Live Dense Reconstruction with a Single Moving Camera

Richard A. Newcombe and Andrew J. Davison
Department of Computing
Imperial College London
London, UK
[rnewcomb,ajd]@doc.ic.ac.uk

Abstract

We present a method which enables rapid and dense reconstruction of scenes browsed by a single live camera. We take point-based real-time structure from motion (SFM) as our starting point, generating accurate 3D camera pose estimates and a sparse point cloud. Our main novel contribution is to use an approximate but smooth base mesh generated from the SFM to predict the view at a bundle of poses around automatically selected reference frames spanning the scene, and then warp the base mesh into highly accurate depth maps based on view-predictive optical flow and a constrained scene flow update. The quality of the resulting depth maps means that a convincing global scene model can be obtained simply by placing them side by side and removing overlapping regions. We show that a cluttered indoor environment can be reconstructed from a live hand-held camera in a few seconds, with all processing performed by current desktop hardware. Real-time monocular dense reconstruction opens up many application areas, and we demonstrate both real-time novel view synthesis and advanced augmented reality where augmentations interact physically with the 3D scene and are correctly clipped by occlusions.

1. Introduction

Real-time monocular camera tracking (alternatively known as real-time SFM or monocular SLAM) has recently seen great progress (e.g. [3, 5]). However, by design such systems build only sparse feature-based scene models, optimised to permit accurate and efficient live camera pose estimation. Their application to problems where more is needed from a vision system than pure camera positioning has therefore been limited.

In many of the main potential application domains of real-time monocular SLAM, it is desirable for the same video input to be used both to estimate ego-motion and to construct a detailed environment model. In mobile robotics, a vision system should ideally enable not just raw localisation but general spatial awareness and scene understanding. In augmented reality, knowledge of camera positioning alone permits the stable addition of virtual 3D objects to video but not the occlusion handling and object/real scene interaction that a scene model allows.

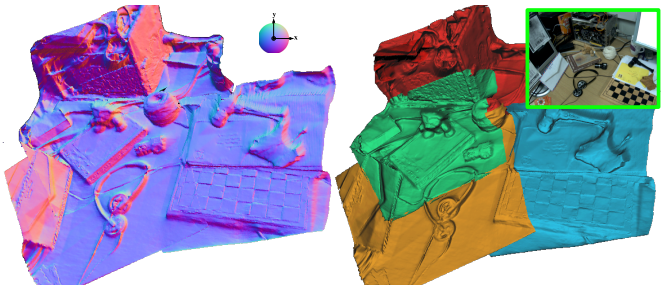

Figure 1. Partial reconstruction generated live from a single moving camera. We render the reconstructed surface's normal map (left) and how this surface is built from four local reconstructions (right). The insert shows an overview of the reconstructed scene.

The dense stereo literature in computer vision has long aimed at the reconstruction of continuous surfaces from sets of registered images, and there are now many examples of impressive results [11]. However, this work has previously been unavailable to real-time applications due to computational cost and other restrictions.

Attempts at real-time monocular tracking and dense modelling to date have had limited results. Pan *et al.* [8] recently presented a system for live modelling of simple objects by tetrahedralisation of an SFM point cloud and visibility reasoning, but in this case high accuracy is only possible for multi-planar scenes.

The work closest in spirit to ours is that of Pollefeys *et al.* [10], who constructed an advanced, real-time system targeted at the dense modelling of urban scenes from a multi-camera rig mounted on a vehicle, aided if necessary by GPS and IMU data. Real-time SFM, plane sweep stereo and

depth map fusion [6] are combined to reconstruct large urban models.

In the approach we propose, we take a number of state-of-the-art components from computer vision and graphics and combine them in a significantly novel fashion to solve real-time monocular dense reconstruction of cluttered natural scenes. Real-time SFM is first used to estimate live camera pose and provide a 3D feature point cloud to which we fit an initial base surface approximating the real scene.

The key to dense reconstruction is precise dense correspondence between images offering sufficient baseline for triangulation, and the main novelty of our approach is the manner in which this is enabled. In a similar manner to [16] we utilise a coarse base surface model as the initial starting point for dense reconstruction. Our base surface permits view predictions to be made between the cameras in a local bundle, and these predictions are then compared with the true images using GPU-based variational optical flow. This permits dense correspondence fields of sub-pixel accuracy to be obtained, even in areas of very low image texture, and this correspondence information is then used to update the base surface prior into highly accurate local depth maps using constrained scene flow optimisation. Depth map creation is pipelined, and multiple depth maps are straightforwardly fused to create complete scene reconstructions.

