2013 13th International Conference on Computer-Aided Design and Computer Graphics

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 the shape in the un-seen part, and propose a two-stage tracking pipeline. The first stage tracks the articulated motion of the human, which improves robustness of tracking by introducing more constraints between the surface points. The second stage tracks movements of non-articulated motion. For long sequences, we stabilize the human surface in the un-seen part by directly warping the 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 provide more accurate and detailed motion information than using the sparse maker points in the 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 et 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 the occluded parts, leading to high-quality motion reconstruction.

Corresponding Author: Xinguo Liu.

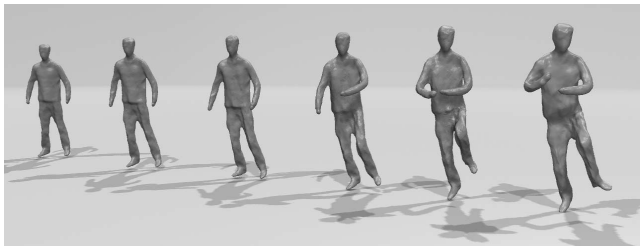


Figure 1: An example of dynamic human surface reconstruction from a single Kinect.

In this paper, we investigate single-view motion reconstruction problem for human motion, using a single 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, but 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 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 the accumulation of tracking error. 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, as 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.

978-1-4799-2576-6/13 $31.00 © 2013 IEEE
DOI 10.1109/CADGraphics.2013.32

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

II. Related Work

The most relevant work to this paper is single-view motion reconstruction, but multi-view methods also provide similar technique fundamentals and background, which are very useful in single-view scenario. Here we introduce the 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. In the recent 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 to constrain human motion; de Aguiar and his colleagues [6] 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 reconstructs 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. [7] completed shape and motion temporally by transferring the geometry information to the occluded regions. As Kinect becomes prevalent, Ye et al. [8] 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. [9] and Wei et al. [10] 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 under-constrained problem, which can be solved by introducing priors. Niloy et al. [11] assumed temporal continently on motion and register dynamic object with slight deformation. Under the same assumption, Süssmuth et al. [12] and Sharf et al. [13] also proposed similar methods for space-time reconstruction. Liao et al. [3] utilized image features to track the shape motion and optimize the positions of these feature points. Their method does not consider depth information in tracking, which may fail in texture-less region. Pekelny et al. [14] and Chang et al. [15] separately proposed similar methods based on articulated motion assumption. These methods improve the tracking robustness by imposing more motion constraints. But the reduced degree of freedom of the articulated motion is unable to represent non-articulated motion. Zeng et al. [16] 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. [17] and Tevs et al. [18] also proposed shape and motion reconstruction frameworks, but they didn't touch the error accumulation problem.

Our method is 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 human motion captured from a fixed Kinect, and the output is the corresponding dynamic surface of the whole body. The system has three main steps:

- **Preparation and Initialization Step** builds a personalized human template for the 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 both the time domain and the spatial range.

The remainder of this paper is organized as follows: Sect. IV introduces the deformation model as the prerequisite knowledge of our method, and Sect. V describes the preparation and initialization step, followed by the two-stage tracking step in Sect. VI, and the temporal filtering step 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 [19] to compute the warping filed which represents the non-rigid deformation behavior of the human surface's ambient space. In concept, the warping filed is interpolated by the local deformations centered on some sampled nodes from the original 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

