Joint Reconstruction of 3D Shape and Non-rigid Motion in a Region-Growing Framework

Ye Liu and Yan Qiu Chen
School of Computer Science
Fudan University, Shanghai, China
{10110240029 chenyq}@fudan.edu.cn

Abstract

This paper addresses the problem of simultaneously estimation of 3D structure and motion of moving surfaces from multi-view sequences. We model the surface locally as a 9-parameter spatio-temporal plane and a region growing mechanism is used to guarantee proper initial values are provided for parameter estimation. The algorithm starts with feature points that have been matched across views and time, and dense structure and motion then grow from these points. As demonstrated in the experiment, our method is able to handle large motion , topology change and frequent self-occlusion and the resulted 3D structure and motion are dense and accurate which is particularly useful for applications that require precise quantitative analysis of 3D motion.

1. Introduction

It is still a challenging task to recover the 3D motion of deforming surfaces from multiple calibrated cameras. This kind of motion can be non-rigid and quite complex, precise measurement of such motion is difficult and its potential applications include human face expression and human performance acquisition, animation, games and even physical science.

A commercialized method is to fix multiple markers onto the object's surface[13, 24], and then track these markers. This kind of method fails to get dense motion field due to the limited number of markers. Recent years, mesh-tracking methods[4, 2, 19] have become popular, they assume a known 3D model (usually triangulated mesh) in the first frame, and then track this model with various cues. It is even possible to track objects in monocular case[16]. The fixed topology nature of triangulated mesh greatly facilitates the long-sequence tracking process, but it is also a drawback when the topology of the scene changes. Besides, accurate 3D model is difficult to obtain, and the noise of the 3D model will be introduced to the estimated motion.

Thus joint estimation of 3D structure and motion seems a better alternative, and there has been a number of methods[12, 1], among which variational scene flow methods[9, 23, 20] have been successful recently. But they are not able to handle relatively large motion, and most of them only consider the binocular case.

We propose a method that can simultaneously recover 3D structure and motion of deforming surfaces accurately from image sequences of multiple views. Starting from a set of feature points that have been successfully matched across views and time, this method then simultaneously estimates 3D position and motion of dense sample points on the surface in a 9-parameter non-linear local plane fitting. Points that have been successfully matched will guide those nearby unmatched points by providing reasonable initial values until all the sample points on the surface are matched. As illustrated later, this method is based on a reasonable assumption, and it handles self-occlusion, topology change and large motion naturally. It does not require silhouette information or low-level cues such as optical flow and stereo matching which are used in many 4D methods. As demonstrated in the experiments, our method is able to recover accurate, dense 3D structure and motion field.

The rest of the paper is organized as follows. The next section will summarize work related to this paper. Section 2 presents the detailed techniques used in our method including the local spatio-temporal model, feature points matching, region growing and some other details. Section 3 gives the experiment results on both synthetic and real datasets. Section 4 discusses the method. And finally, section 5 concludes the paper.

1.1. Related Works

Reconstructing 3D shape from multi-view images is a fundamental problem in computer vision which has been studied for many years. Various methods have been proposed to solve this problem[17]. Recent years, several methods[3, 7, 5] based on the idea of region growing have

been very successful. They start by finding sparse correspondences as seeds for region growing, then the matched points can guide those nearby points that are to be matched until the whole surface is reconstructed. Our method extends the basic idea of region growing but addresses the different problem of simultaneously estimation of 3D shape and motion.

Among those marker based methods, White *et al*. [24] captured the shape and motion of folded cloth for animation using cloth that was printed with known pattern (color triangles). But their motion field is still not dense enough for some applications and it is not always possible to print pattern on arbitrary surfaces, so mark-less methods are necessary.

Among the recent mesh-tracking methods, Aguiar *et al*. [2] tracked a laser-scanned 3D model based on optical flow. Another mesh deformation method was proposed in [4], they estimated the positions of verticals of a triangulated mesh by multiple non-linear optimizations. These two methods are not able to handle topology changes, and accurate 3D model of the first frame is usually difficult to obtain. In [21], a mesh-tracking method that can handle large motion and topology change was proposed by Varanasi *et al*. They track feature points extracted from both the image and the 3D model and then densify the motion field using laplacian diffusion, which is actually a kind of interpolation, thus this method may not guarantee accurate motion field.