In the next section we summarise the components of our method before explaining each in detail, discussing previous work related to each part in context. Our results are illustrated in Section 3 (and via the videos available online), where we highlight potential uses for advanced interactive augmented reality and live view synthesis.

2. Method

2.1. Overview

We illustrate the overlapping processes which make up our reconstruction pipeline in Figure 2. In the primary process, online structure from motion computation provides a real-time estimate of the camera's current pose P_t together with representation of the surveyed scene comprising a set of points $\mathbf{x}_p$ (Figure 2(a)). 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 (2(b)).

Another process make uses of the current base model and selects bundles of cameras which have partially overlapping surface visibility, (Figure 2(c)), each comprising a single reference P^{ref} and several neighbouring comparison frames that will be used in the high quality reconstruction process. Each camera bundle is fed to the dense reconstruction process which produces a high quality dense depth estimate for every pixel in the reference view $\mathbf{D}(u, v)$, obtained by deforming a dense sampling of the initial base meshes vertices using a constrained scene flow update (Figure 2(d)). Finally, each dense depth map is then triangulated in the global frame, $\mathbf{S}_G$, and integrated into the global surface model (Figure 2(e)). Due to the high quality of the depth maps computed, where local surface reconstructions overlap the redundant vertices can simply be trimmed to produce a continuous surface.

We now describe each of these components in detail.

2.2. Structure from Motion

Common to all dense reconstruction systems is the requirement to obtain high quality camera pose estimation in a global frame. We use real-time Structure from Motion to furnish a live camera pose estimate and a sparse point cloud which is the starting point for dense scene reconstruction.

Davison [3] introduced a single camera tracking and mapping system which used sequential probabilistic filtering to build a consistent scene map and live camera pose estimate, but the absolute accuracy was limited by the small number of features tracked per frame. State of the art drift-free camera tracking for limited workspaces, represented by Klein and Murray's Parallel Tracking and Mapping system (PTAM) [5], instead relies on frame-rate pose estimation based on measurements of hundreds of features per frame, interleaved with repeated bundle adjustment in an optimised parallel implementation. It achieves highly accurate estimates for the positions of thousands of scene points and the poses a set of historic camera keyframes, and is therefore ideal for our needs. Note that we assume, as in most of the dense reconstruction literature, that the camera motion estimation from high quality structure from motion is of sufficiently high accuracy that we need not consider its uncertainty in the rest of the reconstruction pipeline.

We collect the only highest quality points from PTAM to pass on to the dense reconstruction pipeline by discarding those with an outlier count greater than 10. Another useful quantity obtained from PTAM is the co-visibility of the scene points in the set of key-frames. 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.

2.3. Base Surface Construction

Given the oriented 3D point cloud from SFM, we now aim to estimate an initial continuous scene surface which will form the basis for dense refinement. The problem of reconstructing a dense 3D surface from oriented point samples has received considerable attention in the computer graphics community and is at the base of techniques in model hole filling, efficient representation, and beautification of models. In particular, implicit surface reconstruction techniques have received considerable attention. In these, surfaces are approximated by fitting a function to the data

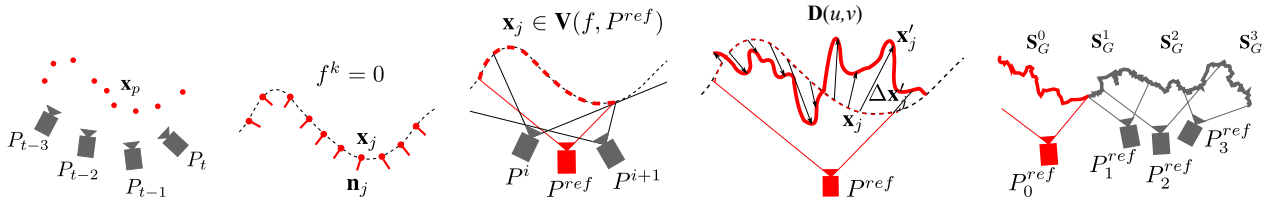

Figure 2. Live dense reconstruction pipeline overview. (a) Online structure from motion provides real-time camera pose estimates and a point map. (b) An implicit 'base surface' is fitted to the point map, and the zero level set is polygonised into a 'base mesh'. (c) Local bundles of cameras with partial visible surface overlap are selected around reference poses P^{ref}. (d) 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. (e) All local reconstructions are integrated into the global surface model and redundant vertices are trimmed.

points $\mathbb{R}^3 \mapsto \mathbb{R}$, $f(\mathbf{x}) = 0$. A reconstructed mesh is extracted by polygonising the function's zero level set [1].

