Long-Range Motion Trajectories Extraction of Articulated Human Using Mesh Evolution

Yuanyuan Wu, Xiaohai He, Byeongkeun Kang, Haiying Song and Truong Q. Nguyen, Fellow, IEEE

Abstract—This letter presents a novel approach to extract reliable dense and long-range motion trajectories of articulated human in a video sequence. Compared with existing approaches that emphasize temporal consistency of each tracked point, we also consider the spatial structure of tracked points on the articulated human. We treat points as a set of vertices, and build a triangle mesh to join them in image space. The problem of extracting long-range motion trajectories is changed to the issue of consistency of mesh evolution over time. First, self-occlusion is detected by a novel mesh-based method and an adaptive motion estimation method is proposed to initialize mesh between successive frames. Furthermore, we propose an iterative algorithm to efficiently adjust vertices of mesh for a physically plausible deformation, which can meet the local rigidity of mesh and silhouette constraints. Finally, we compare the proposed method with the state-of-the-art methods on a set of challenging sequences. Evaluations demonstrate that our method achieves favorable performance in terms of both accuracy and integrity of extracted trajectories.

Index Terms—Motion trajectories, articulated motion, mesh evolution.

I. INTRODUCTION

LONG-range motion trajectories provide more precise and integrated information of a movement and have been extensively used in various applications such as action recognition, motion segmentation, video indexing and retrieval, and video manipulation. It is worth to note that only one camera is set in most of the applications, which leads to the loss of much visual information and brings many challenges. Sparse feature trackers such as KLT feature tracker [1] is often used to extract motion trajectories in video sequence. Moreover, spatially-denser trajectories can be obtained by PV tracker [2] and LDOF tracker [3]. PV tracker builds trajectories by sweeping forward and backward flow fields and also refines motion estimates to enforce long-range consistency. LDOF tracker is based on large displacement optical flow (LDOF) proposed by Brox et al. [4]. These trackers share one essential criterion that if points are lost possibly due to lighting variation, out of plane rotation, occluded or large displacement, then new points will be added. As a result, points in initial video frame may not be fully tracked throughout the video sequence. However, integrated long-range motion trajectories can be obtained by concatenating frame-to-frame optical flow motion fields, such as Lagrangian particle trajectories (LPT) used in action recognition work [5]. As discussed in [2], this class of algorithms may cause trajectories drift by error accumulation. In summary, it is challenging to extract both reliable and long-range motion trajectories throughout the whole video sequence.

Our approach is inspired by the work on dense surface tracking in [6], [7], which both formulate a mesh evolution framework including an iterative mesh deformation step. Differently, [6] performs surface-morphing while [7] provides local rigidity constraints of a surface in the iterative mesh deformation step. By introducing this mesh evolution framework from 3D space to 2D image plane, we extract long-range motion trajectories effectively. Specifically, self-occlusion is first detected by searching the mesh intersection. Next, vertices in the occlusion region and the non-occlusion region will receive specified motion estimations for propagating to the next frame. Last, vertices are gradually approaching to their actual positions by the iterative mesh deformation step, in which different types of drifted vertices are recognized and regularized, and the local rigidity of the mesh is enforced in an efficient way. In this letter, binary silhouettes of articulated human are utilized to recognize and regularize drifted vertices. Similar to several silhouette-based methods [8]–[11], the advantages of using silhouettes have been proven in various applications, e.g. human action, gait recognition, etc.. The extraction of silhouettes from a video commonly entails using techniques such as background subtraction. Fig. 1 shows an overview of the proposed long-range motion trajectories extraction method.

