Robust Non-Rigid Motion Tracking and Surface Reconstruction Using L_0 Regularization

Kaiwen Guo, Feng Xu, Yangang Wang, Yebin Liu, *Member, IEEE*, and Qionghai Dai, *Senior Member, IEEE*

Abstract—We present a new motion tracking technique to robustly reconstruct non-rigid geometries and motions from a single view depth input recorded by a consumer depth sensor. The idea is based on the observation that most non-rigid motions (especially human-related motions) are intrinsically involved in articulate motion subspace. To take this advantage, we propose a novel L_0 based motion regularizer with an iterative solver that implicitly constrains local deformations with articulate structures, leading to reduced solution space and physical plausible deformations. The L_0 strategy is integrated into the available non-rigid motion tracking pipeline, and gradually extracts articulate joints information online with the tracking, which corrects the tracking errors in the results. The information of the articulate joints is used in the following tracking procedure to further improve the tracking accuracy and prevent tracking failures. Extensive experiments over complex human body motions with occlusions, facial and hand motions demonstrate that our approach substantially improves the robustness and accuracy in motion tracking.

Index Terms— Performance capture, non-rigid, single-view, L0

1 Introduction

Acquiring 3D models of deforming objects in real-life is attractive but remains challenging in computer vision and graphics. One kind of approach focuses on articulate motions like human body and hand motions, which are intrinsically driven by skeleton structures. These motions can be reconstructed by modeling the motions on skeletons [1], [2], [3], [4]. However, there are large numbers of deforming objects which cannot be completely modeled by skeletons, e.g., the activity of people grasping a non-rigid deforming pillow (Fig. 1). Besides, the accuracy of tracking is sensitive to the skeleton embedding and the surface skinning [5] strategies, which usually require manual operations to achieve high quality motion tracking [1], [6].

Non-rigid deformation [7], [8], [9] provides an appealing solution for dynamic object modeling since it does not require the build-in skeletons. The basic idea is to deform the vertices of a template model to fit the observation at each time step and follow some smooth motion priors. However, since the space of non-rigid deformation is much larger than that of the skeleton motion, and non-rigid deformation usually employs local optimizations, available non-rigid motion tracking methods are easy to fall into local minimum. Furthermore, they suffer from error accumulation, and would fail when tracking long motion sequences from noisy and incomplete data obtained by a single consumer depth sensor [10]. Robust tracking of complex human body and hand motions using non-rigid motion tracking techniques (without embedded skeleton) is still an open problem.

In this paper, we observe that most of the non-rigid motions implicitly contain articulate motions, which have strong deformation changes on some sparse regions, called joint regions, while keep consistent on other regions. This means when calculating spatial deformation gradient on object surface, only some joint regions have non-zero gradient values while other surface regions keep zero or close to zero.

Based on this key observation, we contribute a novel sparse non-rigid deformation framework to reconstruct non-rigid geometries and motions from a single view depth input via L_0-based motion constraint. In contrast to the widely used L_2 regularizer which sets a smooth constraint for the motion differences between neighboring vertices, the L_0 regularizer allows local non-smooth deformation on several significant deformation parts, i.e., joints of articulate motions, while constraints consistent motions on other regions. This method greatly reduces the solution space and yields a more physically plausible and therefore a more robust and high quality deformation.

For temporal successive frames, however, as all motions are small, the proposed L_0 regularizer is incapable to distinguish the articulate motions from non-rigid surface motions. On the other hand, with more frames accumulated, articulate motions become stronger while pure non-rigid motions always stay small. To this end, we first estimate per-frame motions by L_2 optimization and accumulate the motions of multiple frames until an anchor frame is reached, where the accumulated articulate motion is large enough to be detected. Then we apply L_0 optimization on the anchor frame to refine the tracking results, followed by

- *K. Guo, Y. Liu, and Q. Dai were with the Department of Automation and TNList, Tsinghua University, Beijing 100084, China. E-mail: guokaiwen_neu@126.com, {liuyebin, qhdai}@tsinghua.edu.cn.*
- *F. Xu was with the School of Software and TNList, Tsinghua University, Beijing 100084, China. E-mail: feng-xu@tsinghua.edu.cn.*
- *Y. Wang was with the Microsoft Research, Beijing 100080, China. E-mail: ygwang.thu@gmail.com.*

Manuscript received 8 Dec. 2016; revised 13 Mar. 2017; accepted 23 Mar. 2017. Date of publication 28 Mar. 2017; date of current version 28 Mar. 2018. Recommended for acceptance by T. Ju.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2017.2688331

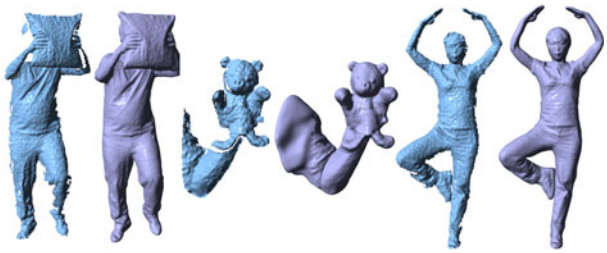


Fig. 1. Three reconstruction results of our method. For each result, we show the input depth and the reconstructed geometry model.

an L_2 optimization to reconstruct the rest non-rigid motions as a second step.

Comparing with the preliminary version [11], besides more thorough comparisons with existing state-of-the-art methods and more complete evaluations of the key components of the technique, this paper also proposes some key technical improvements, leading to a new tracking pipeline which is much more efficient and handles some challenging motions which the method of the original work [11] fails to produce.

To be specific, [11] reconstructs articulate motions independently for each pair of consecutive anchor frames, which means the locations of the articulate motions are repeatedly detected. However, for general objects like human, articulate motions always occur on joint regions which do not change over time. Based on this observation, we improve the articulate motion reconstruction by a new progressive strategy, which updates and refines the locations of articulate motions online as the motion proceeds. One benefit is that, for a newly detected anchor frame, we only need to check whether new joint regions can be detected, while the joint regions already detected in previous frames do not need to be considered any more, thus reducing a lot of computation time in L_0 optimization. More importantly, this strategy involves the information of previous anchor frames into the current one, thus the articulate motion reconstruction can be more robustly and accurately achieved. For example, in the case of occlusion, it is impossible to detect the occluded joints; but if it has already been detected in the previous anchor frames, the joint information can be used here to get a better result.

To fully leverage the online-updated locations of articulate joints, we integrate this information into the L_2 optimization. By simply setting spatial-variant weights on the smooth motion constraints, according to the locations of articulate joints, the L_2 optimization can generate large motions on joint regions while keep other regions with consistent motions, achieving desired motion reconstruction. In this manner, after all joint regions detected by previous anchor frames, the improved L_2 optimization alone is able to reconstruct articulate motions with the information of the joint distribution. So the L_0 optimization and the bidirectional tracking in [11] do not need to be performed again, which further makes our system much faster. In addition, the accuracy of the reconstructed motion is also improved over [11]. [11] solves for an L_0 optimization and an L_2 optimization in reconstructing the motions in anchor frames. This two-step algorithm may generate more errors compared with our unified algorithm that jointly estimates articulate motions and non-rigid motions. The difference is more apparent when handling repeating articulate motions. In this case, the accumulated errors of the two-step algorithm will become noticeable and generate artifacts while our unified algorithm solves this problem very well. Experiments will be shown in the result section.

In this paper, we demonstrate that, with monocular depth input captured by a consumer depth sensor, the proposed approach achieves accurate and robust reconstruction of complex non-rigid motions such as human body motions, facial expressions, hand motions and body motions interacting with objects. Our approach shows more robustness on tracking long sequences (up to 800 frames) with complex motions and significant occlusions, compared with the state-of-the-art non-rigid deformation methods. Furthermore, the technique does not rely on skeleton embedding and skinning weight calculation, thus dramatically reducing the workload of motion reconstruction and enabling much wide categories of objects to be tracked. The data and source code of our work are made public on the project website.[1]

2 Related Work

Techniques of non-rigid motion reconstruction have been widely used in recent years. For example, in movie and game industry, motion marker systems (e.g., Vicon[2]) are successfully applied to capture non-rigid motions of human bodies and faces. Nevertheless, these systems are quite expensive and require actors/actresses to stick a large set of optical markers on bodies or faces. To overcome this drawback, marker-less solutions with video input are extensively investigated in recent decades. Early works on this topic are well surveyed in [12] and [13].