Some earlier work explored the spatio-temporal representation of moving surfaces, in [12], a multi-resolution subdivision surface was fit in order to simultaneously recover 3D structure and motion field. In [1], a method was proposed which divide the surface into a number of sufels, and estimate the normal, motion and reflectance of each sufel. There is no other mechanism in their method that make sure it recovers accurate and dense structure and motion. Vedula *et al*. [22] calculates shape and motion by carving a 6D volume, but due to the computational complexity, the resolution of shape and motion is relatively low. There is also method that uses only silhouette information [6] which is able to handle weakly textured surfaces, but it may not be suitable for precise use.

Some early work exploited the relationship between optical flow and scene flow [25, 18]. Variational methods have achieved good results. An earlier work in [15] solved the problem by minimizing an image-based matching score in a level set framework. Several recent works [9, 23, 20] have achieved precise results on their synthetic datasets. But they did not consider the multi-view case which is more complicate than the binocular case. We have tested our method on their synthetic dataset and compared with their results in the our experiment.

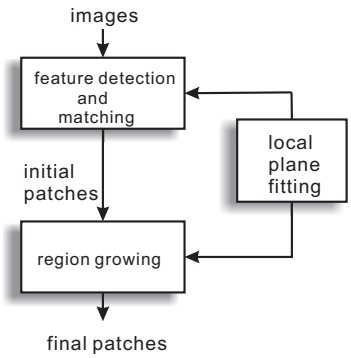


Figure 1. The stages of the proposed method. The local plane model is for both the feature matching and region stages.

2. Method

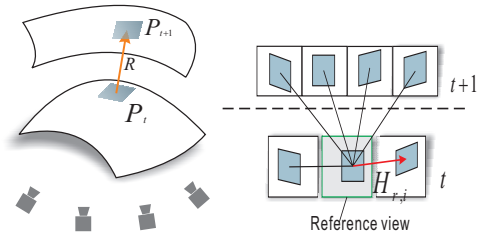
2.1. Method Overview

One major reason why region growing methods in multi-view stereopsis are successful is that they made reasonable assumption about the real world surfaces. If two points on an object's surface are close to each other, their depth and normal are also close to each other. So on the corresponding projected images, points that have been successfully matched provide good initial values of shape (depth, normal) for the nearby unmatched points. This assumption can be further extended to objects in the scene deforming over time: if two points on the surface at time t are close to each other, they are also close in time t+1. This means those points with known depth, normal and motion can provides good initial values for depth, normal and motion of neighboring points.

Region growing requires some stable initial points which are successfully matched across views and time as input. We use SIFT[11] in our method because it's scale and rotation invariant and relatively insensitive to illumination change. The feature points contain information about the local and global structure as well as motion of the surface, how to benefit from them will be deeply discussed in this paper. The proposed method contains roughly three stages (See Fig 1). A local plane model is proposed for both the feature matching and region growing stages.

2.2. Local Spatio-temporal Model

We locally fit a small patch of plane P_t centered at C_t on the surface at time t as showed in Figure1. If the cameras are calibrated, the transform induced by this plane between the images of two views i, j is a 3-dof homography $H_{i,j}$ [8]. The 3 parameters of this transform determine the plane uniquely. The plane can be written as $[p_1, p_2, p_3, 1]X = 0$. P_t is visible to N_t cameras, corresponding images are $\mathbf{V}_t = \{I_t^1, I_t^2, \ldots, I_t^{N_t}\}$. One of them I_t^r is used as reference image, then the first term of our ob-



(a) Local plane fitting. (b) Visible views at t and t+1

Figure 2. (a) A small patch undergoes a rigid transform from t to $t+1$. (b) One view at t is chosen as reference and sample points on it are fixed, with which projections on the other views at t and $t+1$ are compared.

jective function can be written as

$$\mathbf{E}_1 = \sum_{x_t^r \in \Omega} \sum_{I_t^i \in V_t - I_t^r} [I_t^r(x_t^r) - I_t^i(H_{r,i}(x_t^r))]^2, \quad (1)$$

Where x_t^r denotes the 2D coordinate on the reference image projected by a sample X_t on P_t. And Ω is the set of sample points on the patch. This function denotes the SSD (sum of square differences) of images (without reference image) that p_t projects at time t compared with the reference image. If we set x_t^r fixed, the homography $H_{r,i}$ becomes the only unknown, thus $\mathbf{E}_1$ is determined by three parameters $p1, p2, p3$.

