Multi-Frame Scene-Flow Estimation Using a Patch Model and Smooth Motion Prior

Thomas Popham
tpopham@dcs.warwick.ac.uk
Abhir Bhalerao
abhir@dcs.warwick.ac.uk
Roland Wilson
rgw@dcs.warwick.ac.uk

Department of Computer Science,
Warwick University
CV4 7AL

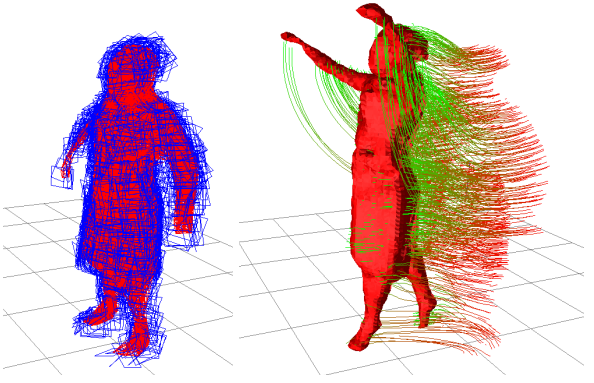
Abstract

This paper addresses the problem of estimating the dense 3D motion of a scene over several frames using a set of calibrated cameras. Most current 3D motion estimation techniques are limited to estimating the motion over a single frame, unless a strong prior model of the scene (such as a skeleton) is introduced. Estimating the 3D motion of a general scene is difficult due to untextured surfaces, complex movements and occlusions. In this paper, we show that it is possible to track the surfaces of a scene over several frames, by introducing an effective prior on the scene motion. Experimental results show that the proposed method estimates the dense scene-flow over multiple frames, without the need for multiple-view reconstructions at every frame. Furthermore, the accuracy of the proposed method is demonstrated by comparing the estimated motion against a ground truth.

1 Introduction

Estimating the motion of a scene has been a longstanding computer vision problem, and most research has focused upon estimating the 2D motion in an image. However, estimating the long-term 3D motion of a general scene remains a challenging problem. Many scene-flow estimation algorithms are therefore only demonstrated for motions over a single frame [4, 18, 19], unless multiple view-reconstructions are carried out at every frame [3, 5, 14], or a model-based approach is adopted [8, 12, 16].

Estimating the scene-flow of a general scene is difficult since some scene surfaces may contain little texture, but may move in a complex way. This leads to an underconstrained problem, unless some prior knowledge about the form of the solution is introduced. For the specific case of tracking humans, a skeleton model can be used to constrain the space of possible solutions[12]. However, for more general scenes, it is usually necessary to enforce smooth motion, with this being the approach taken by nearly all optical-flow techniques [1]. There are two current approaches to enforcing smooth motion in scene-flow estimation: either the estimated motions are regularized within the images [6, 18], or 3D translations are regularized on a surface model, using the assumption that all motion estimates are made with the same level of accuracy [2, 14].



(a) (b)

Figure 1: (a) Fitted patches at frame 60 of 'Skirt' sequence. (b) The patches are then tracked accurately and smoothly to their corresponding positions at frame 100, with the visual hull at frame 100 shown for reference. The colour coding shows the temporal aspect of tracks, with red representing the earlier frames, and green representing the later frames.

This paper takes the approach of estimating the motion directly on the model surface, through the use of a surface patch model (see figure 1). The planar patches are fitted to the scene's surfaces at the first frame using a local reconstruction technique [9], and are then tracked through subsequent frames using a novel smooth motion prior. This prior is the contribution of this paper and it consists of two elements to overcome the limitations of existing scene-flow techniques. Firstly, *rotations* as well as translations are estimated at each point of the scene surface, so that the motion of neighbouring surfaces points can be more accurately predicted than estimates containing only a translation component. Secondly, a measurement covariance is associated with each surface motion estimate, so that the regularization can be carried out in probabilistic sense: surface patches which have accurate motion estimates (due to a sufficient level of texture) are used to constrain neighbouring patches which contain less texture. The proposed method is able to estimate the scene-flow over multiple frames, and results are demonstrated on two multi-camera sequences and compared against manually entered ground truth motions. The rest of this paper is organized as follows: section 2 reviews the related work in the area; section 3 describes the theory of the proposed method; section 4 shows some experimental results on two multi-camera sequences; and section 5 concludes the paper.