In our application, speed is crucial in obtaining an up to date base model. Globally optimal non-parametric surface fitting techniques have suffered from high computational cost in solving large, dense systems of equations [13, 4]. 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 [7]. This method combines some of the best aspects of global and local function approximation and is particularly suited to the sparse, approximately oriented, point cloud obtained from the SFM processing stage, in particular retaining the ability to interpolate over large, low density regions.

Polygonisation of the zero level set, using the method of Bloomenthal [1], is also aided by MSCSRBF since the majority of computation when evaluating the implicit surface is performed at previous levels of the function hierarchy. 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.

2.4. Constrained Scene Flow Dense Reconstruction

Now we detail our novel, simple and iterative dense depth map reconstruction algorithm which relies on high quality camera tracking and the base model obtained in the previous section. In Section 2.6 we then show that the quality of the depth maps allows a simple local mesh model integration pipeline to be used to sequentially build a dense global model.

Each reference frame has a grey-scale image $\mathbf{I}^{Ref}$ and projection matrix $P^{Ref} = \mathrm{K}\left[\mathbf{R}_{cw}^{ref\,\mathrm{T}} | -\mathbf{R}_{cw}^{ref\,\mathrm{T}}\mathbf{t}_{cw}^{ref}\right]$, together with $n \geq 1$ other nearby comparison frames $\mathbf{I}^C_{i\in(1\ldots n)}$. K is the known camera calibration matrix. 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 in Section 2.7.

2.4.1 Scene Motion and Image Motion

To illuminate our method, it useful to think in terms of n static cameras viewing a deformable surface — although of course our method is only applicable to a rigid surface, and the deformation we discuss is only of the *estimate* of its shape. Our goal is to use dense image measurements across the camera bundle to deform the base surface estimate into a new, more accurate shape.

A visible surface point $\mathbf{x}_j = [x_j, y_j, z_j]^{\mathrm{T}}$ gives rise to an image projection $[u_j^i, v_j^i]^{\mathrm{T}} = \mathbf{u}_j^i = \mathbf{P}^i(\mathbf{x}_j)$ in each camera i in the bundle, where $\mathbf{P}^i(\mathbf{x}_j)$ signifies perspective projection according to projection matrix P^i. A small 3D motion of surface point $\Delta\mathbf{x}_j$ leads to a new point position $\mathbf{x}'_j$ with corresponding projection $\mathbf{u}'^i_j$. If $\Delta\mathbf{x}_j$ is sufficiently small then the image displacement $\Delta\mathbf{u}_j^i = \mathbf{u}'^i_j - \mathbf{u}_j^i$ can be approximated by the first order linearisation of $\mathbf{P}^i$ around $\mathbf{x}_j$. In this case $\Delta\mathbf{x}_j$ is called scene flow [15]:

$$\Delta\mathbf{u}_j^i = \mathbf{J}^i_{\mathbf{x}_j}\Delta\mathbf{x}_j \;, \tag{1}$$

where $\mathbf{J}^i_{\mathbf{x}_j}$ is the Jacobian for the projection function evaluated at point $\mathbf{x}_j$ with projection parameters P^i:

$$\mathbf{J}^i_{\mathbf{x}_j} \equiv \left.\frac{\partial\mathbf{P}^i}{\partial\mathbf{x}}\right|_{\mathbf{x}_j} \equiv \left.\begin{bmatrix} \frac{\partial\mathbf{P}^i_u}{\partial x} & \frac{\partial\mathbf{P}^i_u}{\partial y} & \frac{\partial\mathbf{P}^i_u}{\partial z} \\ \frac{\partial\mathbf{P}^i_v}{\partial x} & \frac{\partial\mathbf{P}^i_v}{\partial y} & \frac{\partial\mathbf{P}^i_v}{\partial z} \end{bmatrix}\right|_{\mathbf{x}_j} . \tag{2}$$

2.4.2 Optimising the Base Mesh

Our dense base mesh provides a prediction of the position (and normal) of the visible surface elements in every frame of the bundle. We wish to use image correspondence information to obtain the vertex updates $\Delta\mathbf{x}_j$ required to deform

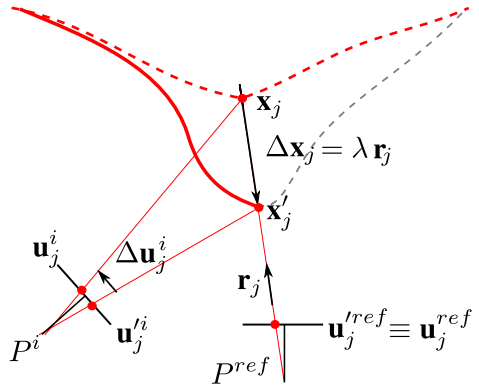

Figure 3. The geometry of constrained scene flow. All deformations of the base surface vertices must occur along viewing rays of the reference camera.

each vertex of the current base model $\mathbf{x}_j$ into a new estimate $\mathbf{x}'_j$ accurately representing the true scene.