The small patch mentioned above undergoes a rigid transform and becomes P_{t+1}. This transform has 6 degrees of freedom, and can be expressed as $\mathbf{R}(p4, p5, p6, p7, p8, p9)$, 3 of the 6 parameters are the rotation angles around the center C_t and the other 3 are translations. If P_{t+1} is visible to N_{t+1} views, and corresponding images are $\mathbf{V}_{t+1} = \{I_{t+1}^1, I_{t+1}^2, \ldots, I_{t+1}^{N_{t+1}}\}$, then the second term of our objective function can be written as

$$\mathbf{E}_2 = \sum_{x_t^r \in \Omega} \sum_{I_{t+1}^i \in V_{t+1}} [I_{t+1}^r(x_t^r) - I_{t+1}^i(x_{t+1}^i)]^2, \quad (2)$$

where

$$x_{t+1}^i = Pm_i \mathbf{R} X_t, \quad (3)$$

Pm_i is the projection matrix of camera i. With x_t fixed, X_t can be computed by intersecting the back-projected ray of x_t with P_t. $\mathbf{E}_2$ denotes the SSD of images that p_{t+1} projects at time $t+1$ compared with the reference image at time t.

Finally , the objective function is the weighted sum of the above two terms

$$\mathbf{E} = \mathbf{E}_1/N_t + \mathbf{E}_2/N_{t+1}. \quad (4)$$

Now the function has 9 parameters: $p_1, p_2, \ldots, p_9$, three of which are from the plane, and the rest six from the rigid transform. Minimizing this function is not easy, because it's non-linear and tends to fall into local minimums. Good initial values are important and necessary. A large part of our work presented in this paper focuses on providing initial values for the optimization process.

2.3. Feature Point Detection and Matching

We extract SIFT feature points along with their descriptors for every image at time t and $t+1$. Feature points from every two images at time t are matched according to their descriptor distances and epipolar constraint. If two feature points are successfully matched (the descriptors are close and satisfies the epipolar constraint), they will not be matched with other feature points. Matched points are triangulated to get 3D points $\mathcal{F}_t$, their descriptors are also kept. Feature points at time $t+1$ are also matched with the same strategy, and thus we get 3D points $\mathcal{F}_{t+1}$. Points in $\mathcal{F}_t$ and $\mathcal{F}_{t+1}$ are exhaustively matched with their descriptors, matched points in $\mathcal{F}_t$ are termed $\mathcal{M}$, which is a subset of $\mathcal{F}_t$. For every 3D point X_f in $\mathcal{M}$, we minimize objective function 4 to get the 9 optimum parameters. Initial values should be offered before optimization. We observe that the feature points, although sometimes sparse, contains not only some isolated 3D positions but also a coarse 3D shape of the object, so normal vector $n(X_f)$ at every point of $\mathcal{M}$ can be approximated by a weighted least square method [14]:

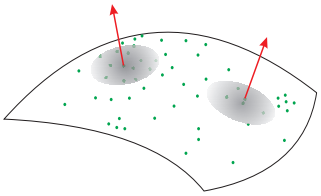


Figure 3. Normal vector of each feature point can be approximated by a weighted least square method. The weight decreases with the Euclidean distance between two points.

$$n(X_f) = \arg\max \sum_{X_f^i \in \mathcal{F}_t} \omega_i [n(X_f)^t (X_f^i - X_f)]^2, \quad (5)$$

where

$$\omega_i = \exp\{\frac{\|X_f^i - X_f\|}{h(X_f)}\}, \quad (6)$$

we choose $h(X_f)$ so that the 10th nearest neighbor has a weight of 0.1. As the gaussian function 6 decreases fast when distance increases, points far away from X_f will make little impact on the normal. And the denser the feature points are in the neighborhood, the closer the estimated normal is to the true value. Reasonable estimation is offered even when feature points are sparse. The initial values of $p1$, $p2$, $p3$ can be computed with 3D position X_f and the estimated normal vector.

Normal vector at t+1 can be estimated likewise, then we can compute a rigid transform (not unique) that turns the plane at time t to the plane at t+1. In this way all the nine parameters have their initial values.

Then X_f is projected to corresponding images, and one view is chosen as reference, A patch of μ by μ pixels centered at 2D point x_f^r is taken as Ω in Equation 1,2, and they are fixed during the optimization.

2.3.1 Visibility Update

Visibility at t and $t+1$ should be updated after optimization. Here we adopt normalized cross-correlation (NCC) as measurement, the sample points on P_t and P_{t+1} are projected to images at t and $t+1$, NCC is calculated between reference image and projected image. If NCC is above θ, we consider it is visible to this view. The patch is considered successfully optimized only when the numbers of visible views at both t and $t+1$ are above 2. And we can compute a matching score S as the mean of all the calculated NCC.

$$S = \frac{\sum_{i \in V_t} Ncc(r,i) + \sum_{j \in V_{t+1}} Ncc(r,j)}{N_t + N_{t+1} - 1}. \quad (7)$$

This matching score tells how successful the optimization is. The higher the score, the more we can trust the optimization result.