2 Related Work

One way to estimate the 3D motion of the scene is to combine 2D motion estimates from several cameras [6, 18, 19]. Since existing state-of-the-art optical techniques can be applied to the problem, scene flow estimates can be readily obtained since the motion regularization can be efficiently performed in each camera image. However, this approach suffers from the disadvantage that the motion regularization is performed with no depth information.

Several methods estimate the motion of the scene at discrete points on the scene surface. Pons *et al*. use a variational approach to estimating the motion at each node of a mesh, which is deformed using the Laplace operator as the regularizing energy [14]. High-quality meshes are shown for a sequence containing small movements, however it is necessary to re-run the reconstruction algorithm at every frame. Carceroni and Kutulakos estimate the complex motion of a set of surface elements (surfels) over a single frame [4]. Mullins *et al*. use a particle filter to estimate the motion of a set of patches using a hierarchical Gaussian Mixture Model as a motion prior [13].

Many multi-camera performance capture methods focus on providing a sequence of temporally consistent meshes, by using the mesh at the previous frame to constrain the mesh at the next frame, often using a mesh-deformation framework [2]. This tends to be carried out using a feature-based approach, and nearly all methods require multi-view reconstructions at every frame [3, 5]. A final group of methods focus on matching surface points between meshes which can be several frames apart. Starck and Hilton [17] use corner, edge and region descriptors in a Markov Random Field framework in order to estimate correspondences between arbitrary frames of captured surfaces. Zaharescu *et al*. [20] define a set of features on the mesh surface, and show superior results compared to SIFT features. The main problem with this approach is that if the scene contains surfaces with no texture, then no surface features will be detected and estimating correspondences will be difficult for these regions.

In this work we take a surface patch-based approach to scene-flow estimation similar to [4, 13], however the main difference is that we employ an effective smooth motion prior, enabling motion estimates to be made over several frames. Since the motions are estimated and regularized on the surface, a common problem involved in optical-flow estimation is avoided: motion-discontinuities due to object boundaries. Since we make use of patches that are densely fitted to the scene's surfaces, the proposed algorithm provides dense motion estimates, even in non-textured areas where feature-based approaches struggle [7, 20] and does not require a computationally intensive stereo algorithm at every frame [5, 14].

3 Smooth Tracking of Planar Patches

3.1 Planar Patch and Motion Model

In order to smoothly track the non-rigid motion of a scene's surfaces, a piecewise-description of the surfaces is used: a set of planar patches. Each planar patch is described by six components: three parameters x,y,z describing the patch centre and three parameters $\theta_x,\theta_y,\theta_z$ describing the patch orientation. The patches are fitted to the scene surfaces using a local surface reconstruction method such as the ones described in [9, 15]. The vector $\mathbf{x}_t$ is used to represent the complete description of a patch at time t: $\mathbf{x}_t = (x,y,z,\theta_x,\theta_y,\theta_z)^T$.

Each patch is assigned a texture model $T(\mathbf{p})$, which is a function of the pixel co-ordinates on the patch surface. An image J_c from camera c may be projected onto the patch by using the projection function $H_c(\mathbf{x},\mathbf{p})$, which maps a point $\mathbf{p}$ on the patch surface into the image plane of camera c, according to the state $\mathbf{x}$ of the patch. The projected image from camera c onto the patch is therefore: $J_c(H_c(\mathbf{x},\mathbf{p}))$. The problem is then to estimate the motion of each patch from time $t-1$ to time t. The state of the patch at time $t-1$ is $\mathbf{x}_{t-1}$ and the motion of the patch between $t-1$ and t is:

$$\mathbf{d}_t = \Delta\mathbf{x}_\mathbf{t} = \mathbf{x}_t - \mathbf{x}_{t-1}. \quad (1)$$

In order to simplify the notation in the following sections, u, v, w shall be used to denote the velocity of the patch in the x, y, z directions, and θ_u, θ_v and θ_w shall denote the rotation motion of the patch.

