Hybrid modelling of non-rigid scenes from RGBD cameras - Supplementary Material

Charles Malleson *Member, IEEE,* Jean-Yves Guillemaut *Member, IEEE,* and Adrian Hilton

I. Generation of input surfel graph from RGBD sequences

The optical flow-based point tracker of Sundaram *et al*. [15] is used to produce a set of 2D tracks $\hat{\mathbf{p}}_p(t)$ from an input RGB image sequence $C(t)$. This tracker is based on concatenation of frame-to-frame optical flow fields, subject to various consistency checks. The optical flow method uses frame-to-frame feature matches in order to handle large displacements. Likely invalid tracks are filtered firstly by doing a forward-backward consistency check to detect occluded regions, secondly by removing points from unstructured image regions, and thirdly by removing tracks on motion boundaries. New tracks are introduced to fill unoccupied areas resulting from disocclusion or appearance of new surface regions, thus maintaining tracking density.

For efficient computation in the subsequent piecewise rigid segmentation and modelling stage, the dense flow field is subsampled (Fig. 1). In the experiments, decimation factors of 4 and 8 were used for the Kinect v1 (VGA resolution) and Kinect v2 (HD resolution) sequences, respectively.

A. Conversion to 3D and connectivity estimation

The 2D point tracks are converted to a set of 3D surfels $\mathcal{P} = \{\mathbf{p}_p(t)\}$, indexed by p. The 3D positions $\mathbf{p}_p(t)$ are obtained by back-projecting the 2D point track $\hat{\mathbf{p}}_p(t)$ using the input depth maps $D(t)$. Filtration of the point tracks is then performed, removing any points which are within a band of a depth edge, as these are liable to switch between local foreground and background depths, and are thus unreliable. For the experiments, a band of 4 pixels was found to be suitable.

The connectivity matrix $\mathbf{E}$ is then established. On the first frame, edges are added for all visible surfels. For all subsequent frames edges are added for all surfels which became visible for the first time in that frame.

The connectivity is estimated using k-nearest neighbours. Both 2D (image plane) and 3D nearest neighbours were considered. Because the sampling of the input point tracks is roughly uniform in the horizontal and vertical directions in the image plane, 2D neighbourhoods tend to produce edges in all directions. If 3D neighbourhoods are used, however, on obliquely viewed surfaces, all the edges produced tend to lie in one direction according to the surface orientation and the resulting graph may not be well connected. For this reason the 2D approach was chosen. The putative edge candidates established in 2D may straddle depth discontinuities (*i.e*. connect two separate surface regions), therefore edges are only added if their projection into the depth map does not cross any depth discontinuities (Fig. 2).

An edge stretch test is then performed in 3D. Edges are removed if the ratio of their maximum to minimum length over the sequence is above a threshold (5.0 was used in all experiments). This disconnects most regions which should not be connected (two surfaces that touch in part of the sequence), while maintaining connections deforming surfaces which stretch moderately. The 4 nearest valid neighbours are kept. For sequences which feature changes in topology, this helps to produce surfel graphs with topology that more closely matches the true underlying topology of the scene (*i.e*. with surface regions separated part of the time not being connected in the surfel graph), but does not guarantee that the surfel graph will be free of errors in topology all cases.

Finally, very short tracks (with fewer than 15 frames) are removed from the surfel graph, as these tend not to have a useful contribution to the final model, but are liable to cause artifacts due to incorrect part assignment and connectivity, by virtue of the limited period over which their motion is observed.

II. Configuration of voxel grids

The following subsections describe the procedure for configuring the volumetric grids for the part and composite models, respectively.

A. Part voxel grids

For each part model $m \in \mathcal{M}$, the set of its intrinsic points $\{\mathbf{r}_p^m \in \mathcal{P}_m\}$ are used to configure the part's voxel grid G_m. While the exact sizing and initial positioning of these grids is not critical, for efficiency it is desirable to have their 3D bounding boxes sized and posed so as to enclose $\mathcal{P}_m$ with close to the minimal volume. The following simple approximate method is used.

The centroid of $\{\mathbf{r}_p^m \in \mathcal{P}_m\}$ is used to define the part's local origin. The eigenvectors of the covariance matrix of these points, taken in descending order, define putative directions for the part's local x, y, and z axes, in order. The x and y axes are then refined by rotating them about the local z axis such that the bounding box volume is minimized.