2.4. Region Growing

We adopt a region growing strategy similar to [3], but we update visibility both at time t and $t+1$. An important aspect of this kind of region growing is to give chance for optimization with different initial values to those points that have not been successfully matched, this has been discussed in [5]. Here, we use a priority queue which determines the priority according to the matching score S, After extracting and matching feature points, successfully matched patches were pushed into the queue. Each time, a patch is popped out from the queue as a seed, and then new patches grow from it and are pushed into the queue if they have been successfully optimized (satisfies the condition mentioned above).

Every visible view at t of the seed is searched to find pixels that have not been visited and are neighboring to the seed's projection on the image. Once such a pixel is found, a new patch is created. New patch's position is determined by intersecting the back-projected ray of the pixel and the plane p_t of the seed, and the nine parameters are inherited from the seed. Then the parameters are optimized, if successful, the new patch will be pushed into the priority queue. The region growing process terminates when the queue is empty.

The optimization can be carried out in every empty pixel, but in order to save computational time, it can be done every β pixel. It is a trade-off between the resolution of the result and computational time.

2.4.1 Coping with Appearance Change

SSD is sensitive to illumination and view point resulted appearance change. In order to cope with this, we estimate scalar factor r_t^i for the patch's pixel values of projection x_t^i and r_{t+1}^i for x_{t+1}^i, x_t^i and x_{t+1}^i in function 1 and 2 should be divided by their scalar:

$$I_t^i(x_t^i) = I_t^i(x_t^i)/r_t^i. \quad (8)$$

and

$$I_{t+1}^i(x_{t+1}^i) = I_{t+1}^i(x_{t+1}^i)/r_{t+1}^i. \quad (9)$$

We don't add this scalar as a new parameter to optimization, because this will increase computational complexity. Instead, we handle it the same way as visibility, first inherited from the seed before optimization, and updated after optimization. This is because both appearance difference and visibility change relatively more slowly than the 9 parameters and lag within some range is allowed.

2.4.2 Summary of Region Growing

Input : Initial priority queue $\mathcal{Q}$ with seeds from feature points.
Output: Collection $\mathcal{C}$ of patches that have been successfully optimized.
while $\mathcal{Q}$ *is not empty* **do**
  Pop a patch p from $\mathcal{Q}$, its visible views at t and $t+1$ are V_t, V_{t+1}
  foreach *image* I_i^t *in* V_t **do**
    Compute the projection of p on I_t^i as x_t^i.
    Find a neighboring pixel of x_t^i that has not been visited.
    if *such a pixel is found* **then**
      Create a new patch p_n.
      Set I_t^i the reference view of p_n.
      Inherit V_t,V_{t+1}, appearance scalar and 9 parameters from p.
      Optimize parameters.
      Update visibility.
      if *succeed (defined in section 2.4)* **then**
        Push p_n into $\mathcal{Q}$.
        Push p_n into $\mathcal{C}$.
      end
    end
  end
end

Algorithm 1: Overall algorithm for region growing.

There are three parameters in our method: patch size μ, NCC threshold θ and region growing interval β.

3. Experiments

We implemented our method with C++, and a Levenberg-Marquardt routine included in GSL was used for non-linear optimization (The Jacobian of the objective function must be provided). Optimization converges within several steps most of the time.

We list the parameters for each experiment.