3.2 Global Energy Model

Since the 3D tracking problem is under-defined, the range of possible patch motions must be limited through the use of a prior, which enforces local motion smoothness. The total energy that must be minimised therefore consists of a data term and a smoothness term:

$$E = E_{data} + \lambda E_{smooth}, \tag{2}$$

where λ controls the strength of the smoothness prior. When estimating the motion of a patch across a single frame, one would expect that the patch texture T to remain consistent with the set of input images J. A suitable cost function for the motion $\mathbf{d}_t$ is the sum-of-squared differences between the patch texture and input images projected onto the patch:

$$E_{data} = \sum_{c \in \mathscr{C}} \sum_{\mathbf{p} \in \mathscr{W}} \left(T(\mathbf{p}) - J_c(H_c(\mathbf{x}_{t-1} + \mathbf{d}_t)\mathbf{p}) \right)^2, \tag{3}$$

where $\mathscr{C}$ is the set of cameras and $\mathscr{W}$ is the set of pixels on the patch. A suitable smoothness energy term may be defined by penalizing the difference between neighbouring motions. A classic example of such an approach is the work of Horn and Schunck [10], who used the following smoothness energy term for 2D optical-flow:

$$E_{smooth} = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2, \tag{4}$$

where u is the flow in the x direction and v is the flow in the y direction. Generalizing this smoothness term to the 3D patch tracking problem yields:

$$E_{smooth} = \left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial z}\right)^2 + \ldots + \left(\frac{\partial \theta_w}{\partial z}\right)^2 \right). \tag{5}$$

3.3 Energy Minimization

In order to minimise the cost function in equation (2), a first order Taylor-series expansion of the projected image J is taken, allowing the patch motion to be predicted using a set of motion templates. The first-order Taylor expansion of the projection function yields:

$$E_{data} = \sum_{c \in \mathscr{C}} \sum_{\mathbf{p} \in \mathscr{W}} \left(T(\mathbf{p}) - J_c(H_c(\mathbf{x}_{t-1}, \mathbf{p})) - \left.\frac{\partial J_c}{\partial \mathbf{x}}\right|_{\mathbf{x}_{t-1}} \mathbf{d}_t \right)^2. \tag{6}$$

Differentiating the total energy $E_{total} = E_{data} + \lambda E_{smooth}$ with respect to the displacement

then gives:

$$\frac{\partial E}{\partial \mathbf{d}} = \nabla J_c \left(T - J_c - \nabla J_c \mathbf{d}_t\right) + \lambda \begin{pmatrix} \nabla^2 u \\ \nabla^2 v \\ \nabla^2 w \\ \nabla^2 \theta_u \\ \nabla^2 \theta_v \\ \nabla^2 \theta_w \end{pmatrix}, \tag{7}$$

where ∇J_c is the simplified notation for the Jacobian matrix $\frac{\partial \mathbf{J}_c}{\partial \mathbf{d}}$ and ∇^2 is the Laplacian operator. Now setting $\frac{\partial E}{\partial \mathbf{x}} = 0$ and dropping the time subscripts,

$$(\nabla J)^T (\nabla J)\mathbf{d} = \nabla J(T - J) + \lambda \nabla^2 \mathbf{d}. \tag{8}$$

Approximating the Laplacian $\nabla^2 u$ with the difference between u and the neighbouring mean $\bar{u}$, allows us to write the motion in terms of the image gradients and the neighbouring motions:

$$\mathbf{d} = \left((\nabla J)^T \nabla J + \lambda \mathrm{I}\right)^{-1} \left(\nabla J(T - J) - \lambda \bar{\mathbf{d}}\right), \tag{9}$$

where I is the identity matrix. This set of equations (six for each patch) can be solved using the Gauss-Seidel algorithm, which updates $\mathbf{d}$ at iteration $k+1$ using the neighbouring information and image terms evaluated at the previous iteration k:

$$\mathbf{d}^{k+1} = \left((\nabla J)^T \nabla J + \lambda \mathrm{I}\right)^{-1} \left(\nabla J(T - J) - \lambda \Delta \bar{\mathbf{d}}^k\right). \tag{10}$$

The model in equation (9) assumes that each patch is tracked with equal error. However, in reality the errors will be different at each patch. The vectors $\mathbf{d}$ and $\bar{\mathbf{d}}$ are therefore treated as the means of Gaussian distributions with covariances S and $\bar{S}$. Therefore equation (10) is modified to give $\mathbf{d}^{k+1}$ as a product of the measurement and neighbouring distributions:

$$S^{k+1} = \left(M^{-1} + (\bar{S}^k)^{-1}\right)^{-1} \tag{11}$$

$$\mathbf{d}^{k+1} = S^{k+1} \left(M^{-1} E^k - (\bar{S}^k)^{-1} \bar{\mathbf{d}}^k\right), \tag{12}$$