For multi-view video input, the shape of moving objects can be directly reconstructed by shape-from-silhouette [14] or stereo matching [15] methods for each frame. After that, techniques like [16] are able to calculate the correspondences among all frames by a non-sequential registration scheme. Besides, a predefined template model can also be used to reconstruct the motion of an object by deforming it to fit the multi-view video input [17], [18], [19], [20]. Beyond that, a skeleton can be further embedded into the template to better capture kinematic motions of moving objects [1], [2], [21], [22]. Besides color cameras, multiple depth cameras are also used in recent years [23], [24]. Recently, Dou et al. [25] used 8 customized RGBD cameras to reconstruct dynamic scenes in real-time. The key volume strategy in their pipeline helps to reconstruct complex motions with topological changes. With the help of the depth information, complex motions are expected to be better reconstructed. Although the above solutions reconstruct articulate and/or non-rigid motions without motion markers, the sophisticated multi-view systems are still not easy to set up and cannot be applied to general environment, which strictly limits their applications.

Monocular color or depth camera is a much more facilitative device for capturing moving objects. Some works focused on rigid scenes. For kinematic body motions, Zhu et al. [26] reconstructed 3D body skeletons by modeling human actions as a union of subspace. Baak et al. [27] and Ye et al. [28] identified a similar pose in a prerecorded database to reconstruct the human pose for a video frame. Wei et al. [10] formulated the pose estimation problem as a *Maximum A Posteriori* (MAP) framework to achieve more robust skeleton estimation. Chen

1. http://media.au.tsinghua.edu.cn/nonrigid.html
2. http://www.vicon.com/

et al. [29] and Ye et al. [30] used fast LBS and template fitting to estimate pose transformations in real time, respectively. However, these techniques only estimate kinematic motions of moving objects, the full surface non-rigid deformations are not reconstructed. Recently, Wu et al. [31] reconstructed the non-rigid body motion with stereo input by exploring BRDF information and scene illumination. Ye and Yang [32] proposed an exponential-maps-based parametrization to estimate 3D poses and shapes. However, these techniques utilize a skeleton to constrain the kinematic motion space, which requires skeleton embedding and skinning weight calculation. These two steps are crucial to the quality of the final results and are difficult to be precisely achieved by automatic methods. Furthermore, the skeleton restricts the techniques to be applied only to articulate objects rather than general objects.

Besides exploiting skeleton models, data-driven methods are also applicable to reconstruct body motions. Zhang et al. [33] trained a regression model using several complete models of the same person with the same mesh topology and then tracked performer's motion based on this model. Bogo et al. [34] exploited texture information to estimate both geometry and appearance of a human body based on an extended shape model.

On the other hand, pure non-rigid registration technique, surveyed in [35], is an alternative solution to avoid using skeleton. For alignment of articulated motions, Chang and Zwicker [36] registered partial scans with articulated motions by solving for the optimal transformation for each part of the shapes. In [37], they furthermore used a reduced deformation model with the linear blend skinning technique to align partial scans of articulated objects. Pekelny and Gotsman [38] detected and tracked rigid components of articulated objects and accumulated their geometries over time. For alignment of general non-rigid motions, Li et al. [39] adopted the embedded deformation model from [8] to simultaneously solve for correspondences, confidence weights and parameters of a warping field. Wand et al. [40] applied a subspace deformation method to generate dense correspondences of the sequence and a complete geometry of the object. Mitra et al. [41] exploited 4D information to estimate motions of the underlying space-time surface. Li et al. [42] reconstructed complete geometry and albedo models of users by non-rigidly registering multiple partial scans in the presence of quasi-rigid motions. Bojsen-Hansen et al. [43] extended the non-rigid surface registration method to handle topological changes in liquid simulations. Liao et al. [44] applied a linear variational deformation technique to stitch partial surfaces at different time instances to generate complete models with corresponding motions, but limited to continuous and predictable motions. Popa et al. [45] achieved space-time reconstruction with a gradual change prior, which caused it difficult to handle fast motions and long sequences. Li et al. [46] and Zollhöfer et al. [47] reconstructed complex motions using template tracking based on ICP-defined correspondences, which achieved the state-of-the-art reconstruction. However, as only smooth motion prior is involved in their deformation models, strong articulate motions and large occlusions are difficult to be handled especially for noisy depth input captured by a consumer depth camera. In this paper, we propose a method that combines the benefits of the skeleton based and non-rigid registration based methods and demonstrate robust and accurate surface motion reconstruction from a single-view depth input.

Most recently, [48] and [49] perform dynamic reconstruction without initial geometry templates, which is more convenient for data recording. However, with only smooth and rigid motion constraints, these techniques only handle slow and controlled motions but not fast and complex motions as we do. Comparisons with our method are demonstrated in the result section and supplementary videos, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2017.2688331. Please note that our technique requires geometry templates and offline processing while [48] achieves real-time performance.

3 Overview

Our goal is to reconstruct the non-rigid motions of deforming objects from a single-view depth sequence. Different from existing solutions for reconstructing articulate motions [1], [21], our method does not require the embedding of a predefined skeleton, while still has the ability to robustly reconstruct the complex articulate motions of dynamic objects. In addition to the input depth sequence, we require the 3D mesh templates of the deforming objects, which are obtained by depth fusion [50] using a single depth sensor. In this way, the whole pipeline only relies on one off-the-shelf depth camera. In most of our sequences, the performer begins with a standard A-pose or T-pose which is the same to the pose we use for modeling the template. Therefore, we use a rigid ICP procedure to initialize the registration between the template and the first depth frame. The energy formulation of the rigid ICP includes both point-to-point and point-to-plane distances of mutual correspondences between the template and the first depth map. To prevent local minimum of the rigid ICP method, we adopt the particle-based global optimization approach similar to [1], [51]. We initialize a group of particles which uniformly sample the 6-degree solution space. With increased iterations, the mass of the distribution fitted by these particles converges to the global minimum of the rigid ICP energy. For the scenes with multiple objects, e.g., a pillow and a performer in one of our experiments, we first use the object labels returned by Kinect SDK to segment different objects. Then the above global optimization method is used to find the solution for each of the objects separately.

Our tracking pipeline automatically reconstructs object motion and locates articulate joints as illustrated in Fig. 2. Overall, it performs a forward-backward tracking strategy before all joints located, and a simple forward tracking afterwards. In the forward-backward strategy, an L_2 based non-rigid registration, which involves the current joint location information, is first performed frame by frame sequentially (step 1 in Fig. 2). Simultaneously, the reconstructed motion is accumulated until prominent articulate motion on new joint is detected at one frame, defined as *anchor frame*. Then the L_0 based motion regularization is triggered to locate the joint and refresh the articulate motion on the joint using the reference from the previous anchor frame (step 2 in Fig. 2). With the newly updated joint location information, the L_2 based

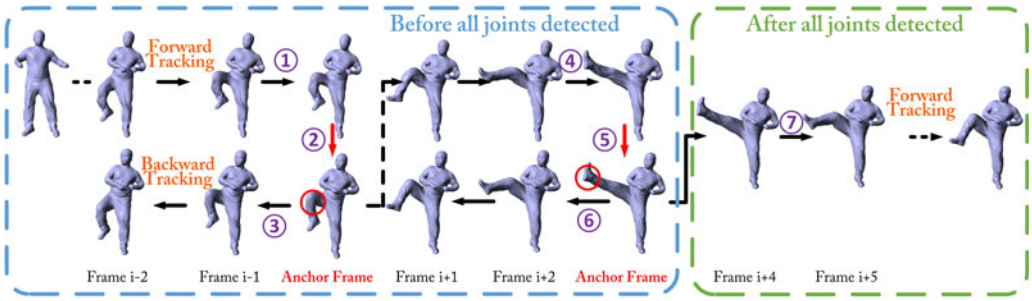


Fig. 2. The pipeline of the proposed method. Please refer to Section 3 for detailed description. The red circles indicate the newly detected joints for each anchor frame.

non-rigid registration is performed again on this anchor frame and also backwards (step 3 in Fig. 2) until the previous anchor frame to refine the in-between articulate motion and reduce the non-rigid tracking errors on the newly detected joint regions. This forward-backward strategy goes on from one anchor frame to the next detected anchor frame (step 4 to 6 in Fig. 2) until all articulate joint regions are located. With the final joint location information, the forward L_2 tracking method alone (step 7 in Fig. 2) is able to simultaneously reconstruct both articulate motions and the rest non-rigid motions. Note that for each input frame, we perform surface detail refinement (see Fig. 3e) as the last step to further reconstruct detail motions recorded by the input depth.