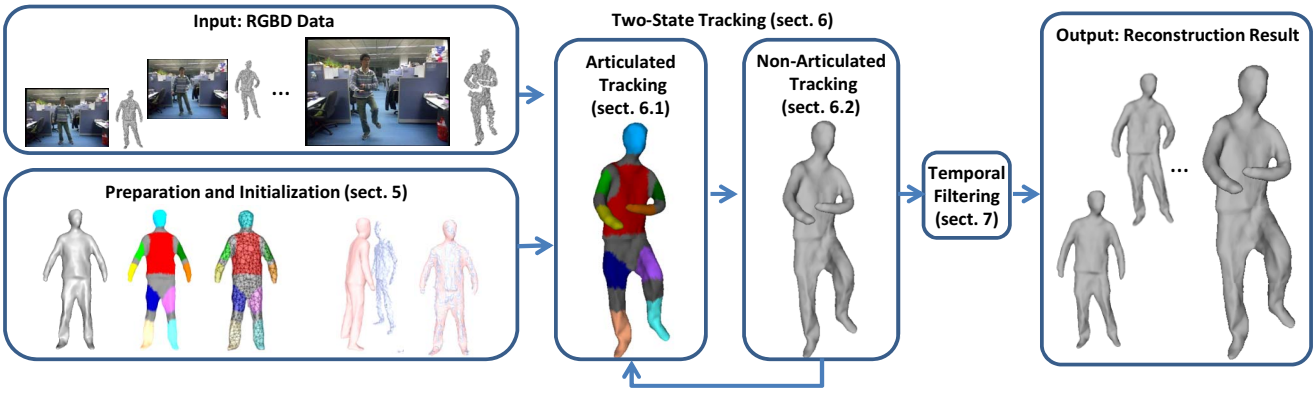


Figure 2: Overview of our method.

p can be transformed to $\widetilde{p}$ by a 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 weight 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 the local deformations of the nodes, we follow Sumner et al. [19] to combine a fitting term E_{fit}, a rigidity term E_{rigid} and a regularization term E_{reg} to formulate the following minimization problem to obtain H_i and l_i for all N nodes:

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

The fitting term constrains the nodes to the desired positions by summing up the 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 a distance metric combining the point-to-point and the point-to-plane distance. We defer the details in Sect.VI.

The rigid term 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 [20] or other variants [21], [22]. Here, we employ a quasi-rigid shape modeling [16] method. This method captures the depth data of a human turning around in front of a fixed Kinect, and then fuses these data into a complete human model.

B. Articulated Parts Segmentation

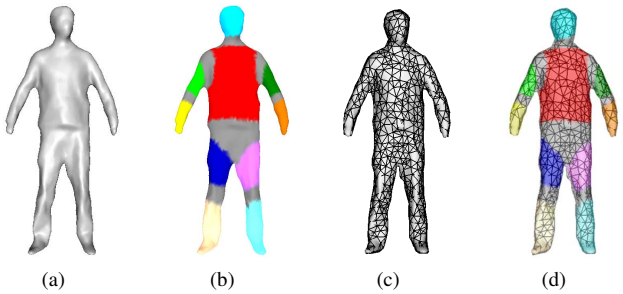
After obtaining the human model, we manually specify the articulated parts and the joints of the human (Fig.3(b)). We also sample a set of nodes and build a deformation graph for non-rigid deformation (Fig. 3(c)). Then we cluster nodes on the deformation graph according to the articulated parts and joints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

C. Initial Rigid Registration

To initialize the rigid pose template in the first frame, we manually choose some correspondences between the template and the depth scan, and compute the rigid transformation which best matches them. Then we use an additional ICP [23] 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 according 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. The first step requires the nodes on the same



(a) (b) (c) (d)

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).

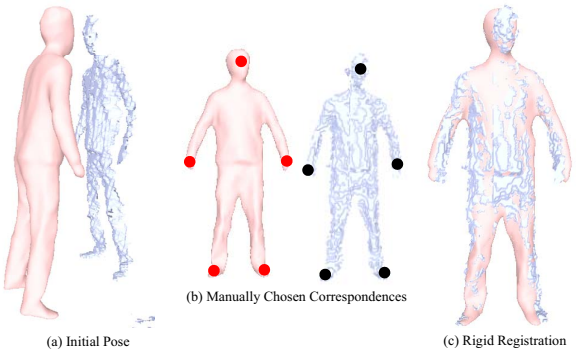


Figure 4: Initial Rigid Registration.