where E^k is estimate $\left((\nabla J)^T \nabla J\right)^{-1} \nabla J(T - J)$, M is the measurement covariance, S^{k+1} is the covariance of the patch at iteration $k+1$, and $\bar{S}^k$ is the covariance of the mean motion used in approximating the Laplacian. When estimating the mean, it is important to include the fact that measurements further away from patch i are less reliable, than those closer to patch i. Therefore the covariances S_i are spread according to the distance from patch i. Thus,

$$(\bar{S}_i^k)^{-1} = \sum_{j \in \mathcal{N}(i)} (f(S_j^k, d_{ij}))^{-1}, \tag{13}$$

$$\bar{\mathbf{d}}_i^k = \bar{S}_i^k \sum_{j \in \mathcal{N}(i)} (f(S_j^k, d_{ij}))^{-1} \mathbf{d}_j'^k, \tag{14}$$

where $\mathbf{d}_j'^k$ is the estimated motion at patch i given than it moves rigidly with respect to the motion of patch j, and $S' = f(S, d)$ is function which spreads the covariance S according

to the distance d_{ij} between patches i and j. It is at this stage that an estimation of the local rotation at each patch becomes advantageous, since the effect of the rotation at j can be included in the predicted translation component of $\mathbf{d}_j'^k$. Through experimentation, it has been found that adding noise according to the square of the distance provides a sufficient model:

$$S' = S + \alpha \mathrm{I}(d_{ij}^2) \tag{15}$$

where I is the identity matrix, and α is the coefficient controlling how much the covariance S is spread. A remaining question to be addressed is how the measurement noise M should be estimated. Intuitively, one would expect the pixel gradients in $(\nabla J)^T \nabla J$ to be inversely proportional to the measurement noise, since a small gradient is more likely to originate from noise rather than a true image gradient, i.e.

$$M \propto ((\nabla J)^T \nabla J)^{-1}. \tag{16}$$

Using some example patches from the 'Katy' sequence, figure 2 shows the estimated measurement covariances for the three translation components. Three patches have much larger uncertainties than the other patches, and it is noticable that these three patches contain much less texture information than other patches. It is also noticeable that the ellipsoids along the arms are aligned along the arm, indicating that these patches are likely to slide up and down the arm, unless some form of regularization is introduced. The advantage of the approach being presented here is clear to see: the patches with more texture information should be used to constrain the motion of patches with less texture information.

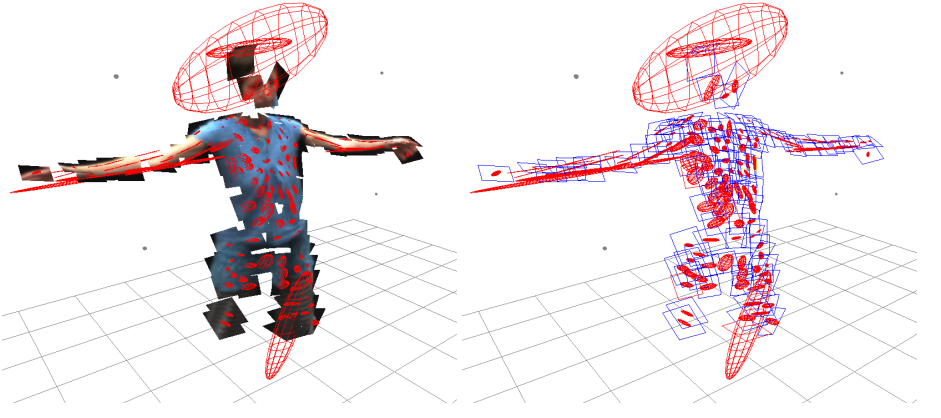


Figure 2: Ellipsoids representing the uncertainty in position for a set of example patches from the Katy sequence. The covariances are estimated using equation (16).