The size of the voxel grid is then set such that it encloses all the points, with a small buffer added at each edge so that an iso-surface can be reliably extracted even for points near the minimal boundaries.

The authors are with The Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH, United Kingdom: e-mail: {charles.malleson, j.guillemaut, a.hilton} @surrey.ac.uk.

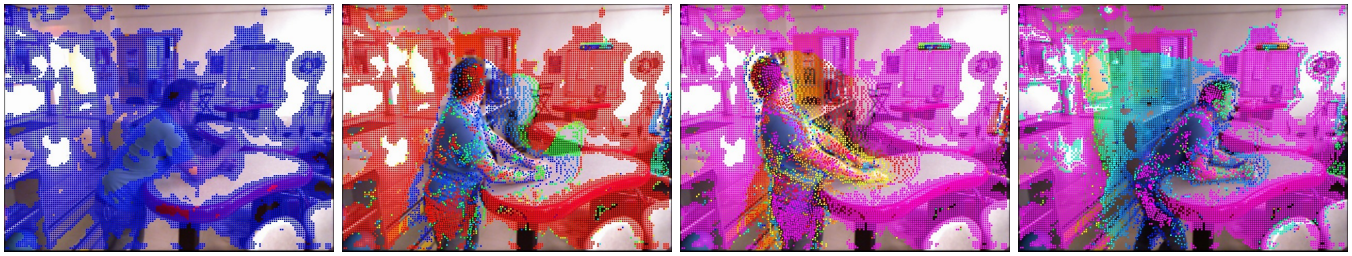


Fig. 1. Four frames of the *Paris* sequence tracked using the optical flow-based point tracking of Sundaram *et al.* [15]. The colour of the tracks indicates their age (linear scale $_0$ $_{150\ \text{frames}}$). Note the newly introduced tracks in the un-occluded regions, and the absence of tracks on the background wall and table, which lack the image structure needed for reliable flow estimation.

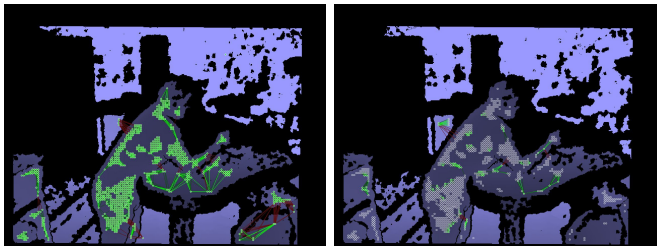


Fig. 2. Adding of surfel graph edges (green) using k-nearest neighbours in 2D from point tracks (white). Edges which straddle depth discontinuities are discarded (red). Left: First frame (edges added for all tracks). Right: Subsequent frame (edges only added for newly visible tracks).

B. Composite voxel grid

The size and pose of the composite voxel grid G_c are determined as described above, but using the mean and covariance matrix eigenvectors of the set of all part grid corner points (as posed at the reference frame t_{cr}).

