Dynamic Human Surface Reconstruction Using a Single Kinect

Ming Zeng† Jiaxiang Zheng‡ Xuan Cheng‡ Bo Jiang‡ Xinguo Liu‡

†*Software School of Xiamen University*
Xiamen, China
Email:mingzeng85@gmail.com

‡*State Key Lab of CAD&CG, Zhejiang University*
Hangzhou, China
Email:xgliu@cad.zju.edu.cn

***Abstract*—This paper presents a system for robust dynamic human surface reconstruction using a single Kinect. The single Kinect provides a self-occluded and noisy RGBD data. Thus it is challenging to track the whole human surface robustly. To overcome both incompleteness and data noise, we adopt a template to confine shape in the un-seen part, and propose a two-stage tracking pipeline. The first stage tracks articulated motion of human, which improves robustness of tracking by introducing more constraints between surface points. The second stage makes further effort to track motion of non-articulated motion. For long sequences, we stabilize the human surface in the un-seen part by directly warping surface from the first frame to the current frame according to sequentially tracked correspondences, preventing surface from collapsing caused by error accumulation. We demonstrate our method by several real captured RGBD data, containing complex human motion. The reconstruction results show the effectiveness and robustness of our method.**

***Keywords*-Kinect; dynamic reconstruction; human body**

I. Introduction

It is very important to reconstruct dense dynamic surface of human motion in computer graphics. This technique can be used in a variety of applications, ranging from virtual film-making and gaming to engineering and surveillance. For example, in the movie industry, the dense dynamic human surface reconstruction can provides more accurate and detailed motion information than sparse maker points from conventional motion capture systems.

Current methods on dense surface reconstruction are mainly two-fold: the first uses multiple sensors around the moving object to capture it from different views (e.g. [1], [2]). The second takes only a single sensor to capture one fixed view of the object. The single-view setup possesses the simplicity by avoiding multi-camera calibration and synchronization, but requires a reasonable inference from spatial and temporal coherence (e.g. [3], [4]). In the latter scenario, current state-of-the-art is the work of Li el al.[4], where a robust single-view geometry and motion reconstruction framework is proposed.In their system, it utilizes a structure light scanner to produce high-accurate depth of the captured object, and uses a template to confine occluded parts, leading to high-quality motion reconstruction.

Corresponding Author: Xinguo Liu.

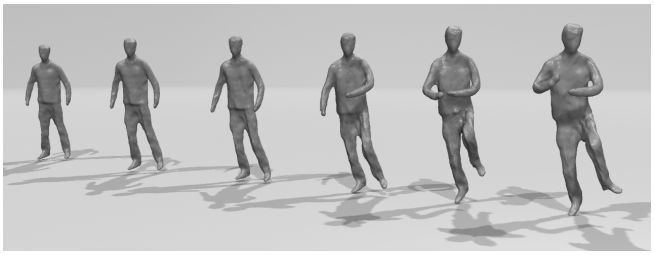

Figure 1: An example of dynamic human surface reconstruction from a single Kinect.

In this paper, we investigate the similar single-view motion reconstruction problem as Li el al.[4], but focus the target on human motion, and use only a Kinect, which provides a low-quality RGBD stream of captured scene. The confined target on human provides motion structure as prior for robust surface tracking. At the same time, the low-quality RGBD data largely interferes surface tracking algorithms, leading to bad dynamic surface reconstruction.

To address the human motion reconstruction problem based on a single Kinect, we propose a two-stage surface tracking framework. In the first stage we leverage the articulated motion property of human motion to track articulated surface deformation. Based on the articulated tracked results, in the second stage, we further track the non-articulated proportion of human motion. This two-stage scheme not only improves both the tracking accuracy and robustness, but also alleviates tracking error accumulation. To avoid shape collapsing on occluded regions which is out of view for an extended period of time, we further design a shape stabilization algorithm. The algorithm combines the current tracked shape with the directly warped shape from first frame, preventing error accumulation caused by shape inference due to occlusion. Thanks to these algorithms, our system is able to reconstruct pleasing surfaces of the human motion, e.g. the example shown in Fig. 1.

In summary, the contributions of this paper are mainly a two-stage framework and two ingredient algorithms therein:

- A two-stage tracking framework containing articulated and non-articulated tracking stages, which treats their corresponding motion respectively, improving tracking

robustness.
- A RGBD flow which integrates both geometry and color feature together to track motion.
- A first-to-current stabilization algorithm to keep reasonable shape in long period un-seen parts.

II. Related Work

The most relevant work to this paper is single-view motion reconstruction, but multi-view methods also provide similar technique fundament and background, which are very useful in single-view scenario. Here we introduce related work on methods of both multi-view and single-view.