4 Method

Given a captured depth sequence $\{D^1, D^2, \ldots, D^n\}$, the proposed tracking strategy performs L_2 based regularizer and/or L_0 based regularizer for each frame D^t. In the following, we will first introduce our novel L_2 based non-rigid registration which involves joint location information, and then our proposed L_0 based motion regularization which update the joint location information. Then we describe our scheme to select between these two regularizers. Finally, we introduce the backward tracking and the detail refinement steps to calculate the final output of our system.

4.1 L_2-Based Registration with Joint Information

Given a depth frame D^t $(t = 1, \ldots, n)$ and a mesh M^{t-1} which is roughly aligned with the current depth D^t, our L_2 based non-rigid registration method further deforms M^{t-1} to fit D^t, guided by the joint information of the deforming object. For conciseness, we ignore the time stamp t in the following derivations. Following the state-of-the-art method [46], the deformation of a mesh M is represented by affine transformations $\{\mathbf{A}_i, \mathbf{t}_i\}$ of some sparse nodes $\{\mathbf{x}_i\}$ on the mesh (Fig. 3b). For a particular mesh vertex $\mathbf{v}_j$, its new position after the non-rigid deformation is formulated as

$$\mathbf{v}'_j = \sum_{\mathbf{x}_i \in \mathcal{N}(\mathbf{v}_j)} w(\mathbf{v}_j, \mathbf{x}_i)[\mathbf{A}_i(\mathbf{v}_j - \mathbf{x}_i) + \mathbf{x}_i + \mathbf{t}_i], \tag{1}$$

where $w(\mathbf{v}_j, \mathbf{x}_i)$ measures the influence of the node $\mathbf{x}_i$ to the vertex $\mathbf{v}_j$. Please refer to [46] for details about extracting $\mathbf{x}_i$ from the mesh and calculating w for all mesh vertices. Given the deformation model, the estimation of $\{\mathbf{A}_i, \mathbf{t}_i\}$ is achieved by minimizing the following energy:

$$E_{\text{tol}} = E_{\text{fit}} + \alpha_{\text{rigid}} E_{\text{rigid}} + \alpha_{\text{smo}} E_{\text{smo}}, \tag{2}$$

where

$$E_{\text{fit}} = \sum_{\mathbf{v}_j \in \mathcal{C}} \alpha_{\text{point}} \|\mathbf{v}'_j - \mathbf{c}_j\|_2^2 + \alpha_{\text{plane}} |\mathbf{n}_{\mathbf{c}_j}^T (\mathbf{v}'_j - \mathbf{c}_j)|^2. \tag{3}$$

which forces vertex $\mathbf{v}_j$ to move to its corresponding depth point $\mathbf{c}_j$ especially along the normal direction of $\mathbf{c}_j$. $\mathcal{C}$ includes all vertices that have correspondences in the depth D. E_{rigid} restricts the affine transformation to be as rigid as possible, which is formulated as

$$\begin{aligned} E_{\text{rigid}} = R(\mathbf{A}_i) = \sum_i \Big(&(\mathbf{a}_{i1}^T \mathbf{a}_{i2})^2 + (\mathbf{a}_{i2}^T \mathbf{a}_{i3})^2 + (\mathbf{a}_{i3}^T \mathbf{a}_{i1})^2 \\ &+ (1 - \mathbf{a}_{i1}^T \mathbf{a}_{i1})^2 + (1 - \mathbf{a}_{i2}^T \mathbf{a}_{i2})^2 + (1 - \mathbf{a}_{i3}^T \mathbf{a}_{i3})^2 \Big), \end{aligned} \tag{4}$$

where $\mathbf{a}_{i1}$, $\mathbf{a}_{i2}$ and $\mathbf{a}_{i3}$ are column vectors of $\mathbf{A}_i$. E_{smo} defines the L_2 regularizer which constrains the consistent motion difference on the spatial domain. Namely, the affine transformation of a node should be as similar as possible to those of its neighboring nodes

$$E_{\text{smo}} = \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} r_{ij} w(\mathbf{x}_j, \mathbf{x}_i) \|\mathbf{A}_i(\mathbf{x}_j - \mathbf{x}_i) + \mathbf{x}_i + \mathbf{t}_i - (\mathbf{x}_j + \mathbf{t}_j)\|_2^2. \tag{5}$$

The neighborhood of the nodes is shown as graph edges in Fig. 3b and is defined by the method in [46]. The minimization of E_{tol} is performed in an Iterative Closest Point (ICP) framework, where $\mathcal{C}$ is updated by closest point searching and parameters are also updated during the iterations. We exactly follow [46] to set parameters in our implementation. Please refer to their paper for details.

The difference between Eqn. (5) and the previous smooth term used in [46] is the parameter r_{ij}, which encodes how much motion smoothness should be assigned to two neighboring nodes i and j. For articulate motions, r_{ij} should be

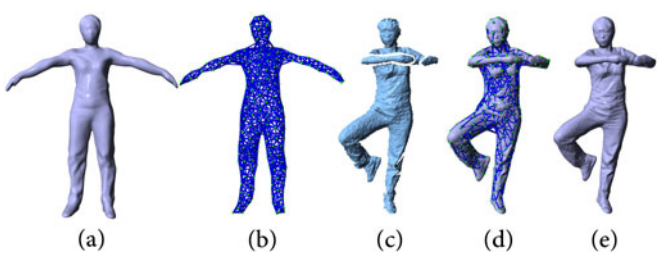


Fig. 3. Non-rigid registration. (a,b) Initial model with nodes and their connectivity; (c) input depth; (d) result of the non-rigid registration; (e) result of surface refinement.

1.0 for nodes on the same rigid part to enforce smooth constraint, while it should be smaller for nodes on different sides of one motion joint to attenuate the smooth constraint. However, previous techniques like [46] do not explore such information, thus all node pairs have the same smoothness strength. On the contrary, our method distinguishes the two kinds of node pairs in processing the motion sequence and uses r_{ij} to adaptively control the smoothness. To be specific, r_{ij} is updated based on the result of L_0 minimization illustrated in the following section. We first initialize $\{r_{ij}\}$ to be 1.0 for all node pairs. Then, with the processing of an input sequence, as more motion joints are detected by the L_0 minimization, more r_{ij}s are assigned small values to reduce the unreasonable smooth constraints around joint regions. Thus our method can reconstruct more accurate and more robust tracking results. In our implementation, r_{ij} is set to 0.1 for the detected joint regions in all tested sequences.

Notice that by simply involving r_{ij} in the smooth term of L_2 based motion registration, the joint location information is successfully involved and we obtain a unified formulation to reconstruct both the articulate motion and the rest non-rigid motion. Comparing with the two-step solution which first performs L_0 based regularization and then L_2 based registration in [11], the unified optimization achieves better results as demonstrated in the result section and the supplementary video, available online.

4.2 L_0-Based Regularization for Joint Update

As illustrated in Section 1, from single-view low quality depth input captured by a consumer depth sensor, pure non-rigid registration can not robustly and accurately reconstruct objects like a human body or human hands, whose motions may have strong occlusions which lead to inaccurate point-to-depth correspondences. But on the other hand, these kinds of objects usually perform articulate motions besides non-rigid motions. To pursue good tracking results, previous works adopt skeleton embedding to explicitly exploit the articulate motion prior, which strictly restricts that possible motion changes only happen on pre-defined skeleton joints and prevents motion changes on other regions. This skeleton embedding is similar to constrain the L_0 norm of spatial motion variation with a pre-defined distribution on the object. Based on this observation, we propose an L_0 based motion regularizer over the existing non-rigid surface deformation framework to implicitly utilize the articulate motion prior without the requirement of skeleton embedding.

Attention should be paid here that, the proposed L_0 regularizer can not be applied to every input frame. Intuitively, although the deformation change between two temporal successive frames contains both articulate motions and non-rigid motions, the magnitude of the articulate motions is too small and ambiguous to be distinguished from the non-rigid motions. If L_0 regularizer is applied to these tiny motions, the articulate motions will also be pruned with the non-rigid motions by the L_0 regularizer, which will lead to tracking failure. Therefore, we only apply L_0 regularizer to some anchor frames, and track the kinematic motion and shape of an anchor frame using the previous anchor frame as a reference.