III. Additional Results

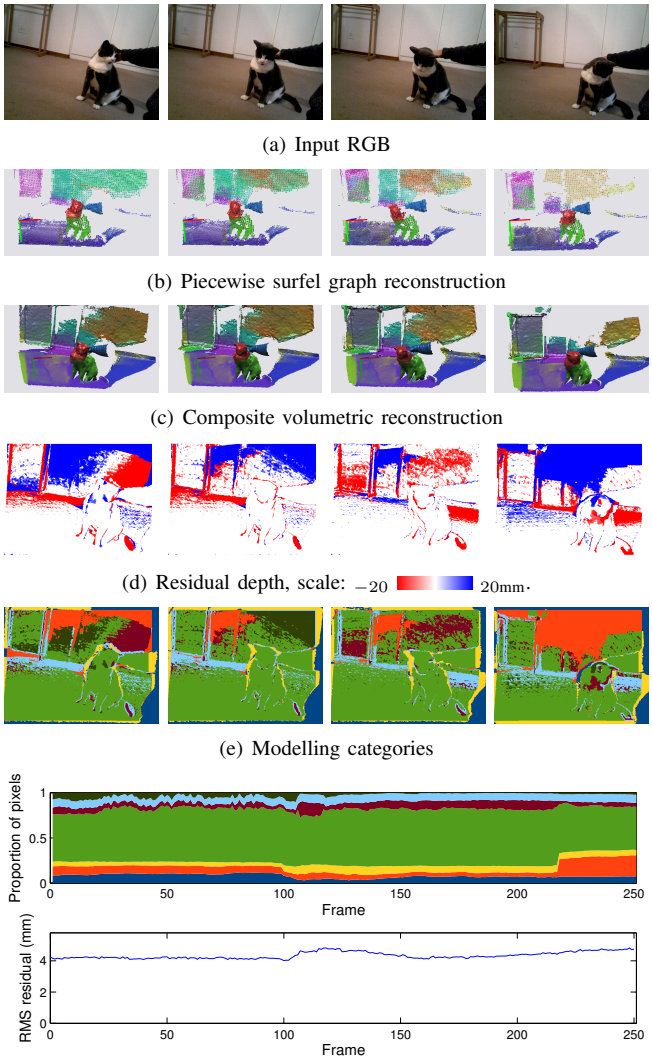
A. Kinect v1

The *Cat* sequence, shown in Fig. 3, features low-textured regions and high image noise; as a result, drift is present in the optical flow-based tracking, which results in multiple parts being used to represent the background. Nevertheless, good quality segmentations and reasonable tracks of the dynamic elements are obtained. Fig. 4 shows the results of the proposed hybrid reconstruction on the piecewise rigid *Rabbit and Deer* scene.

Fig. 5 shows results for the *Turning* sequence with and without the piecewise surfel graph model concatenation and re-iteration (Section IV-C of the main paper) being performed. Note that the piecewise surfel graph is more compact and the volumetric processing results are more complete when using the model combination stage. To further motivate the the approach, results on the *Globe* sequence with and without the piecewise surfel graph model concatenation are shown in the supplementary video.

B. Kinect v2

Fig. 6 shows results for the *Shirt* and *Sitting* sequences. The piecewise surfel graph modelling and subsequent volumetric processing work as intended, resulting in a seamless non-rigid



(a) Input RGB

(b) Piecewise surfel graph reconstruction

(c) Composite volumetric reconstruction

(d) Residual depth, scale: $_{-20}$ $_{20\text{mm}}$.

(e) Modelling categories

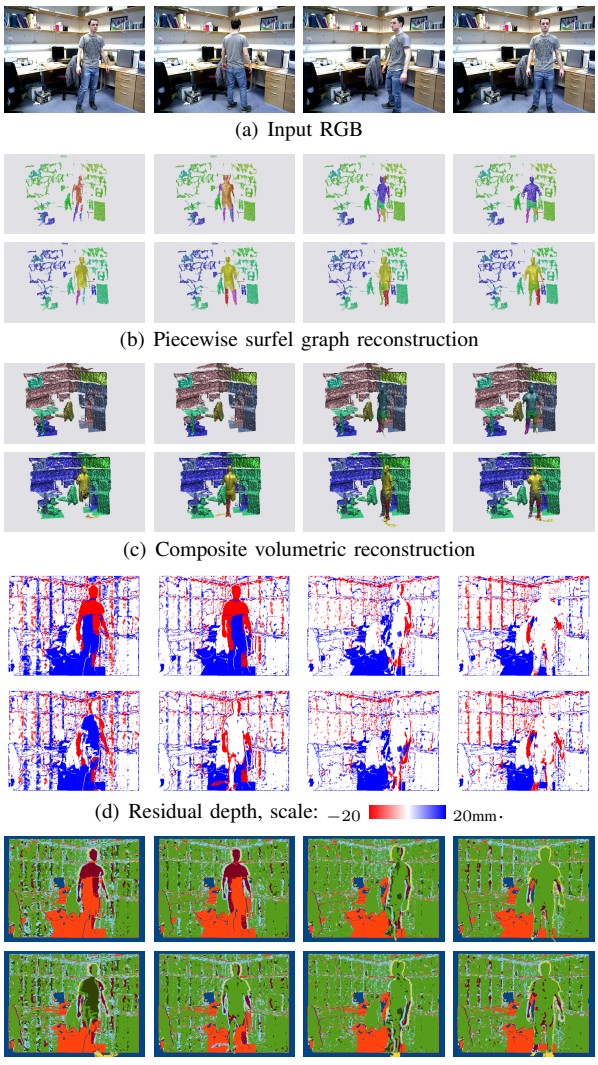
(f) Top: Proportion of image pixels in each modelling category (1-7, see Table 1 in the paper) Bottom: RMS error in consistent (category 4) regions.