dataset	μ	θ	β
ball	7	0.95	1
cloth	12	0.7	2
paint	12	0.7	2
face	12	0.7	2

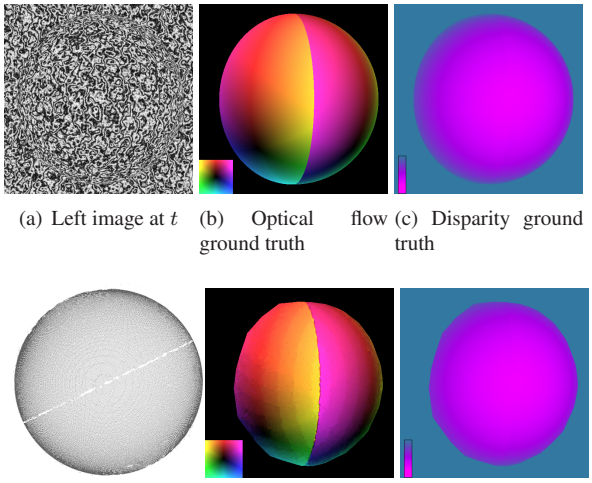
3.1. Synthetic Data

We tested our method on both synthesized data and real data. There are currently no real data with ground truth because this kind of ground truth is difficult to obtain. As a result, several existing methods used synthetic data for evaluation. We used a dataset synthesized by Huguet *et al.* which consists of four images of two hemispheres rotate in different directions at two time t and $t+1$. [9] and some other variational methods represent the scene flow as corresponding optical flow and disparity map, and our method computes scene flow in 3D space directly, So interpolation is required to convert our results to their format, which introduces bias. Table 1 gives the results before and after interpolation compared with [9, 23]. We can see that even after interpolation our method achieves a low optical flow RMS error $RMS_{u,v}$ and an average angular error AAE comparable to [9, 23], and an overall RMS error $RMS_{u,v,d'}$ much lower than the two methods. That's because the disparity map computed by our method is nearly the same with the ground truth (See Fig 4(f) and also the reconstructed 3d shape in Fig 4(d)). Our method handles occlusion

	$RMS_{u,v}$	$RMS_{u,v,d'}$	$AAE_{u,v}$
[9]	0.37	0.83	1.24
[23] with ground truth	0.33	0.58	1.25
[23] with Fill-SGM	0.43	0.75	2.18
Without interpolation	0.24	0.28	1.75
After interpolation	0.32	0.41	1.74

Table 1. Comparison of our method with two variational scene flow methods.

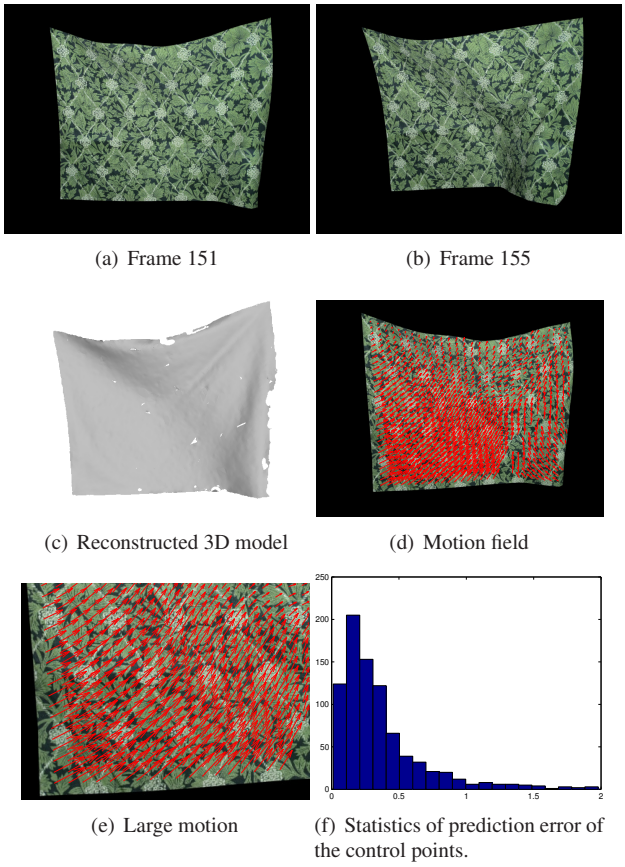
automatically, occluded area will have no data because of unsuccessful optimization. So all the results above do not include occluded areas. Because the synthetic process did not consider illumination change, if optimization succeeds the NCC score should be perfectly high, so we set a high NCC threshold (0.95), and the patch size μ was set to 7 in this experiment.



(a) Left image at t (b) Optical flow ground truth (c) Disparity ground truth

(d) Reconstructed 3D model (e) Optical flow (f) Disparity

Figure 4. (a) Left image of the synthetic ball dataset. (b-c) Optical flow and disparity ground truth encoded with pseudo-color. (d-f) Our results of 3D model, optical flow and disparity.



(a) Frame 151 (b) Frame 155

(c) Reconstructed 3D model (d) Motion field

(e) Large motion (f) Statistics of prediction error of the control points.

Figure 5. Images and results of the synthetic cloth dataset.