Specifically, given the initial vertex positions $\{\mathbf{v}'_j\}$ of the new anchor frame obtained by the L_2 non-rigid tracking in Section 4.1, we estimate the refined implicit articulate transformation $\{\mathbf{A}'_i, \mathbf{t}'_i\}$ by minimizing the following energy function:

$$E'_{\text{tol}} = E'_{\text{data}} + \alpha'_{\text{rigid}} E'_{\text{rigid}} + \alpha'_{\text{reg}} E'_{\text{reg}}. \tag{6}$$

Here, E'_{data} constrains that the transformation to be solved should deform the object of the previous anchor frame to a similar pose obtained by the L_2 optimization, thus the result still fits the input depth

$$E'_{\text{data}} = \sum_j \|\mathbf{v}''_j - \mathbf{v}'_j\|_2^2, \tag{7}$$

where $\mathbf{v}''_j$ is the vertex position defined by the transformation to be solved

$$\mathbf{v}''_j = \sum_{\mathbf{x}_i \in \mathcal{N}(\mathbf{v}_j)} w(\mathbf{v}_j, \mathbf{x}_i)[\mathbf{A}'_i(\mathbf{v}_j - \mathbf{x}_i) + \mathbf{x}_i + \mathbf{t}'_i]. \tag{8}$$

E'_{rigid} has the same formulation as shown in Eqn. (4)

$$E'_{\text{rigid}} = R(\mathbf{A}'_i). \tag{9}$$

E'_{reg} brings the articulate motion prior into the optimization. It constrains that motions defined on the nodes do not change smoothly over the object but only change between sparse pairs of neighboring nodes. This is a plausible assumption because of the fact that the nodes on the same body part mostly share the same motion transform. We therefore formulate this term as an L_0 regularizer as

$$\begin{aligned} E'_{\text{reg}} &= \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \cap \mathcal{E}_i} \|\|\mathbf{D}x_{ij}\|_2\|_0, \\ \mathbf{D}x_{ij} &= \mathbf{A}'_i(\mathbf{x}_j - \mathbf{x}_i) + \mathbf{x}_i + \mathbf{t}'_i - (\mathbf{x}_j + \mathbf{t}'_j). \end{aligned} \tag{10}$$

Here $\mathcal{E}_i = \{x_j | r_{ij} = 1\}$ is the neighbor set in which each edge between x_i an x_j is subject to the unattenuated smooth term constraint. $\|\mathbf{D}x_{ij}\|_2$ represents the magnitude of the motion difference, and E'_{reg} measures the L_0 norm of the motion difference between all pairs of neighboring nodes except the pairs around already detected joint regions. This is reasonable because those nodes may have totally different motions. In our implementation, α'_{rigid} is set to 1,000, and α'_{reg} is set to 1.

Eqn. (6) is difficult to be optimized as the E'_{reg} term brings a discrete counting metric. Inspired by the solver described in [52], we split the optimization into two subproblems by introducing auxiliary variables into the energy function. Notice that the original L_0 optimization is computational intractable, and our solution is only an approximation. However, the proposed method is effective to get a good enough solution.

We introduce auxiliary variables $\{\mathbf{k}_{ij}\}$ and reformulate the optimization problem as

$$\begin{aligned} \min_{\mathbf{A}'_i, \mathbf{t}'_i, \mathbf{k}_{ij}} & E'_{\text{data}} + \alpha'_{\text{rigid}} E'_{\text{rigid}} \\ & + \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \cap \mathcal{E}_i} \lambda \|\|\mathbf{k}_{ij}\|_2\|_0 + \beta \|\mathbf{D}x_{ij} - \mathbf{k}_{ij}\|_2^2. \end{aligned} \tag{11}$$

Here $\mathbf{k}_{ij}$ is an approximation to $\mathbf{D}x_{ij}$. To solve this problem, we alternatively fix $\{\mathbf{A}'_i, \mathbf{t}'_i\}$ to solve $\{\mathbf{k}_{ij}\}$ and fix $\{\mathbf{k}_{ij}\}$ to solve $\{\mathbf{A}'_i, \mathbf{t}'_i\}$. If $\{\mathbf{A}'_i, \mathbf{t}'_i\}$ are fixed, the minimization is formulated as

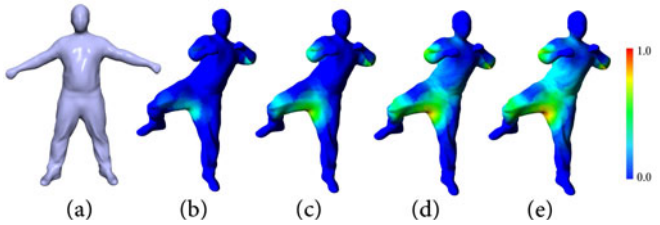


Fig. 4. Color coded normalized magnitude of $\{\mathbf{k}_{ij}\}$ on the vertices during iterations in solving L_0 minimization. Blue color stands for lowest (0.0) magnitude, green for higher and red for the highest (1.0) magnitude. (a) The previous L_0 anchor frame; (b-e) some of the intermediate iteration steps.

$$\min_{\mathbf{k}_{ij}} \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \cap \mathcal{E}_i} \lambda \| \|\mathbf{k}_{ij}\|_2 \|_0 + \beta \|\mathbf{D}x_{ij} - \mathbf{k}_{ij}\|_2^2. \tag{12}$$

As $\{\mathbf{D}x_{ij}\}$ are invariant, Eqn. (12) has a close form solution

$$\mathbf{k}_{ij} = \begin{cases} 0 & \text{if } \|\mathbf{D}x_{ij}\|_2^2 < \lambda/\beta \\ \mathbf{D}x_{ij} & \text{if } \|\mathbf{D}x_{ij}\|_2^2 \geq \lambda/\beta \end{cases}. \tag{13}$$

If $\{\mathbf{k}_{ij}\}$ are fixed, Eqn. (11) has the following formulation:

$$\min_{\mathbf{A}'_i, \mathbf{t}'_i} E'_{\text{data}} + \alpha'_{\text{rigid}} E'_{\text{rigid}} + \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \cap \mathcal{E}_i} \beta \|\mathbf{D}x_{ij} - \mathbf{k}_{ij}\|_2^2. \tag{14}$$

Eqn. (14) formulates a pure L_2 based optimization problem. We solve it by the Gauss-Newton method.

In solving Eqn. (11) with this iterative method, the parameters λ and β need to be changed in the iterations. In all our experiments, we fix λ to be 0.02, and set β to be 1.0 in the first iteration and multiplied by 2 after each iteration until β exceeds 10^6. This guarantees that Eqn. (11) defines a good approximation of the original problem. Fig. 4 illustrates the vertex motion magnitude during the L_0 iteration updates. Comparing with the pose at previous anchor frame, we see that the crotch between two legs has noticeable motion. Correspondingly, this region is successfully detected by the algorithm as an articulate region at the beginning of the iterations. With iterations going on, more articulate regions are implicitly detected, as shown in Figs. 4b, 4c, 4d, and 4e.

After the L_0 minimization, the set of $\{\mathbf{k}_{ij}\}$ is a good indicator of the joints which have apparent motions between this anchor frame and the previous one. Based on this observation, we use the following formulation to update $\{r_{ij}\}$

$$r'_{ij} = \begin{cases} 0.1 & \text{if } \mathbf{k}_{ij} \neq \mathbf{0} \\ r_{ij} & \text{otherwise.} \end{cases} \tag{15}$$

Here, r'_{ij} is the value after update. Notice that with more joint regions detected, more r_{ij}s are set to small values, so the total strength of smooth constraint becomes smaller. To avoid the diminishing of the smooth constraint, we increase the coefficient of smooth energy term by $\alpha_{\text{smo}}^{\text{new}} = \alpha_{\text{smo}}(\sum r_{ij})/(\sum r'_{ij})$. Thus the total smooth energy does not decrease, while the smooth constraint becomes relatively stronger in the non-joint regions. With the newly updated $\{r_{ij}\}$, the L_2 minimization integrates more information about the distribution of articulate motions on the 3D model, thus is more effective in reconstructing motions. From our experiments, we see that after several L_0 optimizations, we are able to detect all motion joints and then the L_2 optimization alone is able to reconstruct both articulate motions and non-rigid motions. This saves much computation time compared to our preliminary solution. The details are demonstrated in the result section.