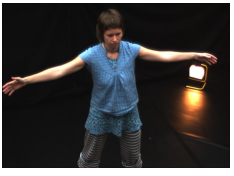
3.4 Motion Field Estimation

Although it is possible that the raw tracked patches could be used for a final application such as video-based rendering [13], it is more likely that the method presented here will be used as tool for applications such as mesh animation [6]. To this end, it is useful to provide motion

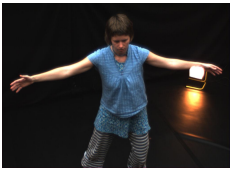
field estimates by interpolating between the motion of neighbouring patches. The 3D motion $(u,v,w,\theta_u,\theta_v,\theta_w)^T$ at the point $\mathbf{a}=(x,y,z)^T$ is given by:

$$\mathbf{d_a} = \Big(\sum_{j\in\mathcal{N}} \big(f(S_j,d_{ij})\big)^{-1}\Big)^{-1} \sum_{j\in\mathcal{N}} \big(f(S_j,d_{ij})\big)^{-1}\mathbf{d}'_j \tag{17}$$

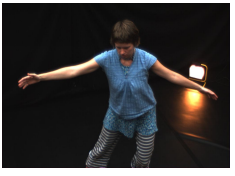
where $\mathbf{d}'_j$ is the predicted motion at the point $\mathbf{a}$ using the motion of patch j, S_j is covariance associated with the motion of patch j, and $f()$ is the spreading function defined in equation 16.



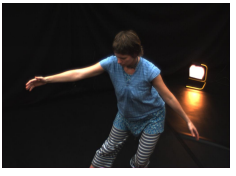
(a) Frame 0



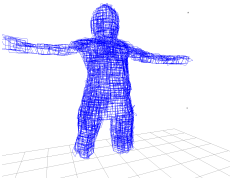
(b) Frame 30



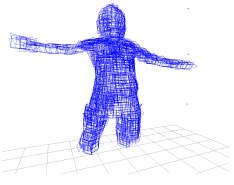
(c) Frame 60



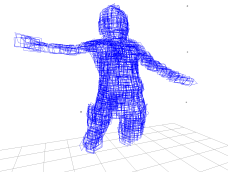
(d) Frame 90



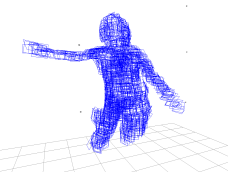
(e) Frame 0



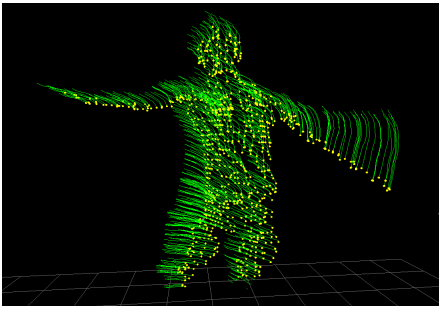
(f) Frame 30



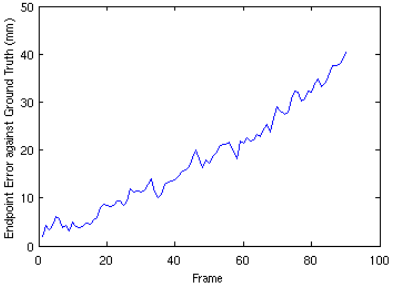
(g) Frame 60



(h) Frame 90



(i) Patch Centres, frames 20 to 70



(j)

Figure 3: (a)-(d) Input images at frames $\{0,30,60,90\}$ of 'Katy' sequence (e) Patches fitted at frame 0 (f)-(h) Tracked patches at frames $\{30,60,90\}$ (i) Tracked patch centres between frames 20 and 70 (j) Average endpoint motion estimation error.

4 Experimental Results and Discussion

4.1 Qualitative Evaluation

The results of the proposed method are demonstrated on two sequences taken in different performance capture studios. The 'Katy' sequence is 90 frames long and was captured at 30 frames per second using 32 cameras. The 'Skirt' sequence was captured at 40 frames per second using 8 cameras, and is part of a publicly available dataset [8]. The two sequences present different challenges to the scene tracker: the 'Katy' sequence contains surfaces with no texture but strong changes in lighting conditions; the 'Skirt' sequence is captured with fewer cameras and contains faster motion than the 'Katy' sequence.

As noted by Vedula *et al*. [18], it is important to ensure that cameras which do not see a particular surface point are not used for scene-flow estimates. Therefore a simple visual hull [11] and OpenGL depth buffering was used to provide a list of visible cameras at each frame for each patch. For the case that the list of visible cameras was empty for a particular patch, tracking was terminated for the patch. It is however possible that 'unseen' patches could be kept for the purposes of regularization, however this is left a topic for further research.