articulated part share the same transformation, which largely reduces the degrees of freedom of the tracking problem, improving tracking robustness and reducing error accumulation. The second step drops the articulation information and uses 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 the deformation nodes on the same rigid part have same motion, while the nodes on the joints parts do not necessarily obey the constraints. Here we modify the deformation formulation in Sect. IV, and integrate the articulated information 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 the coupling between the rotation and the non-zero node positions. Therefore, we represent the transformation in the fashion that uses the original as the rotation center:

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

Now, 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 in the part of RGBD flow. In this term, the first part is the point-to-point distance, and the second part is the 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 the nearby nodes 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 is modified to contain only the 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. To solve the optimization problem in each iteration, we directly substitute the nodes' transformations by their part transformations, according to the articulated constraints, which largely reduces the variables during minimization.

2) RGBD Flow Based Correspondence: To build 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 the RGB image I_t on frame t, and denote the projected positions on the image space as $S_{proj} = \{x^1, x^2, ..., x^n\}$. Then 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 the 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.

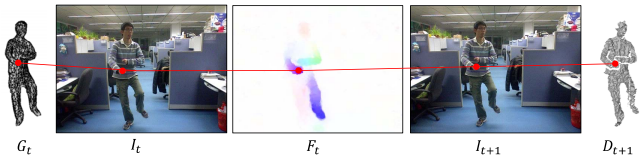


Figure 5: Illustration of RGBD flow based correspondence searching.

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 approximately matches the captured data. However, due to existence of noise or non-articulated deformation (such as folds of cloth), it is essential to refine the tracking result after articulated tracking. 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, of Eq. 3 with the fitting term in Eq. 9, the rigid term in Eq. 5, and the regularization term in Eq. 6. Note that, the regularization term takes all the nearby node pairs into account, including both the across-part and the in-part pairs.

Similar to articulated tracking, we take non-rigid ICP fashion to solve the optimization problem. In each iteration, we update the correspondences between the graph nodes and the target scan. Since the template is close to the target scan, we directly use nearest neighbor searching to find the 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 maintain the tracked shape.

Relaxed Optimization Strategy We use 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 only use the relaxation twice.

2) Motion Stabilization: In a long sequence, the occluded regions 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, which impose less constraints on the occluded regions. While in the articulated tracking, the piece-wise rigid prior imposes a strong motion constraint for un-seen parts, therefore the tracking results of the articulated tracking have less error accumulation than the 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 the 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 the final node-scan correspondence in S_{corr}. To keep the proportion of the articulated motion as large as possible, we articulately register the $\mathcal{T}_1$ to the current scan as described in Sect. VI-A, but with fixed node-scan correspondence S_{corr}. After articulated registration, we continue to non-articulately register the result to the current scan, also with the fixed node-scan correspondence S_{corr}. This scheme avoid obvious error accumulation of long sequence tracking, leading to reasonable shape in the un-seen parts.

VII. Temporal Filtering

Although the two-stage tracking restrains data noise effectively, the minor tracking error due to high-frequent data noise may lead to slight motion jittering. To further filter out these jittering, 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 our method with timing analysis, and then 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 similar to [4] in spirit, 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 some snapshots of "front tai chi", the second row shows the 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 piece-wise rigid constraints, our two-stage tracking produces much more reasonable shapes (highlighted in red dash rectangles) on the one hand and improves the robustness which avoids interference between different parts with very close distance (highlighted in red solid rectangles) on the other hand.

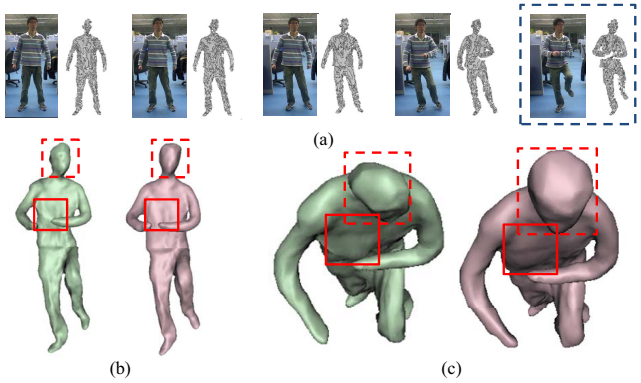


Figure 6: Comparison of without/with motion articulated tracking.