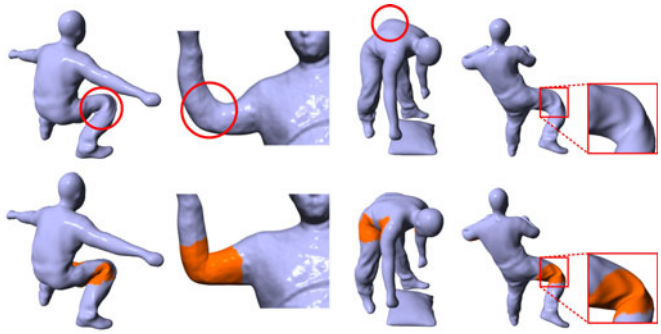


Fig. 5. Comparison of L_0 and L_2 based motion regularization on some anchor frames. The first row shows the tracking results of using L_2, while the second row shows the results of using L_0. The vertices with non-zero motion difference ($\mathbf{k}_{ij} \neq \mathbf{0}$) in the first L_0 iteration are marked orange.

It is also important to note, after the L_0 minimization, the articulate motions are well reconstructed while other non-rigid motions are removed. To reconstruct those non-rigid motions, we run the L_2 based non-rigid registration with the updated joint location information. As the L_2 based non-rigid registration is able to jointly estimate articulate motions and the rest non-rigid motions with correct joint location information, the newly refined result has got rid of the accumulated error of the non-rigid tracking and thereby achieves better results.

Some results on the effectiveness of our proposed L_0 regularization are illustrated in Fig. 5 and the secondary supplementary video, available online. Compared with the traditional non-rigid registration (the top row) which smoothly blends the relative deformation across the human body joints, our L_0 based regularizer (the second row) effectively concentrates these motions to the right joints, and thereby substantially removes the deformation artifacts.

4.3 Anchor Frame Detection

As stated in Section 4.2, since the articulate motions between two neighbor frames are usually small, the pruning based L_0 regularization may incorrectly prune the articulate motions, causing the ineffectiveness of the L_0 optimization. Our key idea to overcome this problem is to accumulate motions of every frame from the previous anchor frame

$$\widetilde{\mathbf{A}}_i^t = \mathbf{A}'_i * \widetilde{\mathbf{A}}_i^{t-1}, \qquad \widetilde{\mathbf{t}}_i^t = \mathbf{t}'_i + \widetilde{\mathbf{t}}_i^{t-1}, \tag{16}$$

where $\{\mathbf{A}'_i, \mathbf{t}'_i\}$ and $\{\widetilde{\mathbf{A}}_i^t, \widetilde{\mathbf{t}}_i^t\}$ denote the current and accumulated motion of node i at time t, respectively. With the accumulation, if the object is performing some articulate motion, the spatial motion variation around the joint of the articulate motion will become larger and larger while the spatial motion variation caused by other non-rigid deformation stays at the same level. By analyzing the distribution of the spatial motion variation, we detect an anchor frame that has large enough articulate motions. The L_0 regularization is then triggered and the pruning algorithm in Section 4.2 is

performed on the detected anchor frame by referring to the previous anchor frame.

In practice, we calculate the variance for all $\|\mathbf{D}x_{ij}\|_2$, where $\mathbf{D}x_{ij}$ is calculated by the accumulated motion $\{\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{t}}_i\}$. If the variance is larger than θ at a particular frame, we set this frame as an anchor frame where the L_0 based motion regularization will be performed. The value of θ in $[0.01, 0.03]$ usually gives reasonable results, while smaller or larger value may bring artifacts. In all our experiments, we set θ to be 0.02. Our supplementary material, available online, shows all the detected anchor frames in several motion sequences.

Notice that the primary goal of detecting an anchor frame and performing L_0 based regularization is to locate new joints on the tracked object. As a consequence, if articulate motion happens on an detected joint, there is no need to locate the joint again. So we check $\|\mathbf{D}x_{ij}\|_2$ only for regions that are not regarded as joints at the current step. With this manner, the number of anchor frames is further reduced which lead to less computation time for handling an input sequence.

4.4 Backward Tracking and Surface Refinement

After refining the tracking result on the newly detected anchor frame, we need to update the frames between the previous anchor frame and the current anchor frame since the L_2 based registration in the forward tracking does not involve the newly detected joints in $\{r_{ij}\}$. To achieve this, we perform a backward L_2 based non-rigid tracking with the newly updated joint location information.

Notice that the number of joints is always limited for any deforming object. When all the joints are located by the L_0 based regularization, there will be no new anchor frames detected and this backward tracking will no longer be triggered, which will also lead to a fast processing of an input sequence.

After the tracking of each input frame, we further reconstruct surface details of the captured objects. To achieve this, we first subdivide the current mesh model and then utilize the method in [46] to synthesize surface details from the captured depth. This step will also keep the final results to be consistent in the temporal domain. After the detail refinement, we take the result of current anchor frame as an initialization to perform L_2 non-rigid tracking for the following frames and detect the next anchor frame. Such tracking cycle goes on until the last detected anchor frame.

5 Experiments

We recorded 12 test sequences consisting of over 7,000 frames using a single Kinect 2.0 camera or an Intel IVCam camera. The Kinect camera is used for capturing full human body motions while the IVCam camera is for capturing hand motions and facial expressions. During data capture, the camera remains fixed. Table 1 shows the details of our captured data. The experiment sequences include fast human motions, e.g. "Sliding" and "SideKick", multiple kinds of objects, e.g. "Puppet" "Pillow_1" "Pillow_2" "Face" and "Hand", and motions with heavy occlusions, e.g., "Pillow_2" "Hand" and "Occlusion". Besides, we also use Vicon data and synthesized data with and without noise for quantitative evaluation.

TABLE 1
Statistics of the Captured Dataset in the Experiments

	No. Frames	No. Anchors	No. Vertices	No. Nodes	Source
Dance	800	4	9,427	260	Kinect
Kongfu	752	8	8,734	249	Kinect
$Pillow_1$	623	5	10,446	249	Kinect
$Pillow_2$	419	4	9,848	281	Kinect
Puppet	800	2	9,995	206	Kinect
Sliding	800	8	8,734	249	Kinect
Girl	800	5	9,501	270	Kinect
SideKick	400	6	8,378	239	Kinect
Face	400	2	9,850	299	IVCam
Hand	300	5	8,923	260	IVCam
Elbow	500	2	9,803	278	Kinect
Occlusion	500	2	10,337	289	Kinect

After data capture, our motion reconstruction method is performed offline. The template modeling step reconstructs a mesh model with about 9,000 vertices. After roughly aligning the template with the first depth frame by sample-based global optimization method [1], the tracking system runs with about 5 frames per minute. For each frame, about 9s is taken by the non-rigid registration. The L_0 based refinement requires 60s for one frame. Notice that we implemented our method in C++ on a PC with an 3.20 GHz 4-core CPU and 16 GB memory.

5.1 Reconstruction Results

Our technique is capable to reconstruct various kinds of motions of different objects, including human body motions, hand motions and their interaction with objects. Some of the results are demonstrated in Fig. 6, where the first column shows the results of pure body motions in the "Sliding" sequence, which indicates that our technique is capable for reconstructing fast motions and handling self-occlusion caused by articulate motions. The second column contains one result of the "Pillow_1" sequence with human-object interactions, where the actor is manipulating a non-rigid pillow. The third column demonstrates human motion with loose cloth. Together with the successful tracking of the human face and the hand motion in Fig. 18 and the results in Fig. 1, it is demonstrated that our method supports various object types with different shapes and topologies, regardless of the existence of articulate structure or not. Our method is also well compatible with surface detail reconstruction method, see the sophisticated geometry obtained on the "Girl" models. For more sequential reconstruction showing the temporal coherency, please refer to our supplementary videos, available online.

5.2 Comparison

We first compare our method with [46] and [47], two state-of-the-art methods which perform the similar task as we do. For quantitative comparison, we use Vicon motion capture system to record the ground truth motions of some sparse markers. And to further achieve dense comparisons, we use pre-reconstructed performance capture data as the ground truth and synthesize single view depth as the input of the three methods. For qualitative comparisons, we run these methods on real captured depth data and render the results

Fig. 6. Results of our technique. For each result, we show a color image, the input depth and the reconstruction result. Notice that the color image is only for viewing the captured motion. It is not used by our system.

for visual comparisons. Recently, [48] and [49] also demonstrated impressive results on dynamic reconstruction, so we also compare our method with these two. Please notice that these two methods have a slightly different goal from ours that they do not use initial templates and do not aim to handle fast and complex motions. In addition, we compare our results with those produced by [33] which tracked body motions using a single RGBD camera based on a regression model trained by several different poses. Finally, we compare our results with the original version [11]. In general, the new version achieves similar results in handling ordinary motions but achieves performance improvement and gives better results in some challenging articulate motions, including repeating motions and occluded motions. We will demonstrate this in the end of this section.