Fig. 3. Hybrid processing of the *Cat* sequence.

(a) Input (b) Surfel graph

(c) Piecewise surfel graph model (d) Composite mesh model

Fig. 4. Hybrid processing of *Rabbit and Deer* sequence.



(a) Input RGB

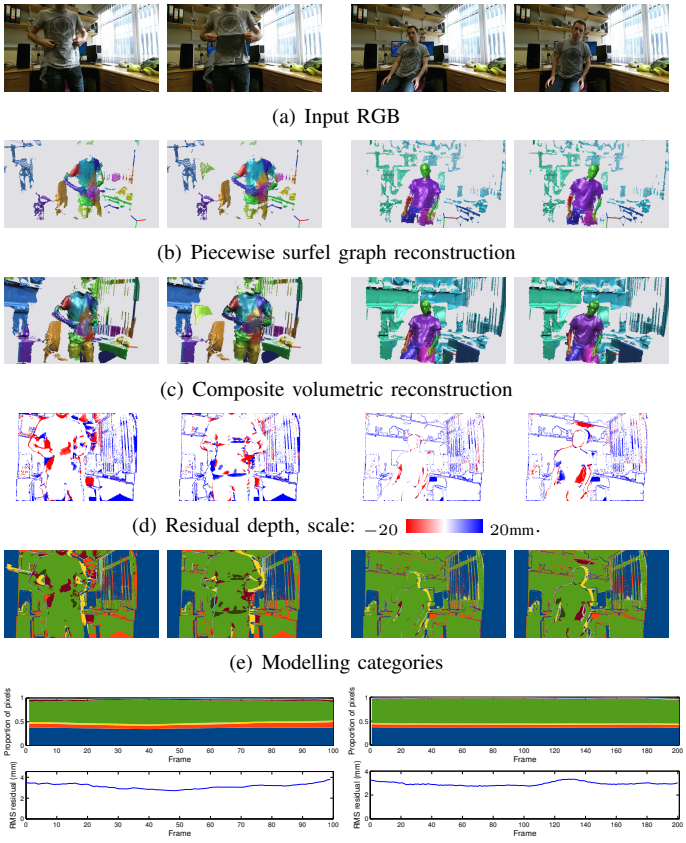
(b) Piecewise surfel graph reconstruction

(c) Composite volumetric reconstruction

(d) Residual depth, scale: -20 20mm.

(e) Modelling categories

Fig. 5. Hybrid processing of *Turning* sequence, without (top rows of (b-e)), and with (bottom rows of (b-e)) the part concatenation and re-iteration stage of the piecewise surfel graph modelling. Note the improved temporal completeness of the output when using the piecewise surfel graph model concatenation.



(a) Input RGB

(b) Piecewise surfel graph reconstruction

(c) Composite volumetric reconstruction

(d) Residual depth, scale: -20 20mm.

(e) Modelling categories

(f) Top: Proportion of image pixels in each modelling category (1-7, see Table I in the main paper) Bottom: RMS error in consistent (category 4) regions.

Fig. 6. Hybrid processing of the *Shirt* (left) and *Sitting* (right) sequences.

surface reconstruction for the subject while simultaneously modelling the background.

The *Entrance* scene shown in Fig. 7, was captured with a Kinect v2[1] as part of the SCENE project [2], which investigated the use of RGB plus depth capture for film production. This sequence is challenging due to significant changes in visibility and limited texture for point tracking, especially for the background (Fig. 7(c)). The opening door is reconstructed correctly despite limited point tracks being available (left hand frame). The static background is almost fully reconstructed, but not as a single part, due to the limited number of, and noise in, the point tracks. Both actors are reasonably well segmented and modelled.

We further test our approach on sequences from the University of Tsinghua dynamic RGBD dataset [31], allowing comparison of the reconstructions with those generated by Guo *et al.* [31]. Note that the results from our method are not directly comparable to Guo *et al.* , since Guo *et al.* use a pre-scanned template of the foreground shape which is deformed using only depth input, whereas we reconstruct the entire scene from scratch without using any pre-scanned

[1]The pre-release version of the Kinect v2 used had issues with synchronization between RGB and depth streams. The best constant offset was used to approximately align the RGB and depth streams after recording, but several frames in the sequence remain poorly synchronised.