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 (e) 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 the alignment qualities. 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 (e) provides more accurate correspondences in this case. Thus, the RGBD flow can be used to get better initial position, and better alignment in turn.

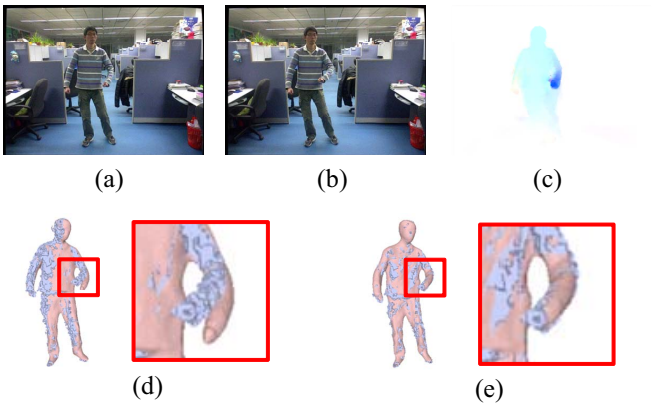


Figure 7: Comparison of without/with RGBD flow.

3) Without/With Motion Stabilization: We test the motion stabilization algorithm of Sect. VI-B2 by comparing the system without and with the motion stabilization. We use two Kinects to capture depth images from front view and back view respectively. 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 the results with the captured back-view data.

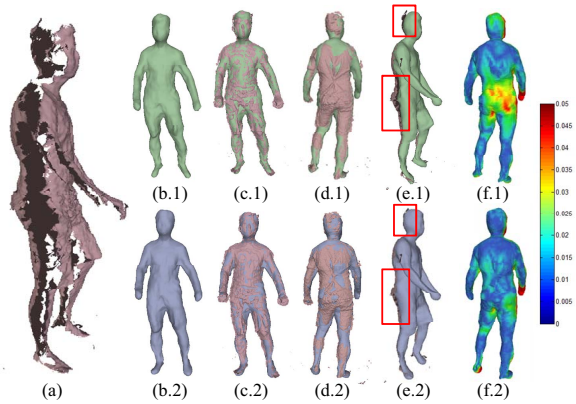


Figure 8: Comparison of without/with motion stabilization.

We show the comparison result in Fig 8. In Fig 8, (a) is the $17th$ frame of the captured data. The right parts of Fig 8 are the tracking results of this frame. The first (second) row is the result 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 the captured data in the back view. (e.1) and (e.2) show this comparison from a 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 the 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 for a dynamic sequence with 50 frames. In our system, the manual labeling of the articulated part is only performed once, while the other steps are needed for each frame. For each step, we list the average timing per frame in brackets. From this table, we find that our current implementation consumes a lot of time in computing optical flow. 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 by our method.

In Fig. 9, we show a "crouching" motion. Each row of this figure represents one frame of the sequence. The first column is color image, the second is depth scan, and the columns from the third to the sixth are the 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 the 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 the 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.