5.2.1 Comparison with [46] and [47]

To achieve the comparison on motion capture data, we first synchronize Vicon and Kinect using infrared flash, and manually register markers of Vicon system with vertices on the template mesh. Then for each frame, after reconstructing the meshes by the three methods, we calculate the average L_2 norm error between the markers and the corresponding vertices. Accumulative error distribution curves for all the three methods are shown in Fig. 7a. Average numerical error of our method on short time range (before frame 400) is 3.08 cm, compared with 4.88 cm of [46] and 12.79 cm of [47]. For longer time range (after frame 400), the average errors of the three methods are 3.86, 7.37, and 17.24 cm respectively, which indicates that our system handles error accumulation much better compared to [46] and [47]. This is due to the L_0 optimization that refines the tracking results on the detected anchor frames and the improved L_2 optimization on the other frames. Visual comparison of this experiment is shown in Fig. 8. We see the artifacts and tracking failures of [46] and [47] in the selected frames.

To quantitatively measure the reconstruction errors on dense surface points, we use motion sequences captured by [1] as the ground truth. We render depth maps from a fixed viewpoint as the input of different reconstruction methods. The reconstruction errors on one frame are demonstrated in Fig. 9, with accumulative error distributions shown in Fig. 7b and average errors shown in Table 2 (the first row). From these comparisons, we see our solution generates better results. This improvement benefits from the L_0 based motion regularizer, which dramatically constrains the motion space and makes it possible to reconstruct complex motions from the limited input. Note that the errors are measured using all the vertices of the body model (not only the visible ones), in which lots of occluded body parts are inferred by the methods. As the motion space is well constrained by the L_0 optimization, we give more plausible results in these parts.

We also add 5 times synthetic noise generated by the Kinect noise model [53] on the ground truth depth. The accumulative error distributions are shown in Fig. 7c and comparisons of average errors are shown in Table 2 (the

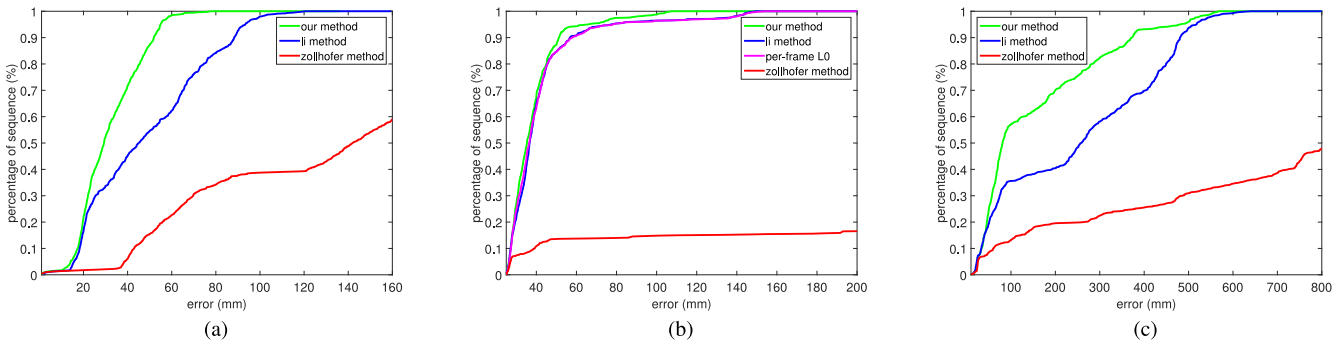


Fig. 7. Comparisons of accumulative error distributions. (a) Accumulative error distributions on Vicon sequence. The x-axis represents the average distance between vertices and their ground truth marker positions; y-axis shows the accumulative error distributions. (b) Accumulative error distributions of different methods using a synthesized depth sequence without noise; (c) accumulative error distributions using the synthesized depth sequence with 5 times Kinect noise. In figures (b) and (c), x-axis represents the average vertex error; y-axis shows the accumulative error distribution.

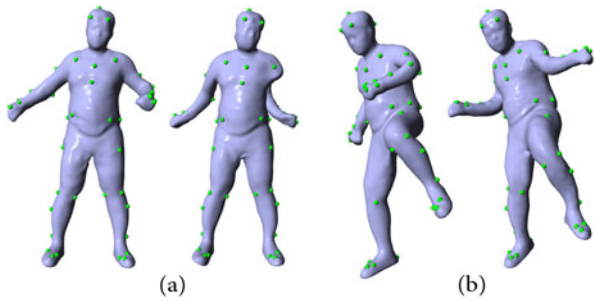


Fig. 8. Visual comparison on Vicon data. (a) Our result (left) and that of [46] (right) on frame 506; (b) our result (left) and that of [47] (right) on frame 625. The green balls represent vertices corresponding to markers in Vicon system.

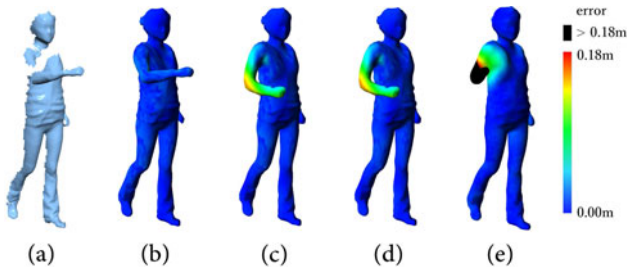


Fig. 9. Visual comparison on a synthetic sequence. (a) Input depth; (b) result of our method; (c) result of per-frame L_0 minimization; (d) result of [46]; (e) result of [47]. The color coded per-vertex error is the euclidean distance between a vertex and its ground truth position.

TABLE 2
Comparisons of Average Errors (mm) on Synthetic Sequences

Method	[46]	[47]	Per-frame L_0	Proposed
without noise	42.1	789.6	42.1	**37.4**
with noise	253.4	814.5	247.2	**153.8**

These errors are obtained by averaging the per-frame errors of all frames in the sequences.

second row). The visual comparison of the three methods on this synthetic experiment is presented in Fig. 10. These experiments indicate that our method generates better tracking results under such large amount of noise, which validates our robustness to noisy input.

In Fig. 11, we visually compare our method with [46] and [47] on real captured data. From the comparison, we see that our method outperforms [46] on the left foot, while [47] fails to track this pose caused by fast motion. In Fig. 18, we compare our method with [46] on face, body and hand sequences. Since there is no evident articulate motion in the face sequence, our method is similar to [46]. However, on articulate sequences of body and hand, our method prevents tracking failures and local misalignment which appear in the results of [46]. More comparisons on motion sequences are shown in the supplementary videos, available online.

5.2.2 Comparison with [48] and [49]

We also compare our algorithm with the latest methods, Newcombe et al. [48] and Dou et al. [49]. Figs. 16 and 17 demonstrate that their methods fail in reconstructing the dynamic scenes of *Sliding* and *Hand*. This is also claimed as a limitation in [48]. On the contrary, our technique robustly handles fast motion due to the L_0 regularization. We also

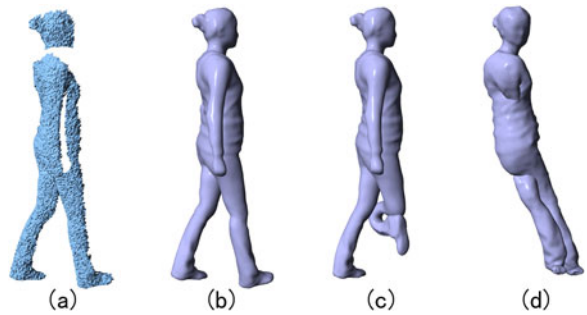


Fig. 10. Visual comparison on a synthetic sequence with 5 times Kinect noise. (a) Input depth; (b) result of our method; (c) result of [46]; (d) result of [47]. Note that [47] has lost tracking of this sequence in previous frames.

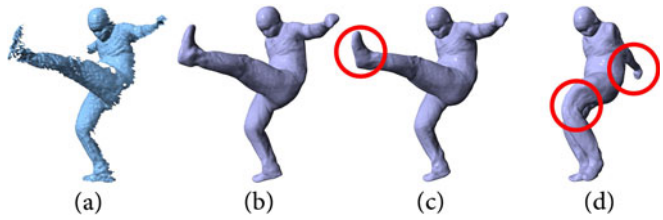


Fig. 11. Visual comparison on *Kongfu* sequence. (a) Input depth; (b) result of our method; (c) result of [46]; (d) result of [47].