In Section 2.5 we will describe how we obtain dense correspondence information using view-predictive optical flow, but for now assume that we can measure the image displacement vectors $\Delta\mathbf{u}^{i=1\dots n}_j$ associated with $\mathbf{x}_j$ and $\mathbf{x}'_j$. Then for $n \geq 2$ views, given $\mathbf{x}_j$ the Jacobian (2) can be computed for each static camera and an over-constrained linear system can be solved for $\Delta\mathbf{x}_j$, giving each surface point's new location $\mathbf{x}'_j = \mathbf{x}_j + \Delta\mathbf{x}_j$.

For each pixel in the reference view $\mathbf{u}^{ref}$, we back-project a ray and intersect it with the base model to determine a 3D vertex position $\mathbf{x}_j$. This is then projected into each of the comparison frames in the bundle to obtain a *predicted* image correspondence $\mathbf{u}^j$. By definition, any deformation of the visible base model does not alter the predicted image correspondence in the reference view itself, where the projection of a surface point before and after deformation is always $\mathbf{u}'^{ref} \equiv \mathbf{u}^{ref}$. Figure 3 shows that this results in an epipolar constraint between the reference view and each comparison view. The scene flow vector is restricted to lie on the ray emanating from the reference camera and intersecting the base model at $\mathbf{x}_j$ and the true surface point:

$$\Delta\mathbf{x}_j = \mathbf{r}_j\lambda_j \ , \tag{3}$$

where the ray line has the unit vector direction $\mathbf{r}_j$:

$$\begin{bmatrix} r^x_j \\ r^y_j \\ r^z_j \end{bmatrix} \equiv \mathbf{r}_j = \frac{\mathbf{R}^{ref}_{cw}[\mathbf{u}^{ref\,\mathrm{T}}_j|1]}{\|\mathbf{R}^{ref}_{cw}[\mathbf{u}^{ref\,\mathrm{T}}_j|1]\|} \ . \tag{4}$$

$\mathbf{R}^{ref}_{cw}$ is the direct cosine matrix that rotates points from the camera into the world frame of reference. Inserting the constrained scene flow (3) back into (1), this simple constraint reduces the number of unknowns in the system from three to one per vertex, λ_j:

$$\Delta\mathbf{u}^i_j = \mathbf{K}^i_j\lambda_j \ , \tag{5}$$

where $\mathbf{K}^i_j \equiv \mathbf{J}^i_{\mathbf{x}_j}\mathbf{r}_j$ is the inner product between each Jacobian with its associated reference ray.

Each displacement vector $\Delta\mathbf{u}^i_j = \mathbf{u}^i_j - \mathbf{u}'^i_j$ is computed from the predicted image projection and $\mathbf{u}'^i_j$, the image correspondence to $\mathbf{u}^{ref}_j$ in frame i from image matching.

We construct the overdetermined system by stacking each of the two linear equations obtained from the two components of each displacement vector associated with each camera $i = 1\dots n$ viewing the model vertex $\mathbf{x}_j$ into the column vectors $k_a \in \mathbf{K}$ and $u_a \in \Delta\mathbf{u}$ where $a = (1\dots 2n)$:

$$\begin{bmatrix} u^1_j \\ v^1_j \\ \vdots \\ u^{2n}_j \\ v^{2n}_j \end{bmatrix} \equiv \Delta\mathbf{u}_j \quad \begin{bmatrix} K[u]^1_j \\ K[v]^1_j \\ \vdots \\ K[u]^{2n}_j \\ K[v]^{2n}_j \end{bmatrix} \equiv \mathbf{K}_j \quad . \tag{6}$$

The least squares solution $\min_{\lambda_j}\|\mathbf{K}_j\lambda_j - \Delta\mathbf{u}_j\|$ is solved by the normal equations,

$$\lambda_j = (\mathbf{K}^{\mathrm{T}}_j\mathbf{K}_j)^{-1}\mathbf{K}_j\Delta\mathbf{u}_j \equiv \frac{\sum_{a=1}^{2n} k_a u_a}{\sum_{a=1}^{2n} k_a^2} \ , \tag{7}$$

resulting in the dense vertex-wise update of the surface:

$$\mathbf{x}'_j = \mathbf{x}_j + \mathbf{r}_j\lambda_j \ . \tag{8}$$

The computation consists of just two inner products of size $2n$ and a scalar division, which is trivially parallelisable on modern GPU hardware. In practice we obtain the final vertex update by iterating (8) using the previous solution, including computation of the intermittent Jacobians. Given that the initial base model vertex is already close to the solution, optimisation continues for up to three iterations or until the average vertex reprojection error is less than $\epsilon = 1e^{-4}$ normalised pixels.