Our contribution with respect to methods [6], [7] is that the mesh evolution framework is proposed for monocular-camera set-up. Self-occlusion of object is one inevitable problem in single-view video, so we proposed an effect way to detect...
the occlusion region. Another problem is that the strategy of mapping after meshing is not applicable in single-view video due to the self-occlusion, so we proposed a strategy of propagating vertices with specified predicted motions. Moreover, some geometric information such as the perspective invariance of surface norms does not extend from surface to silhouette, so we proposed an efficient way to recognize and regularize drifted vertices in 2D. To the best of our knowledge, no previous work has attempted to perform the long-range propagating vertices with specified predicted motions. More-

II. PROPOSED trajectories extraction METHOD

The input to our system is a monocular video sequence of \( M \) frames. The stack of silhouettes \( \{ S^t \}, t \in \{ 1, \ldots, M \} \) is extracted and \( N \) tracked points are sampled uniformly on the reference silhouette \( S^1 \) by a mesh generator algorithm [12], [13]. Let 2-dimensional vector \( p^i_t \in \mathbb{R}^2 \) denote the position of a tracked point \( i \) in frame \( t \), then a big matrix \( A \) is constructed as follows:

\[
A = \begin{bmatrix}
T_1 & p_1^1 & p_2^1 & \cdots & p_M^1 \\
T_2 & p_1^2 & p_2^2 & \cdots & p_M^2 \\
& \vdots & \vdots & \ddots & \vdots \\
T_N & p_1^N & p_2^N & \cdots & p_M^N 
\end{bmatrix}
\]  

(1)

Note that each row of matrix \( A \) is a representation of one fully tracked trajectory \( T_i, i \in \{ 1, \ldots, N \} \). The objective of our approach is to extract a reliable set of long-range trajectories \( \{ T_i \} \). From an alternate point-of-view, \( N \) track points are physically belonging to a human undergoing articulated motion. Therefore, each column of matrix \( A \) is one instant pose of articulated human which is assumed to share the same topology. We consider a planar triangle mesh \( G^1(\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{P}^1) \) which represents a column of matrix \( A \). Let \( \mathcal{V} = \{ 1, \ldots, N \} \) be the set of vertices, \( \mathcal{E} = \{ (i, j), i, j \in \mathcal{V} \} \) be the set of edges, \( \mathcal{F} = \{ (i, j, k), i, j, k \in \mathcal{V} \} \) be the set of faces, \( \mathcal{P}^1 = \{ p_1^1, \ldots, p_N^1 \} \) be the set of vertices positions. We assume that all meshes \( \{ G^1 \} \) share the same topology \( (\mathcal{V}, \mathcal{E}, \mathcal{F}) \) but vary at vertex positions \( \mathcal{P}^1 \). Therefore, the trajectories extraction problem is casted as mesh evolution over time, i.e.

\[
G^1(\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{P}^1) \rightarrow G^1(\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{P}^t)
\]  

(2)

A. Self-Occlusion Detection

Self-occlusion is commonly occurring between moving torso and swinging limbs undergoing articulated motions. By taking the advantage of the deformed mesh, we detect the occlusion region by finding intersected edges of the mesh. As illustrated in Fig. 2(a), during the leg crossing motion, two components of mesh intersect in the occlusion region which is highlighted in red color. In computational geometry, this is a line segment intersection problem which supplies a list of line segments in the Euclidean plane and asks whether any two of them intersect. As illustrated in Fig. 2(b), suppose the two line segments run from \( p_1 \) to \( p_2 \) and from \( p_3 \) to \( p_4 \). Then any point on the first line is represented as \( p_1 + \alpha(p_2 - p_1) \) and similarly \( p_3 + \beta(p_4 - p_3) \) is for any point on the second line, where \( \alpha \) and \( \beta \) are scalar parameters. The two line segments intersect if we can find \( \alpha \) and \( \beta \) such that:

\[
p_1 + \alpha(p_2 - p_1) = p_3 + \beta(p_4 - p_3)
\]  

(3)

Cross both sides with \( p_4 - p_3 \) and \( p_2 - p_1 \) separately, solving for \( \alpha \) and \( \beta \):

\[
\alpha = \| (p_3 - p_1) \times (p_4 - p_3) \| / \| (p_2 - p_1) \times (p_4 - p_3) \| 
\]  

(4)

\[
\beta = \| (p_1 - p_3) \times (p_2 - p_1) \| / \| (p_4 - p_3) \times (p_2 - p_1) \| 
\]  

(5)

If the denominator \( \| (p_2 - p_1) \times (p_4 - p_3) \| \neq 0 \) as well as \( 0 < \alpha < 1 \) and \( 0 < \beta < 1 \), then two lines intersect. Therefore, intersected edges are found in the mesh and corresponding vertices are identified in occlusion region.