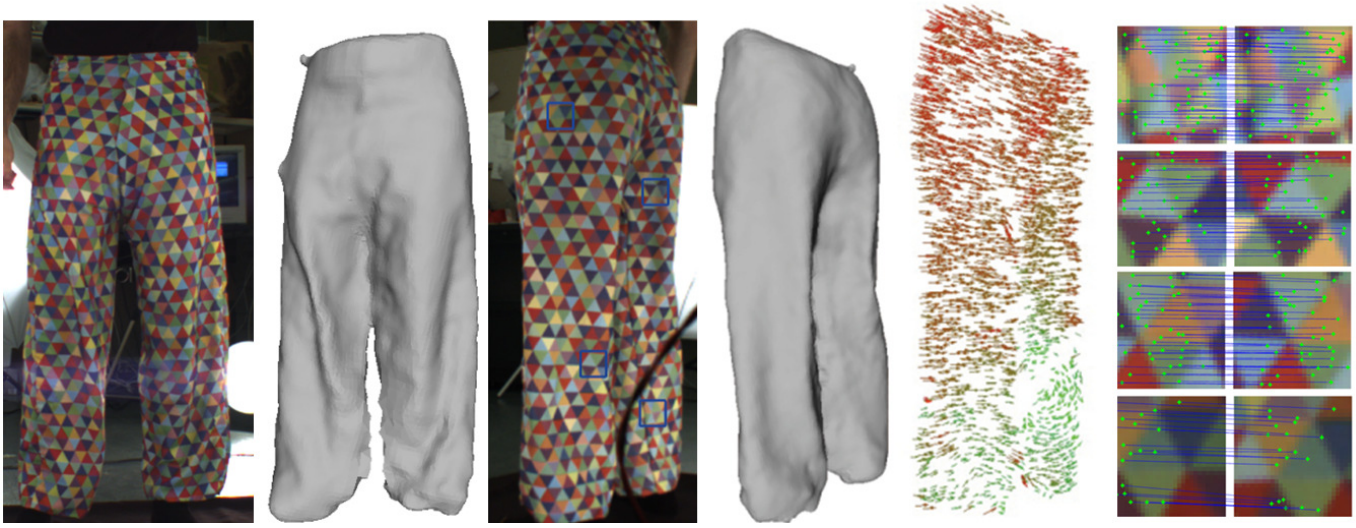


Figure 6. From left to right, input image at t of camera 8, our result of 3D model, image of camera 3, our 3D model, estimated scene flow (sampled and only those visible to camera 3 are displayed.), and close-ups of temporal correspondences in the four windows marked in the third image (only %50 of the points are displayed).

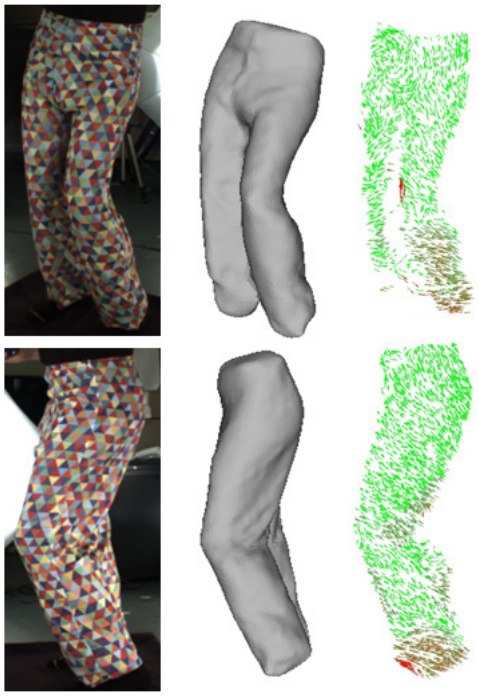


Figure 7. Paint frame 114, camera 8 and camera 3,our results of 3D model and motion field.

Another synthetic experiment was carried out to demonstrate our method's ability to handle large motion. We use synthetic data because the deformation can be controlled and we can compare our results with the ground truth. We simulate in 3DS MAX a piece of cloth with its top fixed waving in wind. An image of real cloth was mapped onto the 3D model, and two virtual cameras were set to capture the deformation of the cloth. The ground truth contains 900 control points whose 3D positions are known in every frame. We selected two frames 151 and 155, the images of camera 1 are showed in figure 5(a) and 5(b). Note the bottom left corner of the cloth deforms with large displacement, in fact the displacement is larger than 30 pixels. We give the close-up look of our result of estimated motion of the control points. Figure 5(f) shows the histogram of the prediction error, most of which is within 1 pixel.