Multi-view motion reconstruction takes multiple cameras (RGB or RGBD) from different views to capture the scene simultaneously, then uses the synchronous color/depth stream to reconstruct shape/motion of the captured scene. This kind of methods can capture almost all parts of the scene (except for self-occluded parts) at the same time, providing more shape/motion constraints for reconstruction. These years, as the rapid developments in hardware, there exists many multi-view methods. de Aguiar et al.[5] used optical flows of multi-view images and Laplacian deformation to track the surface of human motion. Vlasic et al. [1] proposed to leverage both articulation structure and silhouettes of human to constrain human motion; de Aguiar and his colleagues proposed a method based on image feature, silhouettes, and multi-view Stereo; Then, Vlasic et al.[2] integrated multi-view photometric stereo to obtain dynamic geometry, which produces much finer details than previous methods. But this method only reconstruct geometry of each frame separately, and it neither completes whole model from different frames, nor tracks surface temporally. Based on this system, Li et al. [6] completed shape and motion temporally by transferring geometry information to occluded regions. As Kinect began to be prevalent, Ye et al. [7] used three Kinects to reconstruct motion of highly interactive characters.

Single-view motion reconstruction uses only a camera (usually with depth sensor) to reconstruct a dynamic shape. This kind of methods require neither multi-camera calibration nor synchronization, but it can only get one view of the moving object. Shotton et al. [8] and Wei et al. [9] proposed to use a singe Kinect to tracking human skeleton in realtime, but these method do not reconstruct the human surface. To reconstruct the dynamic human surface from single-view is inherently a un-constrained problem, which can be solved by introducing priors. Niloy et al. [10] assumed temporal continently on motion and register dynamic object with slight deformation. Under the same assumption, Süssmuth et al.[11] and Sharf et al.[12] also proposed similar methods for space-time reconstruction. Liao et al. [3] utilized image features to track shape motion and optimize positions of these feature points. Their method does not consider depth information in tracking, which may fail in texture-less region. Pekelny et al. [13] and Chang et al. [14] separately proposed methods based on articulated motion assumption. These methods improve the tracking robustness by imposing more motion constraints. But the degree of freedom of the articulated motion is too low, which is unable to represent non-articulated motion. Zeng et al. [15] considered the as-rigid-as-possible assumption and proposed a shape reconstruction method which compensates slight non-rigid deformation, but this method can not handle large motion. Wand et al. [16] and Tevs et al. [17] also proposed shape and motion reconstruction frameworks. There methods require few assumptions on particular objects, but they hardly study the error accumulation problem.

Our method is most similar to the work of Li et al. [4]. However, we design new algorithms and improve the performance according to our low-priced setup and the specified articulated motion. Specifically, first, we propose a two-stage tracking method which treats articulated and non-articulated motion successively, improving tracking performance both on robustness and accuracy. Second, we use both RGB and depth data to track the shape, which further improve robustness of tracking on geometry-flat region or fast moving parts.

III. Overview

The pipeline of our system is illustrated in Fig. 2. The input is a RGBD stream of a human motion captured from a fixed Kinect, and the output is its frame-to-frame corresponding dynamic surface of the whole body. The system contains three main steps:

- **Preparation and Initialization Step** builds personalized human template for performer, and initializes the template's pose for the first frame.
- **Two-Stage Tracking Step** tracks human motion for each frame, first by an articulated tracker and then by a non-articulated tracker.
- **Temporal Filtering Step** bilaterally smoothes the tracked surface sequence in time domain and detail range.

The remainder of this paper is organized as follows: In Sect. IV we first introduce the deformation model as prerequisite knowledge of our method, and then in Sect. V we describe step of preparation and initialization, followed by the step of two-stage tracking in Sect. VI, and the step of temporal filtering in Sect. VII. We show experimental results in Sect. VIII, and conclude this paper in Sect. IX.

IV. Deformation Model

To represent surface motion of a human body, we adopt the embedded deformation model proposed by Sumner et al. [18] to compute warping filed, which represents non-rigid deformation behavior of the human surface's ambient space. In concept, the warping filed is interpolated by local deformations centered on some sampled nodes from the original