B. Initial Motion Estimation

In order to propagate mesh \( G^{t-1} \) to \( G^t \) in the next frame for a reliable initial guess, we propose to estimate the vertices of \( G^t \) through large displacement optical flow (LDOF) [4], polynomial curve fitting, and patch-based average filtering. LDOF as a recent successful optical flow method, particularly approach the problematic estimation of articulated human motion. However, it does not solve occlusion problem like other optical flow methods. Therefore, an adaptive method is proposed to estimate motion vectors of vertices of \( G^{t-1} \) in different image regions: For a vertex \( p_i^{t-1} \) in non-occlusion region, we perform bicubic spline interpolation of LDOF motion vectors to get the motion vector \( u_i^{t-1} \). For a vertex \( p_i^{t-1} \) in occlusion region, we perform a second-order polynomial curve fitting to construct vertex \( p_i^t \) within the range of a discrete set of previous five positions. Specifically, the fitting model is \( Y_i = B X_i \), where \( B = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix} \) is the unknown coefficients matrix, \( X_i \) and \( Y_i \) respectively are input and output matrices, i.e. \( X_i = [x_{i-1} \ x_{i-2} \ x_{i-3} \ x_{i-4} \ x_{i-5}]^T \), \( Y_i = [p_i^{t-1} \ p_i^{t-2} \ p_i^{t-3} \ p_i^{t-4} \ p_i^{t-5}]^T \). Therefore, the solution of coefficients matrix is \( B = Y_i X_i^T \) and the estimated motion vector is

\[
u_i^{t-1} = B X_i - p_i^{t-1}
\]  

(6)

Moreover, in order to handle the observation noise, we apply a patch-based average filter to obtain smoothing result of motion vectors. Here, a patch is denoted as the set of vertex \( i \) and its adjacent vertices, i.e. \( N(i) = \{ i \} \cup \{ j : (i, j) \in E^i \} \). |\( N(i) |
expressed as a set of pixel points $Q_i$. If a subset $Q_i \subset Q$ is within the unit area of a vertex $i$, we denote the vertex $i$ as the first type of drifted vertices ($\mathcal{V}_1$), and will predict its target position from the pixel points in $Q_i$. As shown in Fig. 3(d), if a vertex is beyond the range of silhouette, we denote it as the second type of drifted vertices ($\mathcal{V}_2$) and predict its target position from support adjacent vertices which are denoted as $N_i = N(i) \cap (\mathcal{V} \setminus \mathcal{V}_2)$. Note that a potential issue could occur where a patch of vertices are all second type of drifted vertices, that is, $N(i) \subset \mathcal{V}_2$ and the set $N_i = \text{null}$. Therefore, we predict the target positions for the second type of drifted vertices in a batch process. The predicted batch of vertices will be removed from set $\mathcal{V}_2$, and keep predicting left vertices until $\mathcal{V}_2$ is empty. We can finally regularize the target position as follows:

$$p^i_{t,\text{Reg}}(k) = \left\{ \begin{array}{ll} \lambda \hat{p}^i_t(k - 1) + (1 - \lambda) \frac{1}{|Q_i|} \sum_{q_j \in Q_i} q_j & \text{if } i \in \mathcal{V}_1 \\ \lambda \hat{p}^i_t(k - 1) + (1 - \lambda) \frac{1}{|N_i|} \sum_{j \in N_i} \hat{p}^i_j(k - 1) & \text{if } i \in \mathcal{V}_2 \\ \hat{p}^i_t(k - 1) & \text{else} \end{array} \right. $$

Here, $|Q_i|$ and $|N_i|$ are the number of elements of set $Q_i$ and $N_i$ respectively. The $\lambda$ term balances the influence of original point and points in support domain; controls the regularization pace. In practice $\lambda = 2/3$ was used for all experiments. Fig. 3(e) and 3(f) show the results of mesh regularization.