3.2. Real Data

Two real datasets were used in our experiment. The paint dataset by White *et al*. [24] captures an actor wearing a pair of loose trousers dancing. Although the trousers are highly textured, its motion is complicate and self-occlusion frequently occurs. The result of frame 29 and 114 is showed in figure 6 and 7. We also give close-ups of temporal correspondence of frame 29 in figure 6, which are obtained by projecting 3D correspondence onto image. The face dataset was captured by us using two cameras. In order to capture accurate motion of human face, we randomly drew texture on the face. We show our result in figure 8. The final result of 3D models in the two experiment is reconstructed from obtained oriented point clouds using a Poisson surface reconstruction approach[10].

We selected two frames 0 and 6 from the face sequence, as there is topology change between these two frames. Note that the person's mouth changes from close at frame 0 to

open at frame 6. Mesh-tracking methods usually can not handle this change. As our method does not assume a fixed topology, it works even with topology change. We give the two 3D models at the two time obtained by our method and also closeup of temporal correspondences near the mouth in figure 9.

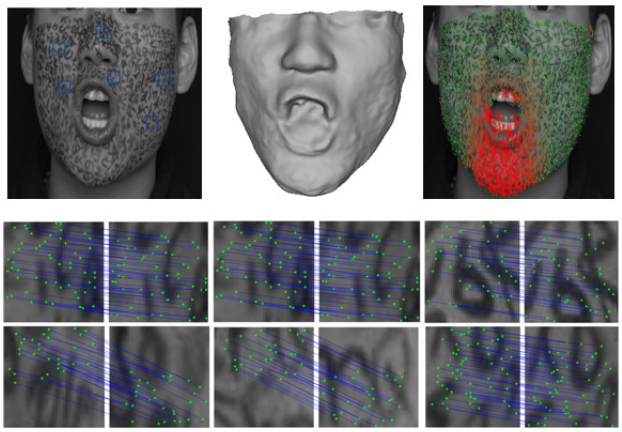


Figure 8. Top:an input image, reconstructed 3D model and motion field. Bottom: close-ups of temporal correspondences in the windows marked in the first image (only %20 of the points are displayed).

4. Discussions

1. The influence made by the number and distribution of matched feature points on final results depends on the complexity of surface's topology. If the topology is simple, the result will not be bad even with a few matched feature points. The feature points can be mismatched, but the outliers will have limited effect on the final result because new patches can not grow from those outliers.

2. The running time of the algorithm also depends on the complexity of surface. For example, the ball dataset takes only 12 min because the topology is simple though we set β to 1, while the paint dataset usually takes over 30 min even with β set to 2. The main computational cost is on the calculation of the Jacobian matrix for optimization. There is still plenty of space to speed up the algorithm.

3. The local rigid model can be replaced with more complicated models depending on different applications. For example, our model may not be suitable for modeling elastic surfaces and thus a model with more parameters can be used alternatively.

5. Conclusion

This paper present a new method that simultaneously recovers structure and motion from multi-view image sequences. This method is based on a reasonable assumption, and starts from sparse feature points. It does not require priori knowledge about the 3D shape and motion, which makes the method able to handle topology change and self-occlusion. Besides, because of the matched feature points, this method can cope with large motion. At the same time, the local optimization nature of the method and the region growing framework makes the reconstruction result accurate and dense.

As our method is still based on computing the gradient of image, it does not work well in weakly textured regions. This can be seen in the experiment, in figure 9, outliers still exist in some regions although we set a relative high NCC threshold 0.7. Our future work will mainly focus on this problem.

Acknowledgement

The research work presented in this paper is supported by National Natural Science Foundation of China, Grant Nos. 60875024 and 61175036; Education Commission of Shanghai Municipality Grant No. 10ZZ03; Science and Technology Commission of Shanghai Municipality, Grant No. 09JC1401500; and Shanghai Leading Academic Discipline Project, Project Number B114.

References

[1] R. Carceroni and K. 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. 1, 2

[2] E. De Aguiar, C. Theobalt, C. Stoll, and H. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 1, 2

[3] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 2009. 1, 4

[4] Y. Furukawa and J. Ponce. Dense 3d motion capture from synchronized video streams. *Image and Geometry Processing for 3-D Cinematography*, pages 193–211, 2010. 1, 2

[5] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. Seitz. Multi-view stereo for community photo collections. In *Proc. ICCV*, pages 1–8, 2007. 1, 4

[6] L. Guan, J. Franco, E. Boyer, and M. Pollefeys. Probabilistic 3D occupancy flow with latent silhouette cues. In *Computer Vision and Pattern Recognition, 2010 IEEE Conference on*, pages 1379–1386. IEEE, 2010. 2