2.4.3 Triangulating the Depth Map

The constrained scene flow optimisation produces a new vertex, $\mathbf{x}'_j$, for each reference pixel, $j \equiv (x, y)$, that is visible in at least one comparison view, from which we compute a dense depth map, $\mathbf{D}_j = \|\mathbf{t}^{ref} - \mathbf{x}'_j\|$. Prior to triangulation a median filter with a support of three pixels is applied to the depth map to reduce any spurious vertex values.

At each depth map pixel the post-processed vertex $\mathbf{v}_j$ is recovered,

$$\mathbf{v}_j = \mathbf{D}_j\mathbf{r}_j + \mathbf{t}^{ref} \ , \tag{9}$$

Each vertex is connected with its neighbours to form the triangle faces. The associated vertex normal vector is also computed, $\mathbf{n}_j = \Delta\mathbf{v}_x \times \Delta\mathbf{v}_y$.

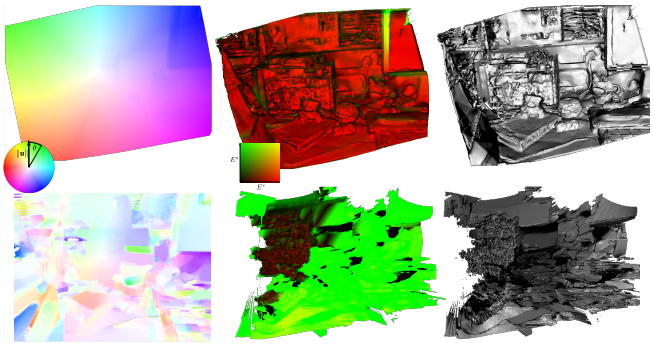

Figure 4. Reconstruction results from two images using model predictive optical flow (top row) and raw optical flow (bottom row). The polygonised results are shown with associated per-vertex error measures (middle column; red indicates high photo-consistency). A rotation around the optical axis induces large displacements of upto 150 pixels resulting in errors in the raw flow field. The ego-motion induced rotational component of the flow field is eliminated using view prediction. We recommend that these images be viewed on-screen and zoomed in.

2.4.4 Surface Errors

For each vertex $\mathbf{v}_j$ of the triangulated depth map we assign a vector $\mathbf{E}_j = [E^s, E^v]$ of measures of reconstruction fidelity. E^s is the per vertex mean reprojection error and E^v is a measure of the visibility of the surface element in the reference view:

$$E^s = \|\mathbf{K}_j \lambda_j - \Delta\mathbf{u}_j\|_2, \; E^v = |\mathbf{r}_j^i \cdot \mathbf{n}_j^i| \; . \qquad (10)$$

2.5. High Quality Dense Correspondence

The image correspondences $\mathbf{u}_j'^i$ required to solve equation (8) must be obtained for each point in the reference view. Great advances have been made in sparse image matching techniques which compute descriptor keys that are useful in wide-baseline matching. These are the mainstay for many geometric vision systems, but require highly textured image regions, do not yield sub-pixel accuracy and and are not applicable to every-pixel image matching. Instead, our image pairs, each consisting of the reference image and a comparison view, are relatively short baseline. We therefore make use of high accuracy, variational optimisation based optical algorithm to obtain dense, sub-pixel quality correspondences. Large improvements to variational optical flow solutions have been gained by utilising an $\mathbf{L}^1$ data fidelity norm and a total variation regularisation of the solution:

$$\mathbf{E}_\mathbf{u} = \int_\Omega \lambda |I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))| d\mathbf{x} \; + \int_\Omega |\nabla \mathbf{u}| d\mathbf{x} \; . \quad (11)$$

Details of solutions to minimising (11) are given in [9], [19]. Here, λ provides a user-controllable trade-off between the data fidelity and regularisation terms. The TV regularisation results in solutions that allow flow field discontinuities that are present at occlusion boundaries and the $\mathbf{L}^1$ data fidelity term increases robustness to image data error.

In our real-time reconstruction application it is necessary to compute the flow fields to obtain n-view correspondences. The number of frames that can be used in a bundle reconstruction is therefore constrained by the time taken to compute the flow fields. Recently, it has been demonstrated that the solution to the variational problem can be reformulated to obtain a highly optimised GPU implementation [17] which we utilise here. The data term in (11) is linearised and the solution is computed in a coarse to fine manner embedded in an iterative warping framework. [17] also includes photometric normalisation, improving robustness to varying lighting conditions or automatic camera parameters.