2) Local Rigid Deformation: To preserve the local rigidity of the deformed mesh, we map the patches to a global coordinate system via per-patch rigid transformations, here the rigid transformation is equivalent to an affine transformation in 2D image plane. As described in simulation (2), we would like to compute the rigid transformation of a reference patch in $\mathcal{P}^1$ to best conform to the corresponding patch in $\mathcal{P}^t$, such that:

$$(R_i, T_i) \leftarrow \arg \min_{j \in N(i)} \left\| p^i_{t,\text{Reg}}(k) - (R_j p^i_j + T_j) \right\|^2 $$

where $R_i$ is the $2 \times 2$ rigid transformation matrix and $T_i$ is the translation vector. This is an instance of procustes problem, which can be solved by procustes analysis [14]. Instead of simply using the rigid transformation of patch $N(i)$, we also consider the rigid transformations from the neighboring patches $\{N(j)\}, j \in N(i)$. This procedure preserves the local rigidity of mesh deformation better. The vertex position is defined as:

$$p^i_{t,\text{RD}}(k) = \frac{1}{|N(i)|} \sum_{j \in N(i)} (R_j p^i_j + T_j) $$

After the mesh regularization and local rigid deformation of mesh, one iteration ends and the next iteration begins with the updated position, i.e. $\hat{p}^i_t(k) = p^i_{t,\text{RD}}(k)$. The iteration stops when satisfy the stopping criteria in equation 9.

III. EXPERIMENTS

A. Datasets and Baselines

To evaluate the efficiency of the proposed method, five challenging sequences from [15], [16] and Weizmann Human...
Action Dataset [17] are used. The challenges of these videos include pose change, self-occlusion, rapid movement, and scale variation. Our method is also compared with some state-of-the-art motion trajectories extraction algorithms including KLT tracker [1], PV tracker [2], LDOF tracker [3] and LPT [5]. Their source codes are provided by the authors and the parameters are tuned to achieve the best results.

B. Long-Range Motion Trajectories Extraction

Fig. 4 illustrates the epitome of five sequences and the extracted long-range motion trajectories (the longest motion trajectory in time is 141 frames from the Skirt sequence). Each sequence has its own characteristics. In the sequence Walking, lightly foreshortening and self-occlusion have occurred when the woman moved her left leg diagonal backward followed by her right leg moving. The sequences Wheeling and Handstanding recorded a complete wheeling action and hand standing action respectively, fast movement and out-of-plane rotation are the main challenges. The sequence Dancing contained complex pose change, foreshortening and self-occlusion. In sequence Skirt, the women moved forward with her arms lift and then turned sideways, undergoing scale variation and out-of-plane rotation. The proposed method achieved robust performance over these challenging sequences.

We also test our approach on the Weizmann Human Action Dataset [17], and some of the results are shown in Fig. 7. The visual results can be found in our project website http://videoprocessing.ucsd.edu/~yuanyuan/trajectories.html.

C. Performance Comparison

To evaluate the accuracy of extracted motion trajectories by the state-of-the-art methods and the proposed method, we illustrate the visual comparisons in Fig. 5, where self-occlusion and fast movement happens in sequence Walking and sequence Wheeling. It is observed that an abundance of points on the leg drifted away or stopped tracking due to self-occlusion and fast movement when using other four methods while the proposed method tracked dense points accurately.

In this paper, the percentage of tracking length in time is computed to evaluate the integrity of extracted motion trajectories. From Table I we can observe that the average percentage of tracking length in time by KLT, PV, LDOF algorithms are less than 100%, that means these algorithms can not continually track dense points throughout all the five sequences. In contrast, integrated trajectories are obtained by LPT and the proposed method.