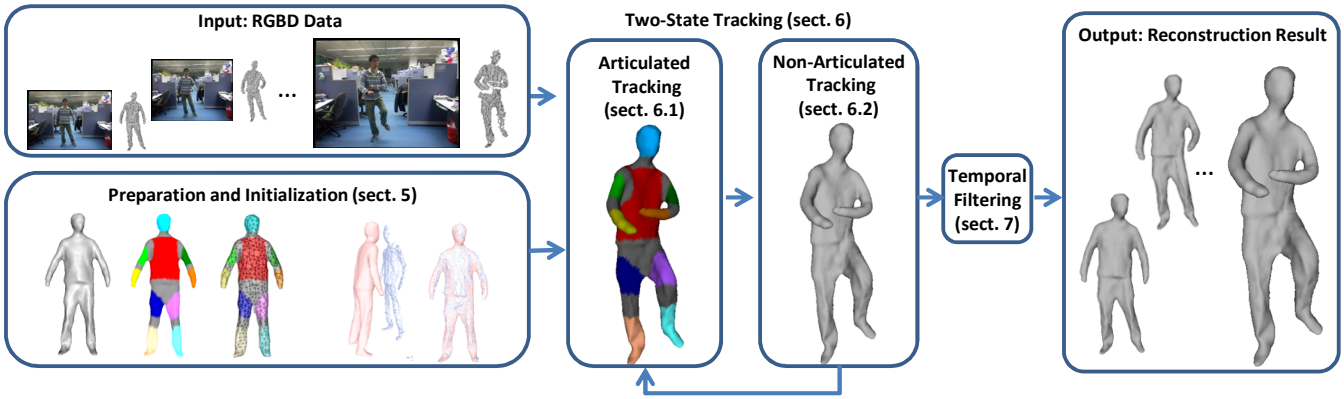

Figure 2: Overview of our method.

surface. Concretely, for a node s^i, it's local deformation can be represented by a 3×3 affine transformation matrix H^i, and a 3×1 translation vector l^i. Under this representation, a point p can be transformed to $\widetilde{p}$ by weighted influence from its nearby nodes, as follows:

$$\widetilde{p} = \sum_{j=1}^{K} w^j [H^j (p - s^j) + s^j + l^j], \tag{1}$$

where w^j is the normalized weights for p's j th-nearest nodes $s^j, j = 1, 2, ..., K$. We define w^j by the distance between p and the nodes as follows:

$$w^j = \frac{1 - ||p - s^j||/d_{max}}{\sum_{k=1}^{K} 1 - ||p - s^k||/d_{max}}, \tag{2}$$

with d_{max} is distance between p and its $K + 1$ th nearest node.

To determine local deformations of nodes, we follow Sumner et al.[18] to combine a fitting term E_{fit}, a rigidity term E_{rigid}, and a regularization term E_{reg} to form the minimization problem to solve H_i and l_i for all N nodes:

$$\min_{H^i, l^i, i=1,...,N} w_{fit} E_{fit} + w_{rigid} E_{rigid} + w_{reg} E_{reg} \tag{3}$$

In the fitting term, it constrains these nodes to desired positions by summing up distances of all m pairs of node-to-target correspondences:

$$E_{fit} = \sum_{i=1}^{m} M(s^i, q^{i*}) \tag{4}$$

where, q^{i*} is node s^i's correspondence in target scan, and $M(x, y)$ is some distance metric, which usually combines the point-to-point and the point-to-plane distance. We defer the details in following sections.

In the rigid term, it constrains the transformation matrix H_i to be rotational:

$$E_{rigid} = \sum_{s^i} Rot(H^i) \tag{5}$$

where $Rot(H) = ||H'H - I||_F^2$.

The regularization term considers the smoothness of the neighboring deformation, which measures the difference of the nearby nodes' transformations:

$$E_{reg} = \sum_{i} \sum_{j \in N(i)} ||H^i(s^j - s^i) + s^i + l^i - (s^j + l^j)||_2^2. \tag{6}$$

V. Preparation and Initialization

A. Template Building

Our system requires beforehand a template of the performer, which can be obtained by any human modeling algorithms, e.g. KinectFusion [19] or its other variants [20], [21]. Here, we employ quasi-rigid shape modeling [15] to build the human model. This method captures depth data of a self-turned human in front of a single fixed Kinect, and then fuses these data into a complete human model.

B. Articulated Parts Segmentation

After obtaining the human model, we manually specify articulated parts and joints of the human (Fig.3(b)). On this human model, we also sample nodes and build a deformation graph for non-rigid deformation (Fig. 3(c)). Then we cluster nodes on the deformation graph according to the articulated parts and joints (Fig. 3(d)).

C. Initial Rigid Registration

To initialize the rigid pose template, in the first frame, we manually choose some correspondences on the template and the scan, and compute the rigid transformation which best matches them, and then we use an additional ICP [22] procedure to refine the result. Fig. 4 show the procedure of initial rigid registration.

VI. Two-Stage Tracking

After transforming the template to its initial pose, we begin to track its surface to RGBD data of each frame. Leveraging the articulation structure obtained in the preparation step, we take a two-stage scheme to track human motion.

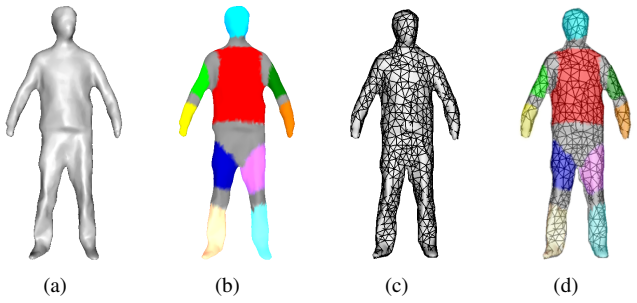

Figure 3: Static human model and its segmentation result. (a) human model. (b) articulated parts segmentation, the grey parts are joint regions, other colors indicate rigid region. (c) deformation graph. (d) clusters of the deformation graph according to the segmentation result in (b).