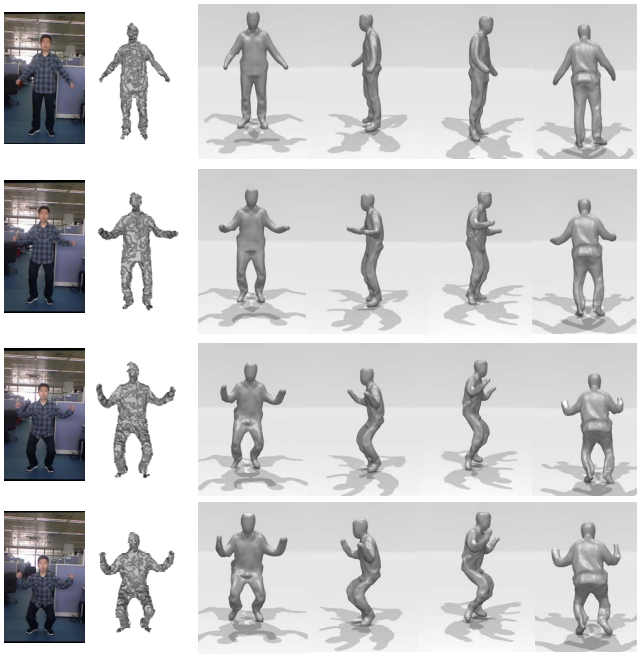


Figure 9: Reconstruction results of "crouching".

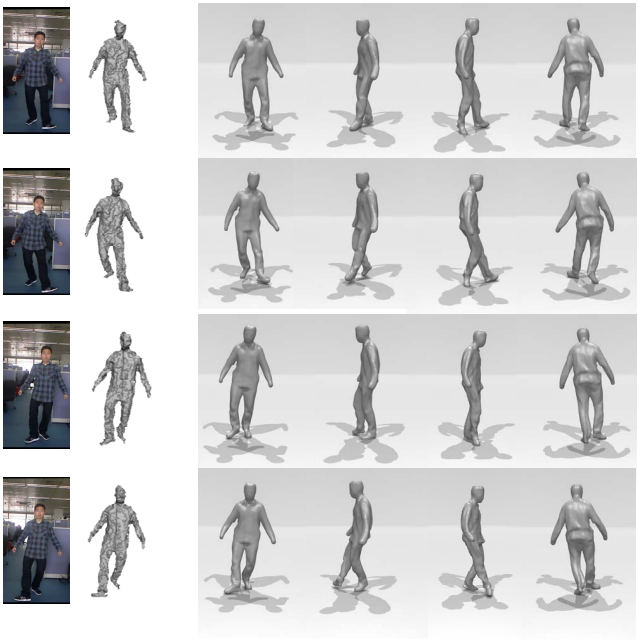


Figure 10: Reconstruction results of "side kicking".

IX. Conclusion and Future Work

We have presented a robust framework for dynamic human motion reconstruction using a single Kinect. The hardware setup is simple which does not require high-priced hardware or multi-camera calibration/synchronization. We incorporate piece-wise rigid prior of human motion into the tracking pipeline and propose a motion stabilization algorithm, which not only largely improves tracking robustness,

but also imposes shape constraints on the un-seen parts. We demonstrate our method by extensive evaluation and several dynamic reconstructions examples from RGBD sequences of human motion.

There are large space to improve in our system. In the future work, an avenue is to explore the geometry feature for more robust matching between frames. Besides, investigating how to reduce the reconstruction time may open a new path to some interactive applications, e.g. online human motion acquisition and editing.

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 Trans. Graph.*, vol. 28, no. 5, 2009.

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

[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 *CVPR 2007*.

[6] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

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

[8] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt, "Performance capture of interacting characters with handheld kinects," in *ECCV 2012*.

[9] 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 *CVPR 2011*.

[10] 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.

[11] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann, "Dynamic geometry registration," in *SGP 2007*.

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

[13] 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.

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

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

[16] M. Zeng, J. Zheng, X. Cheng, and X. Liu, "Templateless quasi-rigid shape modeling with implicit loop-closure," in *CVPR 2013*.

[17] 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 Trans. Graph.*, vol. 28, no. 2, pp. 1–15, 2009.

[18] 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 Trans. Graph.*, vol. 31, no. 2, pp. 12:1–12:15, 2012.

[19] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 80:1–80:8, 2007.

[20] 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 2011*.

[21] 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.

[22] 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.

[23] 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.