2.5.1 View-Predictive Optical Flow Correspondence

Our dense base mesh provides not only a prediction of the position and normal of a visible element at a given frame, but also of every pixel's intensity. We back-project the reference frame image onto the base model surface and project the textured model into the comparison camera frame to synthesize an image prediction in that camera. This is performed efficiently on GPU hardware using projective texturing. Variational optical flow (11) is then applied *between each synthesized image for a comparison camera and the true image captured there*, to arrive at $\Delta\mathbf{u}_j^i$ directly.

The improvement from using this *model predictive optical flow* is twofold: first, rotational flow components induced by a rotational difference between the reference and comparison view poses are removed; and second, the distance to corresponding pixels is reduced wherever the base model is approximately correct. It is important to note that a valid spatio-temporal derivative must exist at the coarsest level of the multi-scale optic flow optimisation, which places a limit on the largest displacement measurable between corresponding image points. View prediction greatly increases the applicability of optic flow to dense wider baseline matching. Figure 4 compares reconstruction results from constrained scene flow using (a) model predictive optical flow, and (b) raw optical flow. Model prediction allows us to widen the baseline over which optical flow can be used for dense correspondence, and over a short baseline improves correct correspondences in regions of homogeneous or repetitive texture.

2.5.2 Minimally Destructive View Prediction

Correspondences between two frames can only be computed when the projected surface regions are co-visible. Unfortunately, where the base model is imperfect, view synthesis will induce false discontinuities in the predicted im-

age. We therefore obtain a base model mesh in a given reference frame with reduced false discontinuities by smoothing the predicted reference view depth map. We use a Gaussian kernel with a support of 15 pixels. The smoothed depth map is then triangulated and transformed back into the global frame for use in view prediction.

2.5.3 Iterating Model Prediction

We further utilise the model predictive optical flow by performing multiple iterations of the reconstruction algorithm. Prior to triangulation a denoised version of the depth map $\mathbf{D}'$ is constructed by minimising a g-weighted TV-$\mathbf{L}^1$ denoising functional:

$$\mathbf{E}_d = \int_\Omega \gamma|\mathbf{D}(\mathbf{x}) - \mathbf{D}'(\mathbf{x})|d\mathbf{x} + \int_\Omega g(\mathbf{x})|\nabla\mathbf{D}'|d\mathbf{x} \ , \quad (12)$$

where $g(|\nabla I_{ref}|) = \exp(-\alpha|\nabla I_{ref}|^\beta)$ maps the gradient magnitude between 0 and 1, we use $\alpha = 10$, $\beta = 1$. The TV-$\mathbf{L}^1$ model has been shown to effectively remove structure below a certain scale while preserving image contrast and all larger scale structures in the image [9]. Here, the isotropic regularisation weight g from the reference image ensures that strong image boundaries, often indicators of object boundaries, are not smoothed over in the depth map, thereby providing a form of segmentation prior on the construction of $\mathbf{D}'$ [14].

$\mathbf{D}'$ is then triangulated and used in place of the original base model, improving view prediction and and the quality of optical flow computation, and ultimately increasing photo-consistency in the reconstructed model. Figure 5 illustrates the results of processing a second iteration.

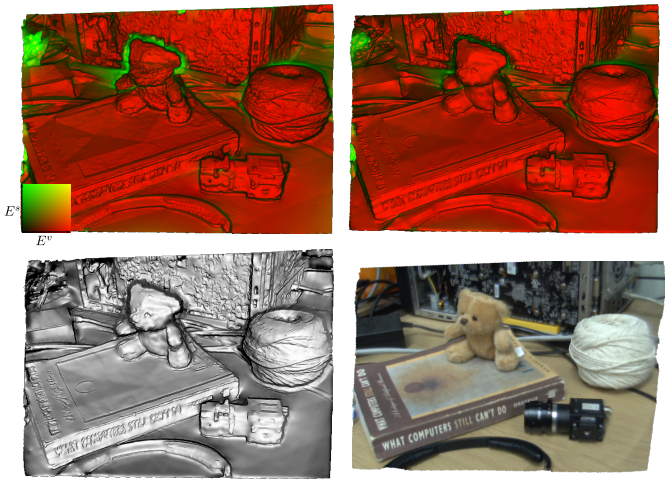

Figure 5. A local surface reconstruction using a total of four images. From one scene flow iteration and associated error measures (top left). A second iteration results in high photo consistency (top right), indicated by red colouring. The resulting Phong shaded reconstruction (bottom left) and a synthesised view using the reference camera image to texture the surface model (bottom right).