[7] M. Habbecke and L. Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *Computer Vision and Pattern Recognition, 2007 IEEE Conference on*, pages 1–8. IEEE, 2007. 1

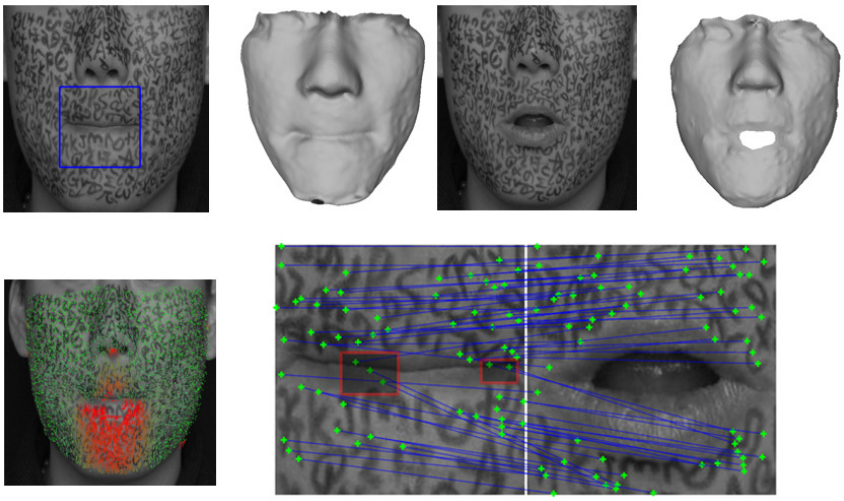


Figure 9. Top: from left to right,input image at frame 0 of camera 1, our result of 3D model of frame 0, image at frame 6 of camera 1, our 3D model of frame 6. Bottom: estimated scene flow, and close-up of temporal correspondences in the window marked in the first image above (only %1 of the points are displayed). Boxed points are outliers.

[8] R. Hartley and A. Zisserman. *Multiple view geometry*, volume 642. Cambridge university press Cambridge, UK, 2000. 2

[9] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *Computer Vision, 2007. The Proceedings of the Seventh IEEE International Conference on*, pages 1–7. IEEE, 2007. 1, 2, 5

[10] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 61–70. Eurographics Association, 2006. 6

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[12] J. Neumann and Y. Aloimonos. Spatio-temporal stereo using multi-resolution subdivision surfaces. *International Journal of Computer Vision*, 47(1):181–193, 2002. 1, 2

[13] S. Park and J. Hodgins. Capturing and animating skin deformation in human motion. *ACM Transactions on Graphics (TOG)*, 25(3):881–889, 2006. 1

[14] M. Pauly, L. Kobbelt, and M. Gross. Point-based multiscale surface representation. *ACM Transactions on Graphics (TOG)*, 25(2):177–193, 2006. 3

[15] J. 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. 2

[16] M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3-d tracking. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 1

[17] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Conference on*, volume 1, pages 519–528. IEEE, 2006. 1

[18] S. Simon, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-Dimensional Scene Flow. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999. 2

[19] T. Tung and T. Matsuyama. Dynamic surface matching by geodesic mapping for 3D animation transfer. In *Computer Vision and Pattern Recognition, 2010 IEEE Conference on*, pages 1402–1409. IEEE, 2010. 1

[20] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt. Joint Estimation of Motion, Structure and Geometry from Stereo Sequences. *Computer Vision–ECCV 2010*, pages 568–581, 2010. 1, 2

[21] K. Varanasi, A. Zaharescu, E. Boyer, and R. Horaud. Temporal surface tracking using mesh evolution. *Computer Vision–ECCV 2008*, pages 30–43, 2008. 2

[22] S. Vedula, S. Baker, S. Seitz, and T. Kanade. Shape and motion carving in 6D. In *Computer Vision and Pattern Recognition, 2000 IEEE Conference on*, volume 2, pages 592–598. IEEE, 2002. 2

[23] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. In *ECCV*, pages 739–751, 2008. 1, 2, 5

[24] R. White, K. Crane, and D. Forsyth. Capturing and animating occluded cloth. *ACM Transactions on Graphics (TOG)*, 26(3):34, 2007. 1, 2, 6

[25] Y. Zhang and C. Kambhamettu. On 3D scene flow and structure estimation. In *Computer Vision and Pattern Recognition, 2001 IEEE Conference on*, pages II–778 – II–785. IEEE, 2001. 2