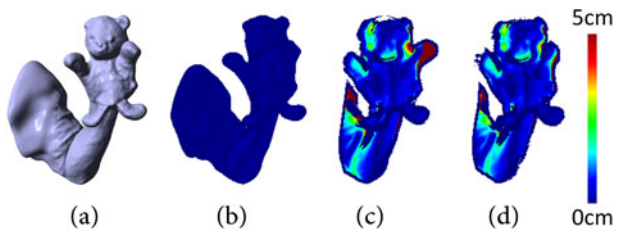


Fig. 12. Comparison of error distribution on sequence *Puppet*. (a) Presents the ground truth geometry; (b-d) present maps of error distributions of our method, [48] and [49], respectively.

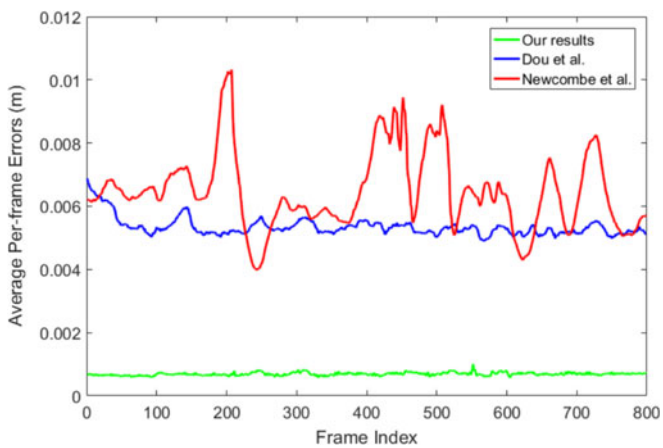


Fig. 13. Comparison of per-frame average numerical errors on sequence *Puppet*. Green, red and blue curves present pre-frame average numerical errors of our method, [48] and [49], respectively.

quantitatively compare our algorithm with [48] and [49]. Fig. 12 shows the error distributions on *Puppet* sequence, and Fig. 13 presents average numerical errors of the three methods. Our method achieves substantially lower numerical errors than the other two methods. Notice that [48] and [49] do not aim to the same goal as we do. They mainly focus on reconstructing a scene without initial templates while we

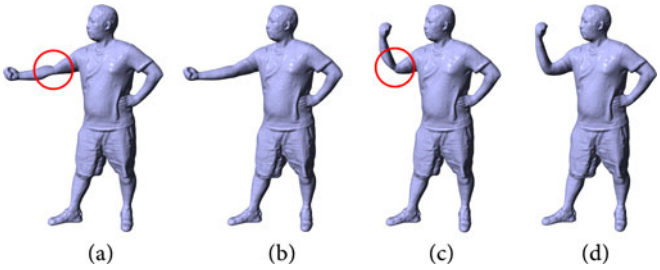


Fig. 14. Comparison with [11] on repeating motions. (a, c) Results of the method in [11]; (b, d) results of our method.

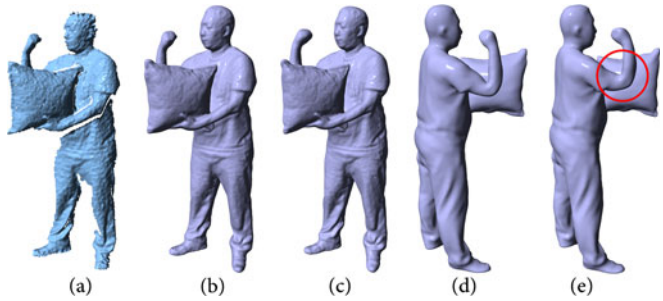


Fig. 15. Comparison with [11] on occlusion motions. (a) Presents captured depth map; (b) and (d) present the reconstructed results of our method on visible and invisible regions, respectively; (c) and (e) present reconstructed results of [11] on visible and invisible regions, respectively. Note [11] generates an unnatural bent elbow in (e).

use templates to reconstruct complex and fast motions in the scene. The comparisons corresponding to Figs. 16 and 17 are also included in the primary video.

5.2.3 Comparison with [11]

Frequently repeating motions may cause error accumulation in the reconstructed articulate motions in [11]. As our improved L_2 based method performs unified optimization to simultaneously reconstruct articulate and non-rigid motions after joint region detection, we get better results in handling this kind of motions. Fig. 14 shows the reconstructed pose for the eighth elbow bending motion in the "Elbow" sequence. Notice that the accumulated errors in [11] lead to noticeable artifacts while our method gives reasonable results. The sequence result is in the primary supplementary video, available online.

Our method also achieves more physically correct results on invisible regions of the reconstructed scene. In Fig. 15, a performer bends his arm, and the arm gets occluded by a pillow in the later part of the sequence. Due to the occlusion, [11] produces unnatural bending motion around the performer's elbow. Our method has detected the joint of the elbow in previous unoccluded frames and this information is exploited in these later frames by low smooth weights around the elbow, thus generating more correct reconstructed results.

Besides the quality of the result, our method also achieves better performance than [11]. First, we reduce the number of L_0 optimization to $2 \sim 8$ times for all our motion sequences, which saves at most 30 times of L_0 optimization (about 30 minutes) at most. Furthermore, after all joint regions detected, the time consuming bidirectional tracking does not need to be performed and is replaced by a forward tracking strategy. This means up to half of the computation time is saved. Numerical comparisons on computation time of the entire sequence are shown in Table 3. Comparisons on L_0 computation time of each sequence are presented in Table. 4.

5.2.4 Comparison with [33]

We compare our method with [33] which also uses a single depth camera to track body motions. [33] trains a regression model for surface deformations based on a group of complete models (usually 8 models) with different poses of the same performer wearing the identical clothes. For this comparison, we select 8 models from our results of "*Sliding*" and "*Sidekick*" sequences (the first row in Fig. 19). In these two sequences, the same performer wears the identical clothes, and these 8 models have the same mesh topology and represent different articulated motions of the performer. Based on the regression model trained by these poses, we track the body motions using the same non-rigid registration method as in [33]. Due to the fast motions and intense depth noise, [33] fails to reconstruct the motions around the performer's feet. By contrast, benefitting from

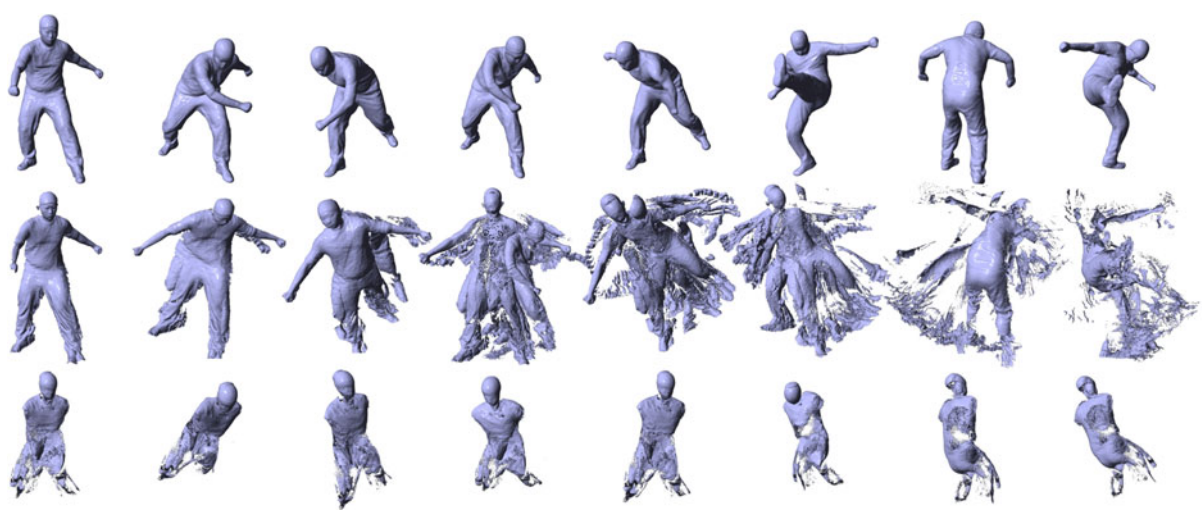


Fig. 16. Comparison of our method with [48] and [49] on sequence *Sliding*. Results of each row are arranged from left to right in chronological order. The 1st, 2nd and 3rd rows present results of our method, [48] and [49] respectively.