Figures 3(a-d) show views of the 'Katy' sequence from one camera at 30-frame intervals. Note that this sequence contains surfaces with little texture and changes in lighting conditions. Figure 3(e) shows the 1400 patches fitted to the scene surfaces at frame 0, figures 3(f-h) show the tracked patches at frames $\{30, 60, 90\}$ and figure 3(i) shows the trajectories of the patch centres between frames 20 and 70. The resulting tracked motion is smooth and coherent over the 90 frames. Note that even on the arms, where there is little texture, the motion of the surface is accurately tracked over the sequence. This shows the advantage of a regularized texture model approach over a feature-based one [7, 20], for which it would be impossible to estimate correspondences in non-textured regions.

Figures 4(a-d) show the views of the 'Skirt' sequence at frames $\{60, 70, 80, 90\}$, and figures 4(e-h) show the patches which are fitted once at frame 60 and then tracked up to frame 90. Figures 4(i-l) show a possible application of the tracker: animation of a visual hull mesh model.

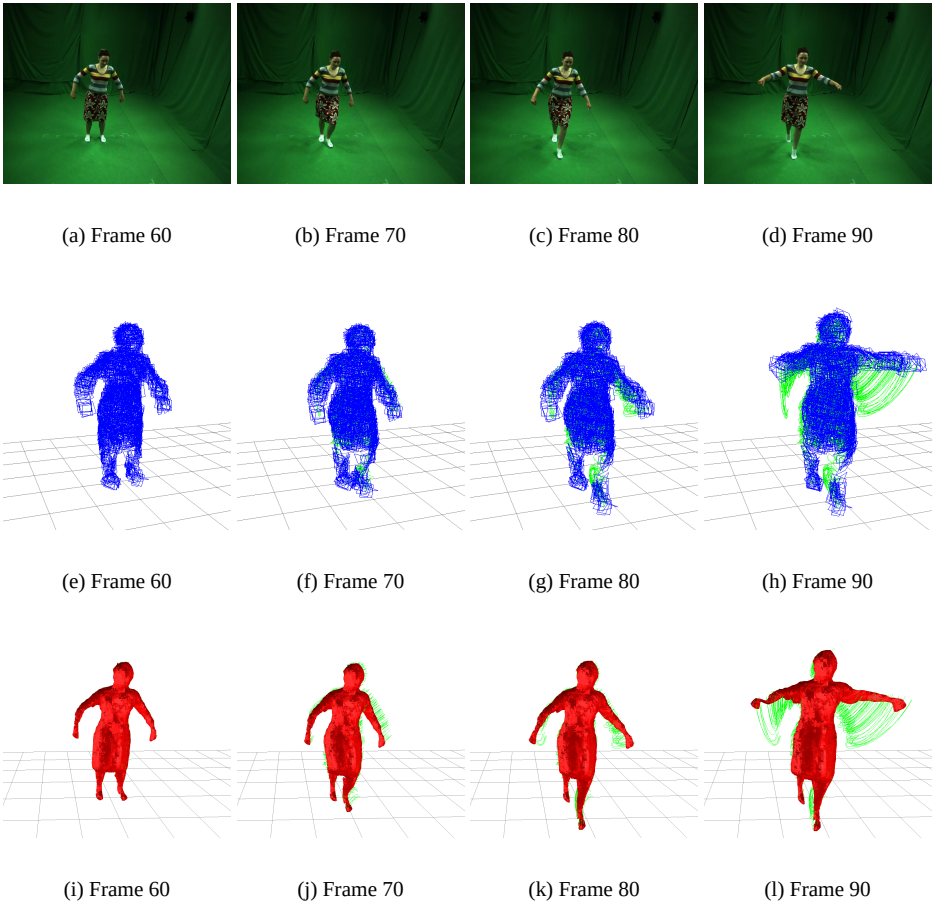
4.2 Numerical Evaluation

In order to assess the accuracy of the tracker, a simple GUI tool was constructed to allow a user to manually input 2D positions of a feature over time from two different views, so that ground truth tracks could be established. In order to ensure reliability, ground truth tracks were only accepted if the triangulation error between the two views was less than 6mm.

The motion field from section 3.4 was used to animate the ground truth points from their initial positions at the first frame to the final frame. Figure 3(j) shows the average error between the estimated motion and the ground truth motion on the Katy sequence. As shown in the figure, the final average error after 90 frames is around 40mm, and this error is steadily accumulated throughout the sequence.