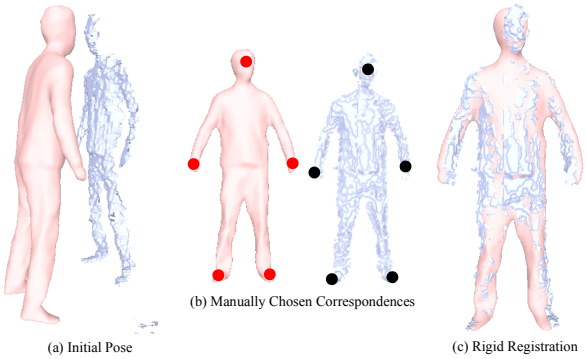

Figure 4: Initial Rigid Registration.

The first step requires the nodes on the same articulated part share the same transformation, which largely reduces the degrees of freedom of the tracking problem, both improving tracking robustness and reducing error accumulation. The second step drops the articulation information and use a general non-rigid deformation step to refine the tracking result. To stabilize the motion, with the final node-scan correspondence set, we take an extra two-stage tracking from the template on the first frame to current scan, leading to more robust results in the un-seen parts.

A. Articulated Tracking

1) Articulated Deformation Formulation: The articulated tracking ensures deformation nodes on the same rigid part have same motion, while nodes on the joints parts do not necessarily obey the constraints. Here we modify deformation formulation in Sect. IV and integrate this segmentation into it. In the original formulation, the deformation is presented by a local transformation of a node s^i, which is centered at s^i's position. In this presentation, nodes on the same rigid part will not have same translation vector l due to coupling between the rotation and non-zero node positions. Therefore, we represent the transformation in the original-centered fashion:

$$\begin{aligned} \widetilde{p} &= H^i \cdot p + t^i \\ \widetilde{n} &= H^i \cdot n \end{aligned} \tag{7}$$

Here, we deduce articulated deformation energy.

First, it ensures that nodes in the same rigid part P^h have the same transformation, i.e.:

$$H^i = R^h, t^i = T^h, \forall s^i \in P^h \tag{8}$$

where, R^h and T^h are rotation matrix and translation vector of part P^h.

Second, it also require the transformed nodes are close to the target scan according to the fitting term:

$$\begin{aligned} E_{fit} = \sum_{s^i} \quad \{ \quad & ||H^i \cdot s^i + t^i - q^{i*}||_2^2 \\ & + \rho \cdot (n^{i*} \cdot (H^i \cdot s^i + t^i - q^{i*}))^2 \quad \} \end{aligned} \tag{9}$$

where q^{i*} and n^{i*} are the position and normal of the node s^i's correspondence on the target scan. The correspondence searching will be described bellow. In this term, the first part is point-to-point distance, and the second part is point-to-plane distance. The parameter $\rho = 0.1$ balances these two distances.

Third, it requires that the nearby nodes have similar transformations. Unlike Eq.6, we do not impose smooth constraints between all nearby nodes, we only require nearby nodes which across different parts to be consistent (note that nodes on the same rigid part are inherently the same by Eq.8). Therefore, the regularization term of Eq.6 should be modified so that it only contains part-across node pairs:

$$E_{reg} = \sum_i \sum_{j \in N(i), P(i) \neq P(j)} ||H^i \cdot s^i + t^i - (H^j \cdot s^i + t^j)||_2^2 \tag{10}$$

Combining the fitting term Eq. 9, the regularization term Eq.10, the rigid term Eq.5, together with the cluster constraints Eq.8, we achieve the final optimization problem:

$$\begin{aligned} \min_{H^i, T^i} E_{tot}^{seg} &= E_{fit} + w_{rigid} \cdot E_{rigid} + w_{reg} \cdot E_{reg} \\ s.t. \quad & H^i = R^h, t^i = T^h, \forall (s^i, h) | s^i \in P^h \end{aligned} \tag{11}$$

In our implementation, we set weight $w_{rigid} = 1000000$ and $w_{reg} = 2500$. We adopt non-rigid ICP fashion to iteratively solve the optimization problem. In every iteration, we update the correspondence between nodes and the target scan, construct new optimization function, and solve it. To solve the optimization problem in each iteration, we directly substitute nodes' transformations by their part transformations, according to the articulated constraints. This way largely reduces the variables when solving this problem.

2) RGBD Flow Based Correspondence Searching: To search the correspondence for the fitting energy Eq. 9, we utilize both color and depth information to track the motion. For the registered deformation graph G_t in t-th frame, we

project the visible nodes set $S_{visible} = \{s^1, s^2, ..., s^n\}$ onto RGB image I_t on frame t, and denote the projected positions on the image space as $S_{proj} = \{x^1, x^2, ..., x^n\}$. Using the optical flow [23], we can estimate the optical flow $F_t(x)$ from current RGB image I_t to the next RGB image I_{t+1}. From $F_t(x)$ we can obtain the visible node set $S_{visible}$'s projected position $S_{proj}*$ on the I_{t+1}:

$$S^*_{proj} = \{x^1 + F_t(x^1), x^2 + F_t(x^2), ..., x^n + F_t(x^n)\} \tag{12}$$

Then we can look up in the target scan D_{t+1} according to 2D coordinates on the RGB image, thereby find the nodes's correspondence in the target scan D_{t+1}. Fig.5 illustrates the correspondence finding procedure based on RGBD flow.