To further evaluate the accuracy of integrated motion trajectories extracted by LPT and the proposed method, we compute the tracking error based on the provided benchmarks of joint center positions in every frame [15], [16]. Fig. 6 presents the standard deviation of the offset distances in every frame of five sequences. It is observed that the proposed method outperforms LPT with smaller value of the standard deviation of offset distances. It is worth to point out that taking advantages of silhouettes may be the main reason that makes the proposed method superior to LPT. Silhouette constraints play an important role in recognizing and regularizing drifted vertices, therefore avoiding the accumulation of errors during the tracking.

IV. CONCLUSION

This letter presents a novel effective and reliable long-range motion trajectories extraction method based on mesh evolution and silhouette constraints. Experiments on challenging video sequences show that the proposed method guarantees the
integrity and accuracy of dense points tracking and performs better than several state-of-the-art methods. Since the proposed method is applicable to partial occlusion not full occlusion, it is limited to some challenge actions like spinning around and severe shape deformation. The proposed method is suitable for applications where accuracy of the motion estimation is vital.

REFERENCES

[1] Jianbo Shi and Carlo Tomasi, “Good features to track,” in Computer Vision and Pattern Recognition. 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on. IEEE, 1994, pp. 593–600.

[2] Peter Sand and Seth Teller, “Particle video: Long-range motion estimation using point trajectories,” International Journal of Computer Vision, vol. 80, no. 1, pp. 72–91, 2008.

[3] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer, “Dense point trajectories by gpu-accelerated large displacement optical flow,” in Computer Vision–ECCV 2010, pp. 438–451. Springer, 2010.

[4] Thomas Brox, Christoph Bregler, and Jitendra Malik, “Large displacement optical flow,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 41–48.

[5] Shandong Wu, Omar Oreifej, and Muharar Shah, “Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories,” in Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 1419–1426.

[6] Kiran Varanasi, Andrei Zaharescu, Edmond Boyer, and Radu Horaud, “Temporal surface tracking using mesh evolution,” in Computer Vision–ECCV 2008, pp. 30–43. Springer, 2008.

[7] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic, “Iterative mesh deformation for dense surface tracking,” in Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. IEEE, 2009, pp. 1465–1472.

[8] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri, “Actions as space-time shapes,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 29, no. 12, pp. 2247–2253, 2007.

[9] Sruti Das Choudhury and Tardi Tjahjadi, “Silhouette-based gait recognition using procrustes shape analysis and elliptic fourier descriptors,” Pattern Recognition, vol. 45, no. 9, pp. 3414–3426, 2012.

[10] Mohamed F Abdelkader, Wael Abd-Almageed, Anuj Srivastava, and Rama Chellappa, “Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds,” Computer Vision and Image Understanding, vol. 115, no. 3, pp. 439–455, 2011.

[11] Alexander Andre Chaaraoui, Pau Climent-Pérez, and Francisco Florez-Revuelta, “Silhouette-based human action recognition using sequences of key poses,” Pattern Recognition Letters, vol. 34, no. 15, pp. 1799–1807, 2013.

[12] Per-Olof Persson and Gilbert Strang, “A simple mesh generator in matlab,” SIAM review, vol. 46, no. 2, pp. 329–345, 2004.

[13] Gabriel Peyré, “The numerical tours of signal processing,” Computing in Science & Engineering, vol. 13, no. 4, pp. 94–97, 2011.

[14] John C Gower and Garritt B Dijksterhuis, Procrustes problems, vol. 3, Oxford University Press Oxford, 2004.

[15] Juergen Gall, Carsten Stoll, Edilson De Aguiar, Christian Theobalt, Bodo Rosenhahn, and H-P Seidel, “Motion capture using joint skeleton tracking and surface estimation,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 1746–1753.

[16] Ping Wang and James M Rehg, “A modular approach to the analysis and evaluation of particle filters for figure tracking,” in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. IEEE, 2006, vol. 1, pp. 790–797.

[17] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri, “Actions as space-time shapes,” in The Tenth IEEE International Conference on Computer Vision (ICCV’05), 2005, pp. 1395–1402.