4.3 Discussion

A direct comparison with different scene-flow algorithms is difficult, however it is apparent the the proposed method is clearly able to estimate the scene-flow over multiple frames rather than a single frame [4, 14, 18, 19]. Since the proposed method imposes the motion prior on



(a) Frame 60 (b) Frame 70 (c) Frame 80 (d) Frame 90

(e) Frame 60 (f) Frame 70 (g) Frame 80 (h) Frame 90

(i) Frame 60 (j) Frame 70 (k) Frame 80 (l) Frame 90

Figure 4: (a)-(d) Input images at frames $\{60, 70, 80, 90\}$ of 'Skirt' sequence (e) Patches fitted at frame 60 (f)-(h) Tracked patches at frames $\{70, 80, 90\}$ (i-l) A visual hull mesh animated from frame 60 to frames $\{70, 80, 90\}$.

the scene surfaces rather than in the images, it does not suffer from the common optical-flow problem of smoothening across object boundaries. It is also worth noting that the proposed method has been demonstrated on a sequence containing regions with very little texture, whilst most scene-flow sequences tend to be highly textured.

Obviously the proposed method does not generate temporally consistent meshes over long sequences, however it is worth noting that many of these techniques are only able to achieve the visually superior results with a multi-view reconstruction at every frame [3, 5, 14]. The proposed method does not need a reconstruction at every frame, and is able to provide surface point tracks across time providing full motion information.

Finally, a comparison with feature-based mesh-matching techniques is also helpful. Most feature-based methods usually produce less than 100 correspondences across 20 frames [7, 20], however the proposed approach provides more than 1000 correspondences over a longer

time period. The advantage of the proposed method is that it can incorporate information from regions which contain no features but do contain some information to constrain the motion of the scene (see figure 2).

5 Conclusions

A patch-based scene-flow estimation method has been proposed which is able to estimate the motion over several frames. The proposed method makes use of a patch-based representation that allows both translations and rotations to be estimated at each point on the surface, increasing the effectiveness of the local rigid motion prior. In addition, the novel neighbourhood prior takes account of measurement uncertainties by attaching a measurement covariance to each patch, which helps when propagating information to neighbouring regions. The main topic for further investigation is the inclusion of other priors such as: preservation of geodesic distances [17] and/or multi-view stereo constraints.

References

[1] S. Baker, D. Scharstein, JP Lewis, S. Roth, M.J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[2] M. Botsch and O. Sorkine. On Linear Variational Surface Deformation Methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, 2008.

[3] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. *ACM Transactions on Graphics*, 27(3):99, 2008.

[4] R.L. Carceroni and K.N. Kutulakos. Multi-View Scene Capture by Surfel Sampling: From Video Streams to Non-Rigid 3D Motion, Shape and Reflectance. *International Journal of Computer Vision*, 49(2):175–214, 2002.

[5] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *International Conference on Computer Graphics and Interactive Techniques*. ACM New York, NY, USA, 2008.

[6] Edilson de Aguiar, Christian Theobalt, Carsten Stoll, and Hans-Peter Seidel. Markerless deformable mesh tracking for human shape and motion capture. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Minneapolis, USA, June 2007. IEEE, IEEE.

[7] A. Doshi, A. Hilton, and J. Starck. An empirical study of non-rigid surface feature matching. In *Visual Media Production (CVMP 2008), 5th European Conference on*, pages 1–10, 2008.

[8] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1753, 2009.

[9] M. Habbecke and L. Kobbelt. A Surface-Growing Approach to Multi-View Stereo Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[10] B.K.P. Horn and B.G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17: 185–203, 1981.

[11] A. Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[12] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3): 90–126, 2006.

[13] A. Mullins, A. Bowen, R. Wilson, and N. Rajpoot. Video Based Rendering using Surfaces Patches. *3DTV Conference, 2007*, pages 1–4, 2007.

[14] J.P. Pons, R. Keriven, and O. Faugeras. Multi-View Stereo Reconstruction and Scene Flow Estimation with a Global Image-Based Matching Score. *International Journal of Computer Vision*, 72(2):179–193, 2007.

[15] T. Popham and R. Wilson. Selecting Surface Features for Accurate Surface Reconstruction. In *British Machine Vision Conference*, 2009.

[16] J. Starck and A. Hilton. Model-based multiple view reconstruction of people. In *IEEE International Conference on Computer Vision*, pages 915–922, 2003.

[17] J. Starck and A. Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[18] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 475–480, 2005.

[19] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. In *European Conference on Computer Vision*, pages 739–751, 2008.

[20] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. *IEEE Computer Vision and Pattern Recognition*, 0:373–380, 2009.