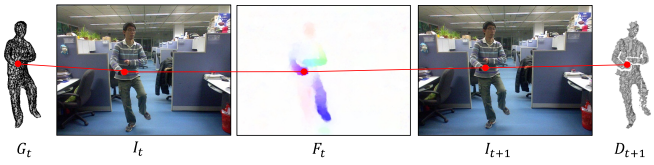

Figure 5: Illustration of RGBD flow based correspondence searching.

In the step of articulated tracking, we first use this RGBD flow based method to build correspondence, which can jump geometry-flat regions. Then we we take nearest neighbor search on Euclidean space to search correspondence and use normal direction to reject incompatible correspondence.

B. Non-Articulated Tracking

1) Frame-to-Frame Non-Articulated Tracking: After articulated tracking, each part of the human has approximately matched the captured data. However, due to existence of data noise or non-articulated deformation (such as folds of cloth), it is essential to refine the tracking result after articulated. In this non-articulated tracking step, we drop the articulated constraints in Eq. 8, and non-rigidly register the template to the target frame by solving the original optimization problem, i.e., Eq. 3. Here, the fitting term is Eq. 9, the rigid term is Eq. 5, and the regularization term is Eq. 6. Note that, here the regularization term takes all the nearby node pairs into account, including across-part and in-part pairs.

Similar to articulated tracking, we take non-rigid ICP fashion to solve the optimization problem. In each iteration, we re-search and update correspondences between graph nodes and the target scan. Since the template is close to the target scan, we directly use nearest neighbor searching to find correspondences. Besides, the non-articulated deformation will be too flexible to keep an as-rigid-as-possible shape, so in the procedure of optimizing Eq. 3, we need to adaptively control the weights in Eq. 3 to keep the tracked shape reasonable.

Relaxed Optimization Strategy We take Li's method [4] to dynamically change the weights. At the first iteration, we set large weights $w_{rigid} = 2500$ and $w_{reg} = 100$, so that to keep relatively stronger rigidity. As optimization proceeds, when the total energy changes slightly, we relax the rigidity of the template by setting lower w_{rigid} and w_{reg}. Specifically, we denote the energies of previous iteration and current iteration as E_{prev} and E_{curr}, respectively. If $|E_{curr} - E_{prev}| < \beta$, we reduce weights of rotation and regularization by half :$w_{rigid} \rightarrow \frac{1}{2} w_{rigid}, w_{reg} \rightarrow \frac{1}{2} w_{reg}$. Here we take the threshold $\beta = 0.0005$. To avoid over-flexibility of the template, we limit the times of the relaxation no more than 2.

2) Motion Stabilization: In a long sequence, occluded regions which are un-seen for an extend period is susceptible to error accumulation, since the observation does not provide any position constraints on these parts. These error accumulation are mainly caused by the non-articulated tracking, since the non-articulated tracking has too much deformation flexibility, which impose less constraints on the occluded regions. While in the articulated tracking, the piece-wise rigid prior impose a strong motion constraint for un-seen parts, therefore the tracking results of articulated tracking have less error accumulation than non-articulated tracking. In this spirit, we propose a motion stabilization algorithm to reduce error accumulation by directly warping template in the first frame $\mathcal{T}_1$ to current scan. More concretely, after frame-to-frame articulated tracking (sect. VI-A) and non-articulated tracking (sect. VI-B1), we record the set of final node-scan correspondence in S_{corr}. To keep proportion of articulated motion as large as possible, we articulated register the $\mathcal{T}_1$ to current scan as described in Sect. VI-A, but with fixed node-scan correspondence S_{corr}. After articulated registration, we go on to non-articulately register the result to current scan, also with the fixed node-scan correspondence S_{corr}. This scheme avoid obvious error accumulation due to long sequence tracking, leading to reasonable shape in the un-seen parts.

VII. Temporal Filtering

Although the two-stage tracking restrains data noise effectively, minor tracking error due to high-frequent data noise may lead to slight motion flicker. To further filter out these flicker, as a post-processing step in our system, we take a bilateral smoothing filter [1] on the whole sequence of the tracked motion, and output the filtered sequence as the final result.

VIII. Experiments

In this section, we first evaluate effectiveness of techniques used in our method and give timing analysis, and then we demonstrate our method by dynamic reconstructions from several RGBD sequences with challenging human

motions. In all our experiments, the resolution of RGB image and depth image are 640×480, and the capture fps is 30.