2.6. Local Model Integration

A number of algorithms have been developed to enable the fusion of multiple depth maps, traditionally obtained from structured light and range scanners, into a global model [11]. These methods can be broadly split into two categories. Those in the first class make of use of the computed depth maps, dense oriented point samples, or volumetric signed distance functions computed from the depth maps to fit a globally consistent function that is then polygonised. This class of algorithm includes optimal approaches [2, 20], though is less amenable to larger scale reconstructions due to prohibitively large memory requirements for the volumetric representation of the global functions. It is noted however that recent advances in solving variational problems efficiently on GPU hardware increase the possibility of using such methods in the future [18]. The second class of algorithms work directly on the partial mesh reconstructions. One of the earliest methods, zippered polygon meshes [12], obtained a consistent mesh topology by removing overlapping regions to form meshes that are connected at the new boundaries. [6] have demonstrated real time fusion of noisy depth maps to obtain a set of depth maps with reduced errors and free space violations that are then polygonised. Sparse solution methods such as the method used in the base mesh construction pipeline in this paper [7] and [4] are also currently computationally infeasible when a finer grain polygonisation of the level set is desired to achieve every-pixel mapping.

Due to the consistent quality of the depth maps produced we have found that a very simple approach can suffice. Given a newly triangulated mesh obtained at reference P^{ref}, we render a depth map of the currently integrated dense reconstruction into P^{ref} and remove the vertices in the new model 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 $E^v < 0.9$ and $E^s > 1e^{-3}$.

2.7. Camera Bundle Selection

Each local reconstruction requires a camera bundle, consisting of a reference view and a number of nearby comparison views, and we aim to select camera bundles to span the whole scene automatically. As the camera browses the scene the integrated model is projected into the virtual current camera, enabling the ratio of pixels in the current frame that cover the current reconstruction, $\mathbf{V}_r$, to be computed. We maintain a rolling buffer of the last 60 frames and camera poses from which to select bundle members.

When $\mathbf{V}_r$ is less than an overlap constant $\epsilon_r = 0.3$ a new

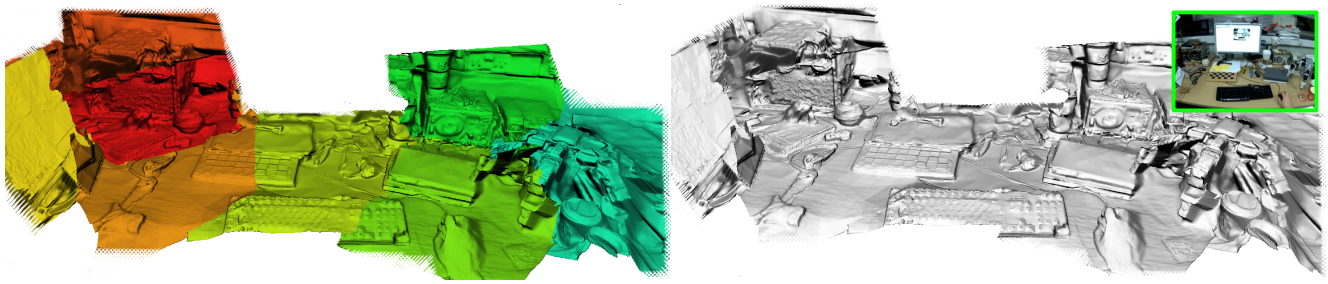

Figure 6. Full scene reconstruction obtained during live scene browsing. Eight camera bundles each containing four images including the reference were used for reconstruction. Left: each colour indicates a region reconstructed from a single camera bundle. Right: alternative rendering using diffuse Phong shading using the surface normal vectors. Various objects on the desktop are easily recognisable (we recommend viewing this image on-screen and zoomed in).

reference frame is initialised as the current frame. Given the new reference frame pose we obtain a prediction of the surface co-visibility with each subsequent frame $\mathbf{V}_c$ by projecting the intersection of the base surface and reference frustum into the new frame. When $\mathbf{V}_c < \epsilon_c$ where $\epsilon_c = 0.7$, we take all images in the frame buffer and select n informative views. The method for view selection is based on obtaining the largest coverage of different translation only predicted optic flow fields. 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 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 optic flow computation used in the dense reconstruction.

3. Results

Our results have been obtained with a hand-held Point Grey Flea2 camera capturing at 30Hz with 640×480 resolution and equipped with a 80° horizontal field of view lens. The camera intrinsics were calibrated using PTAM's built-in tool, including radial distortion. 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 1 shows a partial reconstruction, including a number of low texture objects, obtained using four comparison images per bundle from a slowly moving camera. To give an indication of scale and the reconstruction baseline used, here the camera was approximately $300mm$ 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 accurate sub-pixel correspondence. Utilising the optic flow magnitude computed between each predicted and real image we find that the final variance of the flow residual magnitudes is less than 0.04 pixels for a typical bundle reconstruction.