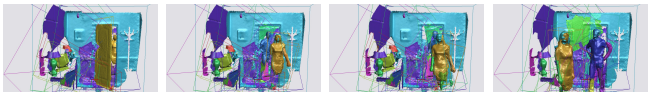
[2]EU Project SCENE www.3d-scene.eu

(a) Input RGB

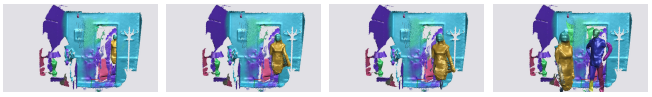
(b) Input depth

(c) Raw surfel graph

(d) Piecewise surfel graph reconstruction



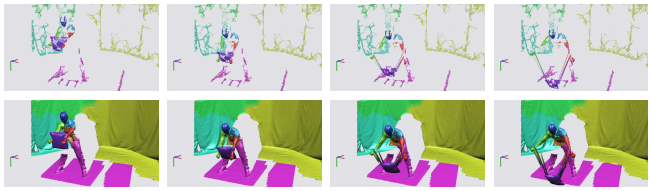
(e) Piecewise volumetric reconstruction



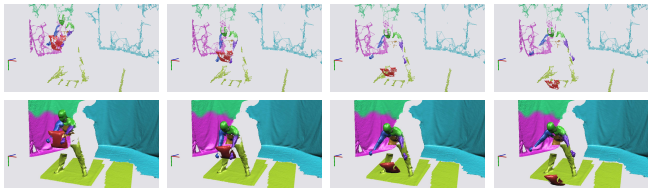
(f) Composite volumetric reconstruction

Fig. 7. Hybrid processing of the *Entrance* sequence featuring a background set with an opening door as well as two dynamic actors.

(a) Input



(b) Results without edge stretch test and mesh cleanup



(c) Results with edge stretch test and mesh cleanup

(d) Results of Guo *et al.* [31] (using a template shape)

Fig. 8. Results on the *Pillow1* sequence showing piecewise surfel graph and composite mesh results without (b) and with (c) stretched edge removal/mesh filtering. Note that the stretched edge removal and filtering of the composite mesh enables handling of the change in topology that occurs when the pillow is dropped. A comparison with Guo *et al.* [31] is shown in (d).

templates and use the colour images from the input as well as the depth. Results for the *Pillow1* and *Pillow2* sequences are shown in Figs 8 and 9. The results on the *Pillow1* sequence demonstrate the handling changes in topology in the input sequence. The stretched edge removal and output mesh filtering in our approach mitigate the severe artifacts caused by the subject's arms being joined to the pillow in the output model. Some less severe artifacts are, however, still present in the form of holes in the surfaces where they were connected. Note that, for this sequence, it would be possible to mitigate topology problems in the output by (*e.g.* manually) selecting a suitable reference frame, t_{cr}, where the arm and pillow are not in contact with one another. Future work could investigate methods for improved handling of changing topology within our proposed framework.