A. Evaluation

1) Without/With Articulated Tracking: We record a sequence of human motion, called "front tai chi" to compare methods without/with articulated constraints. The version without articulated tracking is in spirit like [4], which does not impose any semantic priors, leading to results without guarantee to be near human body space. Through this comparison, we demonstrate the effectiveness of our articulated tracking to produce semantic-reasonable results. As shown in Fig. 6, the top row are snapshots of "front tai chi", the second row shows results of one frame (indicated by blue dash rectangle), where green and red meshes are the results without and with articulated tracking, respectively. Fig. 6 (b) is the frontal view, while Fig. 6(c) is the top view. From them, we show that, through imposing piecewise rigid constraints, our two-stage tracking on the one hand produces much more reasonable shapes (highlighted in red dash rectangles); and on the other hand improves the robustness which avoids interference between different parts with very close distance (highlighted in red solid rectangles).

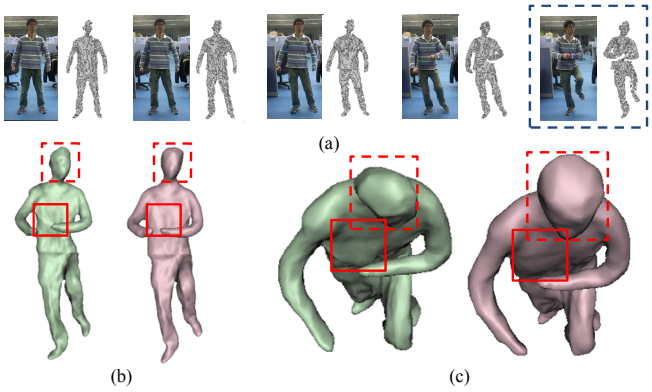

Figure 6: Comparison of without/with motion articulated tracking.

2) Without/With RGBD Flow: Fig.7 compares the tracking results without/with the RGBD flow on two successive frames with fast arm movements. In this figure, (a) and (b) are two successive frames, and (c) is visualization of the optical flow between these frames. (d) and (f) show tracking results (in red) of frame (b) without/with RGBD flow, respectively. We overlap the depth image (in blue) of (b) onto these two results to show qualities of alignment. From this comparison, we can see that the nearest neighbor searching alone (d) is unable to align fast moving regions, while the RGBD flow induced by the optical flow (f) provides relative more accurate correspondences in this case. Thus, the RGBD flow can be used in the first few iterations to get a better initial position, leading to a better alignment.

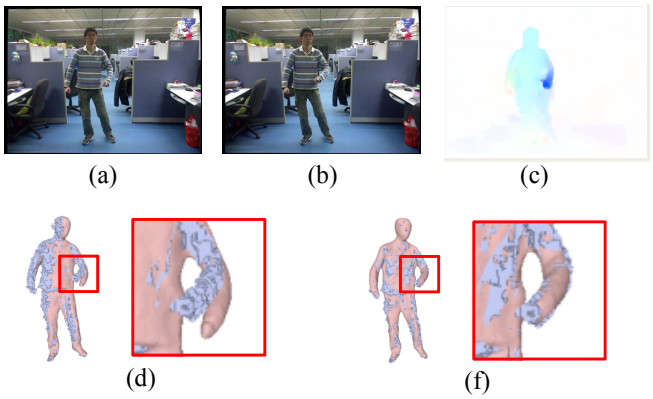

Figure 7: Comparison of without/with RGBD flow.

3) Without/With Motion Stabilization: We test the motion stabilization algorithm proposed in Sect. VI-B2 by comparing the system without and with the motion stabilization. We use two Kinects from front view and back view respectively to capture depth images from these two views. The front-view data is used to reconstruct the human motion, while the back-view data is used as a real-capture data to evaluate the back-view reconstruction of single-view methods. We therefore run our system without and with the motion stabilization on the front-view depthes, and qualitatively and quantitatively compare the back regions of their results with the captured back-view data.

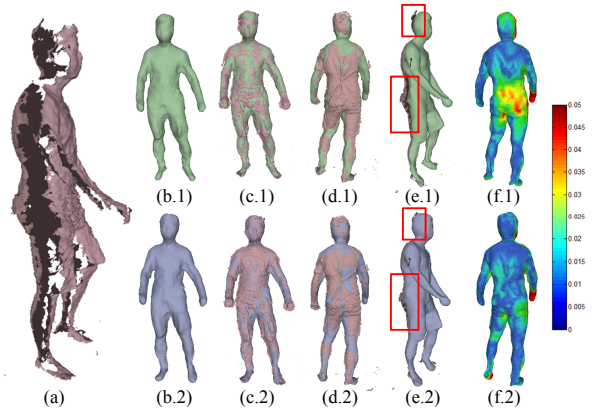

Figure 8: Comparison of without/with motion stabilization.