The full reconstruction pipeline has been thoroughly tested on a cluttered desktop environment. In Figure 6 a rendering of a full fused reconstruction is shown, with colour-coding indicating the contribution of each camera bundle. This reconstruction was made in under thirty seconds as the camera naturally browsed the scene. The fast camera motion in this experiment slightly reduces image and therefore reconstruction quality compared to the more controlled situation in Figure 1, but this reconstruction is representative of live operation of the system and highly usable.

At the heart of our dense reconstruction pipeline is view synthesis; this can also be channelled as an output in itself. Figure 5 demonstrates generation of a synthetic view using a single local reconstruction, textured using the reference image. More extensive view synthesis examples are provided in the online videos. We also demonstrate utilisation of the global dense surface model in a physics simulator. Excerpts of a simulated vehicle interacting with the desktop reconstruction are given in Figure 7. Our dense reconstruction permits augmented reality demonstrations far beyond those of [5] that are restricted to a single plane and do not provide synthetic object occlusion or interaction.

4. Conclusions

We have presented a system which offers a significant advance in real-time monocular geometric vision, enabling fully automatic and accurate dense reconstruction in the context of live camera tracking. We foresee a large number of applications for our system, and that it will give the possibility to re-visit a variety of live video analysis problems in computer vision, such as segmentation or object recogni-

[1] http://www.doc.ic.ac.uk/~rnewcomb/CVPR2010/

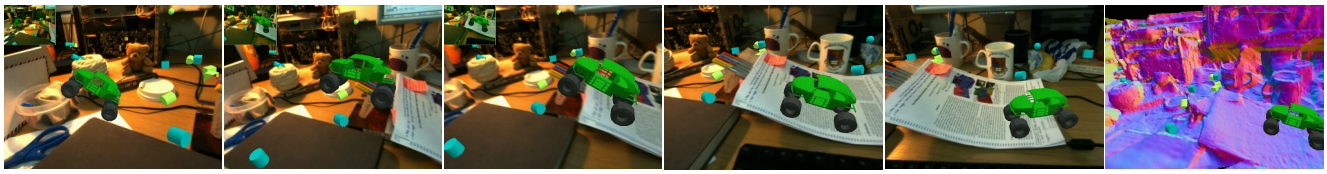

Figure 7. Use of the desktop reconstruction for advanced augmented reality, a car game with live physics simulation. Far right: the car is seen sitting on the reconstructed surface. The other views are stills from our video where the car is displayed with the live camera view, jumping off a makeshift ramp, interacting with other objects and exhibiting accurate occlusion clipping.

tion, with the benefit of dense surface geometry calculated as a matter of course with every image. Our own particular interest is to apply the system to advanced mobile robotic platforms which aim to interact with and manipulate a scene with the benefit of physically predictive models.

Acknowledgements

This research was funded by European Research Council Starting Grant 210346 and a DTA scholarship to R. Newcombe. We are very grateful to Murray Shanahan, Steven Lovegrove, Thomas Pock, Andreas Fidjeland, Alexandros Bouganis, Owen Holland and others for helpful discussions and software collaboration.

References

[1] J. Bloomenthal. An implicit surface polygonizer. In *Graphics Gems IV*, pages 324–349. Academic Press, 1994. 3

[2] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *ACM Transactions on Graphics (SIGGRAPH)*, 1996. 6

[3] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2003. 1, 2

[4] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Eurographics Symposium on Geometry Processing*, 2006. 3, 6

[5] 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. 1, 2, 7

[6] 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, 6

[7] 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. 3, 6

[8] 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. 1

[9] T. Pock. *Fast Total Variation for Computer Vision*. PhD thesis, Graz University of Technology, January 2008. 5, 6

[10] 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. 1

[11] 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. 1, 6

[12] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *ACM Transactions on Graphics (SIGGRAPH)*, 1994. 6

[13] G. Turk and J. F. OBrien. Variational implicit surfaces. Technical Report GIT-GVU-99-15, 1999. 3

[14] M. Unger, T. Pock, and H. Bischof. Continuous globally optimal image segmentation with local constraints. In *Proceedings of the Computer Vision Winter Workshop*, 2008. 6

[15] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999. 3

[16] G. Vogiatzis, P. H. S. Torr, S. M. Seitz, and R. Cipolla. Reconstructing relief surfaces. *Image and Vision Computing (IVC)*, 26(3):397–404, 2008. 2

[17] 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. 5

[18] 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. 6

[19] 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*, 2007. 5

[20] 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)*, 2007. 6