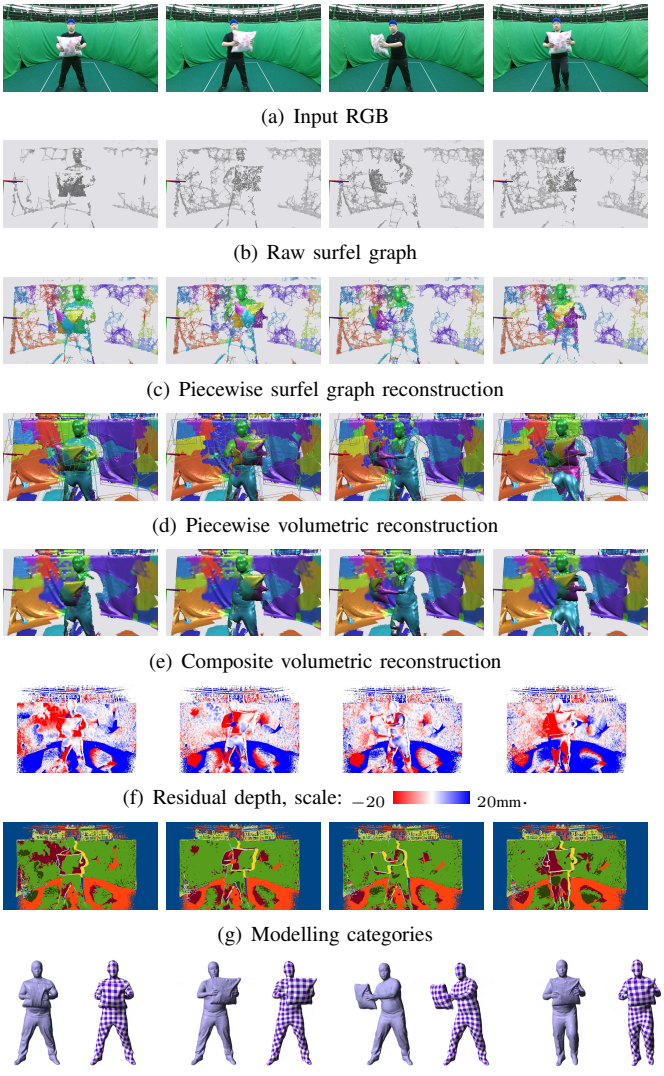
C. Effect of segmentation regularization weighting

Varying the parameters of λ_s, and MDL_m controls the weighting of the segmentation regularization (smoothness and number of models used). As the weightings are increased, fewer parts are used, resulting in smaller deforming regions being combined and modelled as single parts. Fig. 10 shows the segmentation and modelling results for the *Pillow1* sequence over a range of regularization weightings producing between 2 and 72 parts. This illustrates that when the regularization is weighted too highly, there are not enough parts to properly represent the non-rigid scene, an under-segmentation causing severe artifacts in the output shape and motion (Fig. 10(d)). On the other hand, if too little weight is placed on the regularization terms, an over-segmentation occurs, which results in redundant models, particularly in the background. This can result in the output model being less complete (Fig. 10(a)). At intermediate values (*e.g.* Fig. 10(b)), the compactness of the model and its modelling fidelity are balanced, leading to a compact model that accurately represents the shape and motion in the scene.

D. Evaluation using RGBD scene flow for point tracking

The point tracks used in the main experiments are obtained by lifting 2D optical flow-based tracks from Sundaram *et al.* [15] to 3D using the depth maps (Section I). An alternative to this would be to obtain 3D point tracks from an RGBD scene flow approach.

In additional experiments shown in Fig. 11 and in the supplementary video, the primal-dual RGBD scene flow of Jaimez *et al.* [17] is used to generate frame-to-frame 3D flows. Similar to Sundaram, these frame-to-frame flows are converted to point tracks by concatenation with sub-pixel interpolation, discontinuing tracks when the flows are not forward-backward consistent, or when the gradient of the flow field suggests that they are close to an object boundary. Fig. 11 compares results for optical flow and scene flow-based tracks on the *Cat*, *Dog* and *Paris* scenes. In all three cases, the results using lifted optical flow-based tracks have fewer artifacts in the output model. This may be due to the specific flow implementations, for instance the fact that Sundaram uses features to obtain



(a) Input RGB

(b) Raw surfel graph

(c) Piecewise surfel graph reconstruction

(d) Piecewise volumetric reconstruction

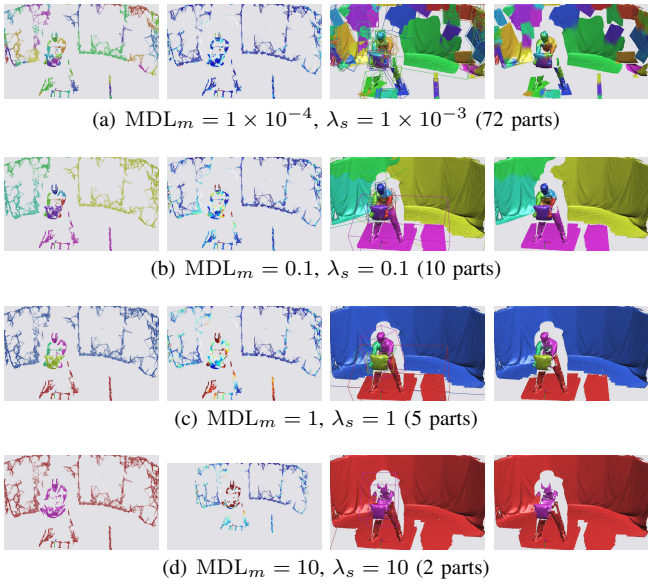
(e) Composite volumetric reconstruction

(f) Residual depth, scale: -20 20mm.