We show the comparison result in Fig 8. In this figure, (a) is the $17th$ frame of the captured data, we only use the front part as our input, while use the back part to evaluate results. The right parts of this figure are tracking results of this frame. The first (second) row is the result of the method without (with) motion stabilization. (b.1) and (b.2) show the front view of the results. (c.1) and (c.2) show the results overlapped by captured data in front view. Here, they are almost the same visually, since both meshes

are aligned to the capture data. (d.1) and (d.2) show the results overlapped by captured data in back view. (e.1) and (e.2) show this comparison from side view to display their difference clearly (see highlighted regions). We can observe that without stabilization, the shape is susceptible to over-deformation in the back due to missing data. (f.1) and (f.2) visualize the nearest distance $dist$ from the reconstructed mesh to the captured data. The average $dist$ of all vertices of pipelines without/with stabilization are 0.021m and 0.012m, respectively. It's obvious from this comparison that the stabilization keeps the un-seen parts in a more stable shape.

4) Timing: The entire pipeline of our method is implemented with C++ and tested on a desktop computer with Intel Core2 Duo E7400 2.80GH CPU (we used only one core) and 2GB RAM. We show in Table I the time of each step of a dynamic sequence with 50 frames. In our system, for a model, the manual labeling of articulated part is only one time, while other steps are needed for each frame. On each step, we list the average timing of per frame in brackets. From this table, we find that our current implementation consume relatively much time in computation of optical flow. This is because we use code of Liu [23] to this task, which computes optical flow for the whole image. We can significantly reduce the computation time by restricting the computation on regions covered by the human.

Table I: Timing of each step of our method (min)

label	opt. comp.	art. track	non-art. track	total
~ 2	72 (1.44)	10 (0.2)	12.7 (0.25)	96.7 (1.93)

B. Results

In this section, we show more reconstruction results produced by our method.

In Fig. 9, we show a "crouching" motion. Each row of this figure represents one frame of the sequence. In each row, the first column is color image, the second is depth scan, and from third to sixth columns are reconstruction results viewed from frontal, left, right, and back sides, respectively. The human shape in this sequence exhibits large deformation, and our tracking and reconstruction result capture motion/deformation details in exposed regions. In the un-seen regions, our method also produces convincing deformation details.

We also test our method on a long sequence "side kicking" which containing hundreds of frames. In Fig. 10, we show some snapshots of this sequence, where top frame and the bottom frame are 98 frames away. From this figure we can see that our reconstruction results keep a pleasing shape, even undergoing large motions which last for a long time.

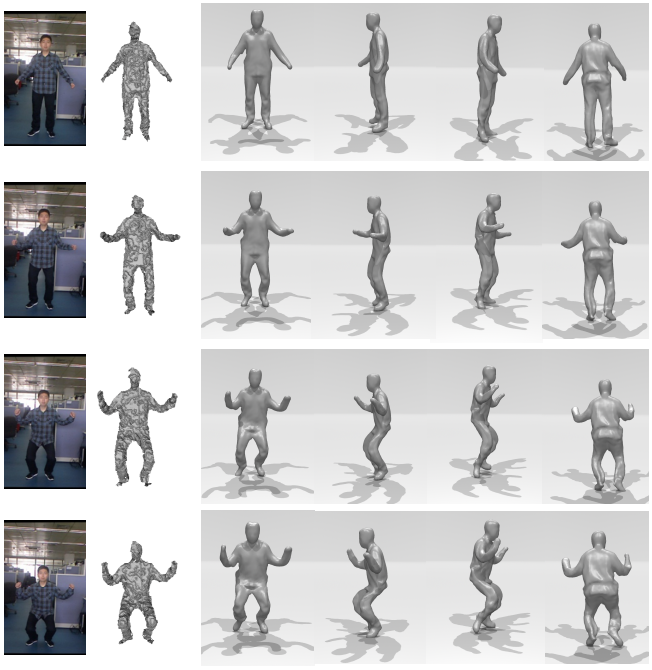

Figure 9: Reconstruction results of "crouching".

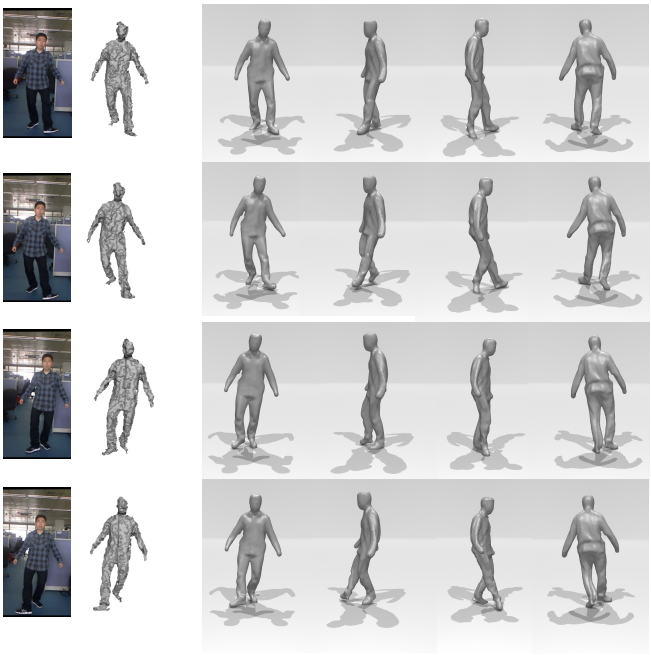

Figure 10: Reconstruction results of "side kicking".

