J. Weickert, S. Ishikawa, and A. Imiya. Linear Scale-Space has First been Proposed in Japan. *Journal of Mathematical Imaging and Vision*, 10(3):237–252 (1999). 2.3.2

A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof. Dense reconstruction on-the-fly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012). 7.3.2, 7.5

H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396 (1995). 6.1.3

M. Werlberger. *Convex Approaches for High Performance Video Processing*. Ph.D. thesis, Graz University of Technology (2012). 4.5.3

M.D. Wheeler, Y. Sato, and K. Ikeuchi. Consensus surfaces for modeling 3D objects from multiple range images. In *Proceedings of the International Conference on Computer Vision (ICCV)* (1998). 7.3.1

T. Whelan, H. Johannsson, M. Kaess, J.J. Leonard, and J.B. McDonald. Robust Tracking for Real-Time Dense RGB-D Mapping with Kintinuous. Technical Report MIT-CSAIL-TR-2012-031, Computer Science and Artificial Intelligence Laboratory, MIT (2012a). 9.3.4

T. Whelan, J. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard. Kintinuous: Spatially Extended KinectFusion. In *Workshop on RGB-D: Advanced Reasoning with Depth Cameras, in conjunction with Robotics: Science and Systems* (2012b). 9.3.4

B. Williams, G. Klein, and I. Reid. Real-Time SLAM Relocalisation. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2007). 2.1.2

O. Williams and A. W. Fitzgibbon. Gaussian process implicit surfaces . In *Gaussian Processes in Practice* (2007). 6.1.3

J. Woetzel and R. Koch. Real-time multi-stereo depth estimation on GPU with approximative discontinuity handling. In *Proceedings of the Conference on Visual Media Production (CVMP)* (2004). 2.2.3

D. Wood, D. Azuma, W. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. Surface Light Fields for 3D Photography. In *Proceedings of SIGGRAPH* (2000). 10.2.2

O. J. Woodford, F. Perbet, M. Pham, C. Hernandez, G. Vogiatzis, A. Maki, B. Stenger, and R. Cipolla. Live 3D Shape Reconstruction, Recognition and Registration. In *Workshop on Live Dense Reconstruction from Moving Cameras at ICCV* (2011). 2.2.3, 7.3, 7.18

K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems. In *Proceedings of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation* (2010). 6.1.4

Y. Xu and A. K. Roy-Chowdhury. Integrating Motion, Illumination, and Structure in Video Sequences with Applications in Illumination-Invariant Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(5):793–806 (2007). 8.2.5

Y. Xu and A. K. Roy-Chowdhury. Inverse Compositional Estimation of 3D Pose And Lighting in Dynamic Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(7):1300–1307 (2008). 8.2.5

Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-Depth Super Resolution for Range Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2007). 4.3.1

R. Yang, G. Welch, and G. Bishop. Real-Time Consensus-Based Scene Reconstruction Using Commodity Graphics Hardware. *Computer Graphics Forum*, 22(2):207–216 (2003). 2.2.3

K.-J. Yoon and I.S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* (2006). 4.2.2

R. Zabih and J. Woodfill. Non-parametric Local Transforms for Computing Visual Correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV)* (1994). 4.2.2

C. Zach. Fast and High Quality Fusion of Depth Maps. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2008). 2.3.3, 6.1.4, 7.2, 7.3, 7.3.2, 7.17

C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast Global Labeling for Real-Time Stereo Using Multiple Plane Sweeps. In *Vision, Modeling and Visualization* (2008). 2.3.3, 7.3.1

C. Zach, K. Karner, B. Reitinger, and H. Bischof. Space carving on 3D graphics hardware. Technical report, Technical University of Graz (2004). 7.2

C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Proceedings of the DAGM Symposium on Pattern Recognition* (2007a). 2.3.3, 2.3.3, 5.1, 7.3.2, 9.1.1

C. Zach, T. Pock, and H. Bischof. A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2007b). 2.3.3, 6.1.4, 7.2, 7.3.1, 7.3.2, 7.3.2

C. Zach, M. Sormann, and K. F. Karner. High-Performance Multi-View Reconstruction. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2006). 2.2.3, 2.3.3, 5.4, 7.2, 7.3.1, 7.3.1

M. Zeng, F. Zhao, J. Zheng, and X. Liu. A Memory-Efficient KinectFusion Using Octree. In *Computational Visual Media*, volume 7633, pages 234–241 (2012). 9.3.4, 10.2.3

D. Zhang and M. Hebert. Harmonic Maps and Their Applications in Surface Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1999). 8.3

Y. Zhang, M. Gong, and Y.-H. Yang. Local stereo matching with 3D adaptive cost aggregation for slanted surface modeling and sub-pixel accuracy. In *Proceedings of the International Conference on Pattern Recognition (ICPR)* (2008). 4.2.2

Z. Zhang. On Local Matching of Free-Form Curves. In *BMVC92*, pages 345–356 (1992). 8.3.1

Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision (IJCV)*, 13(2):119–152 (1994). 8.3.1

K. Zhou, M. Gong, X. Huang, and B Guo. Data-Parallel Octrees for Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics (VGC)*, 17(5) (2011). 6.1.3

M. Zhu. *Fast numerical algorithms for total variation based image restoration*. Ph.D. thesis, University of California at Los Angeles (2008). 3.4.4

H. Zimmer, A. Bruhn, L. Valgaerts, M. Breuß, J. Weickert, B. Rosenhahn, and H.-P. Seidel. PDE-Based Anisotropic Disparity-Driven Stereo Vision. In *Vision, Modeling and Visualization* (2008). 2.3.2