(g) Modelling categories

(h) Results of Guo *et al.* [31] (using a template shape)

Fig. 9. Hybrid processing of the *Pillow2* sequence comparing the result to Guo *et al.* [31]. Although our result suffers from some reconstruction artifacts, it reconstructs the entire dynamic scene including background without a template.



(a) $\mathrm{MDL}_m = 1 \times 10^{-4}$, $\lambda_s = 1 \times 10^{-3}$ (72 parts)

(b) $\mathrm{MDL}_m = 0.1$, $\lambda_s = 0.1$ (10 parts)

(c) $\mathrm{MDL}_m = 1$, $\lambda_s = 1$ (5 parts)

(d) $\mathrm{MDL}_m = 10$, $\lambda_s = 10$ (2 parts)

Fig. 10. Modelling results for the *Pillow1* sequence using a various regularization weightings. From left to right: piecewise surfel graph model, piecewise surfel graph model error (linear scale: 0 20mm, w.r.t. input surfel graph), piecewise volumetric model, and composite volumetric model. Note that as the regularization is increased, the granularity of the segmentation becomes coarser, and smaller deformations are no longer represented.

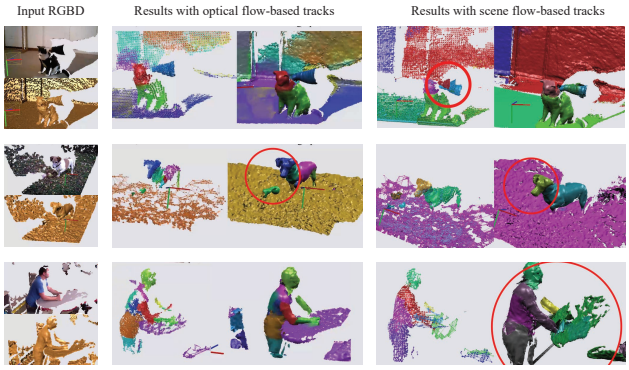


Fig. 11. Comparison of results on the *Cat*, *Dog* and *Paris* scenes (left) using depth-lifted optical flow tracks (centre) and RGBD scene flow-based tracks (right). Note the increased level of reconstruction artifacts when using the scene flow-based tracks: phantom arm reconstructions in the *Cat* sequence, missing tennis ball in the *Dog* sequence, and errors in tracking the table top in the *Paris* sequence.

Processing time (minutes)	*Paris*	*Globe*	*Dog*	*Shirt*
Num. frames	251	300	610	100
Num. 2D point tracks	31.5 k	11.7 k	42.5 k	33.6 k
Num. surfels	2.7 k	5.8 k	3.8 k	11.7 k
Flow-based point tracking	105	126	750	890
Surfel graph generation	3	4	7	10
Piecewise surfel modelling	9	27	102	9
Piecewise volumetric modelling	24	58	150	330
Composite volumetric modelling	1	1	1	1
Residual depth map computation	2	2	6	10
Total (hours)	2.4	3.6	16.9	20.8

TABLE I
PROCESSING TIMES FOR EACH STAGE OF THE HYBRID RECONSTRUCTION PROCEDURE FOR A RANGE OF SEQUENCES (3.4 GHZ INTEL CORE I7, NVIDIA GEFORCE GTX 560 TI GPU).

better long-range flow estimation, making it less likely to lose tracks. With the *Cat* sequence, however, the low-textured wall in the background has been better tracked by the scene flow approach leading to a more compact and coherent model of the background. We note that our approach is agnostic as to the source of the input point tracks.

IV. COMPUTATION TIME

The stages of processing take of the order of minutes to hours for typical sequences (Table I). Most of the code is unoptimized and runs on a single CPU thread, with the exception of the volumetric fusion, which is implemented on the GPU. Parts of the piecewise surfel graph modelling could potentially be parallelized on the GPU or multiple threads of the CPU. A significant speed-up could be obtained by using a GPU implementation of the point tracking [15], which currently takes the bulk of the processing time.