IX. Conclusion

We have presented a robust framework for dynamic human motion reconstruction from a single Kinect. Our hardware setup is simple that uses only a single commodity RGBD sensor. In this setup, it does not require high-priced hardware or multi-camera calibration/synchronization. We proposed three main algorithms to robustly reconstruct human motion from the single-view noisy RGBD stream. First,

we design a two-stage tracking which tracks articulated and non-articulated proportion of the human motion subsequently. This two-stage scheme employs piece-wise rigid prior of human motion, which not only largely improves tracking robustness, but also imposes shape constraints on the un-seen parts. Second, we adopt a novel correspondence searching based on RGBD flow, leading to more reliable tracking on geometry-flat but texture-abundant regions. Third, we propose a motion stabilization algorithm to prevent tracked shape on long-term un-seen regions from collapsing.We demonstrate our method by extensive evaluation and several dynamic reconstructions examples from RGBD sequences of human motion.

ACKNOWLEDGMENT

This work was partially supported by NSFC (No. 61379068).

REFERENCES

[1] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 97:1–97:9, Aug. 2008.

[2] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik, "Dynamic shape capture using multi-view photometric stereo," *ACM Transactions on Graphics*, vol. 28, no. 5, p. 174, 2009.

[3] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong, "Modeling deformable objects from a single depth camera," in *Proceedings of ICCV 2009*. IEEE, 2009, pp. 167–174.

[4] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 175:1–175:10, Dec. 2009.

[5] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel, "Marker-less deformable mesh tracking for human shape and motion capture," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE. Minneapolis, USA: IEEE, June 2007, pp. 1–8.

[6] H. Li, L. Luo, D. Vlasic, P. Peers, J. Popović, M. Pauly, and S. Rusinkiewicz, "Temporally coherent completion of dynamic shapes," *ACM Transactions on Graphics*, vol. 31, no. 1, pp. 2:1–2:11, 2012.

[7] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt, "Performance capture of interacting characters with handheld kinects," in *Proceedings of the 12th European conference on Computer Vision - Volume Part II*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 828–841.

[8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11, 2011, pp. 1297–1304.

[9] X. Wei, P. Zhang, and J. Chai, "Accurate realtime full-body motion capture using a single depth camera," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 188:1–188:12, Nov. 2012.

[10] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann, "Dynamic geometry registration," in *Proceedings of the fifth Eurographics symposium on Geometry processing*, ser. SGP '07. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 173–182.

[11] J. SÜssmuth, M. Winter, and G. U. Greiner, "Reconstructing animated meshes from time-varying point clouds," *Computer Graphics Forum*, vol. 27, no. 5, pp. 1469–1476, 2008.

[12] A. Sharf, D. A. Alcantara, T. Lewiner, C. Greif, A. Sheffer, N. Amenta, and D. Cohen-Or, "Space-time surface reconstruction using incompressible flow," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 110:1–110:10, Dec. 2008.

[13] Y. Pekelny and C. Gotsman, "Articulated object reconstruction and markerless motion capture from depth video," *Comput. Graph. Forum*, pp. 399–408, 2008.

[14] W. Chang and M. Zwicker, "Global registration of dynamic range scans for articulated model reconstruction," *ACM Transactions on Graphics*, vol. 30, no. 3, pp. 1–15, 2011.

[15] M. Zeng, J. Zheng, X. Cheng, and X. Liu, "Templateless quasi-rigid shape modeling with implicit loop-closure," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[16] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling, "Efficient reconstruction of nonrigid shape and motion from real-time 3D scanner data," *ACM Transactions on Graphics*, vol. 28, no. 2, pp. 1–15, 2009.

[17] A. R. T. Tevs, A. Berner, M. Wand, I. V. O. Ihrke, M. Bokeloh, J. Kerber, and H.-p. Seidel, "Animation Cartography - Intrinsic Reconstruction of Shape and Motion," *ACM Transaction on Graphics*, vol. 31, no. 2, pp. 12:1–12:15, 2012.

[18] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 80, 2007.

[19] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *ISMAR '11*, 2011, pp. 127–136.

[20] M. Zeng, F. Zhao, J. Zheng, and X. Liu, "Octree-based fusion for realtime 3d reconstruction," *Graphical Models*, vol. 75, no. 3, pp. 126–136, 2013.

[21] T. Whelan, J. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard, "Kintinuous: Spatially extended kinectfusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, July 2012.

[22] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[23] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.