Fig. 17. Comparison of our method with [48] and [49] on sequence *Hand*. Results of each row are arranged from left to right in chronological order. The 1st, 2nd and 3rd rows present results of our method, [48] and [49], respectively.

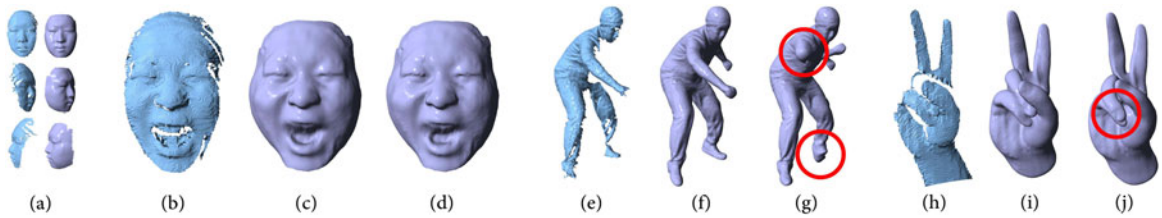


Fig. 18. Visual comparisons with [46] on *face*, *hand* and *sliding* sequences. (b, e, h) Depth input; (c, f, i) reconstruction results of our method; (d, g, j) reconstruction results of [46]; (a) represents the first frame of *face* sequence and its template rendered at 3 different views. Note that the Intel IVCam sensor only captures the front face of the performer. Since we track the facial expression using a general non-rigid deformation method and do not exploit facial priors, the unobserved side face of the performer only moves with the front face due to the smooth term of the energy, thus generating some unnatural deformations on both sides of the performer's face.

the ℓ_0 sparse constraint and improved ℓ_2 tracking method, our algorithm produces better results. The corresponding results are also demonstrated in the primary video.

5.3 Validation

In this section, we validate some key components of our method. First, we evaluate the effectiveness of the anchor frame detection by comparing it with a naive strategy that uses all frames as anchors. Furthermore, we test how robust our method works against the results of the anchor frame detection. As our system relies on initial templates, we then evaluate how the accuracy of the templates influences the results of our method. Finally, as it is also applicable to use L_1 minimization to replace L_0 used in our method, we compare the reconstructed results of the two strategies.

5.3.1 Anchor Frame Detection

Figs. 9c, 7b and Table 2 have demonstrated the reconstruction errors of using all frames as anchors. Due to the small movement between two consecutive frames, the L_0 scheme can not distinguish the articulate motion, thus all motions

TABLE 3
Performance Comparisons with [11] on the Entire Sequence

	Our method		[11]	
Seq.	L_0 frames	Total time	L_0 frames	Total time
Dance	4	297 mins	16	415 mins
Kongfu	8	317 mins	26	432 mins
Pillow_1	5	202 mins	9	341 mins
Pillow_2	4	135 mins	5	209 mins
Puppet	2	310 mins	2	378 mins
Sliding	8	375 mins	35	497 mins
Girl	5	331 mins	14	441 mins
SideKick	6	144 mins	17	232 mins
Face	2	171 mins	2	212 mins
Hand	5	120 mins	6	164 mins
Elbow	2	153 mins	9	269 mins

TABLE 4
Performance Comparisons with [11] on L_0 Optimization

Seq.	Our method	[11]
Dance	233 secs	941 secs
Kongfu	445 secs	1,615 secs
Pillow_1	285 secs	522 secs
Pillow_2	228 secs	289 secs
Puppet	114 secs	131 secs
Sliding	465 secs	2,236 secs
Girl	279 secs	855 secs
SideKick	371 secs	1,035 secs
Face	97 secs	99 secs
Hand	315 secs	354 secs
Elbow	117 secs	575 secs

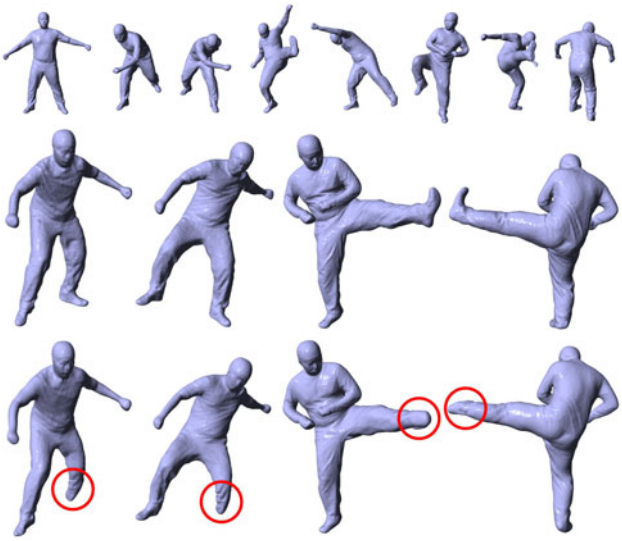


Fig. 19. Comparison of our algorithm with [33] on "*Sliding*" and "*Sidekick*" sequences. The 1st row represents 8 models selected from our results of the two sequences. These models present difference articulated motions of the performer and are used to train a regression model for surface deformations. The performer in both of the sequences wears the identical clothes, and the models have the same mesh topology. The 2nd and 3rd rows demonstrate the results of our method and [33], respectively.

are pruned so that the geometry model remains fixed after L_0 optimization. Therefore, with the following L_2 optimization, the performance of per-frame L_0 is similar to traditional L_2 regularization.

To evaluate the robustness of our method against anchor frame selection, we randomly shift anchor frames around their original positions. Results of some selected frames are shown in Fig. 20. From our experiments, we see that with 10 frames shifted for anchor frames, our method always gives reasonable results, which indicates that our method is insensitive to anchor frame selection.

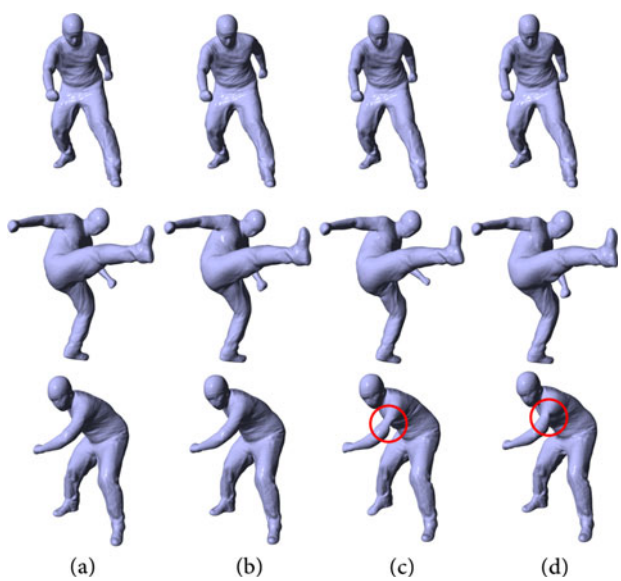


Fig. 20. Evaluating the robustness to anchor frame selection. (a) Results of original anchor frame selection; (b) results of randomly shifting anchor frames by 10 frames around their positions; (c, d) results of randomly shifting by 20 and 30 frames respectively. The three rows correspond to frame 150, 447 and 595 of *Kongfu* sequence respectively.

Fig. 21. Evaluating the robustness to the quality of initial templates. (a) Original template (top) and 75 percent (middle) and 50 percent (bottom) reconstructed ones respectively. (b)-(d) Selected results of different poses.

But severe changes of anchor frames may violate articulate motion assumption and generate artifacts, as shown in the last row of (c) and (d) in Fig. 20. Notice that the original anchor frames are usually about 50 frames apart in a normal speed motion sequence, so 10 frames shift is relatively a large shift.

5.3.2 *Initial Template*

We evaluate our method using initial templates of different qualities, reconstructed by 75 and 50 percent of the original model. We test our method using these templates on "Sliding" sequence. Some frames of the reconstructed results are shown in Fig. 21. For the 75 and 50 percent reconstructed templates, all articulate motions are well reconstructed while only synthesized details appear to be a little different to the result using the original template, indicating that our method tolerates considerable smoothness and does not always require high quality templates.

5.3.3 L_1 *Optimization*

We compare L_1 sparsity constraint with the proposed L_0 method. The difference is in Eqn. (10), where the L_1 regularizer is $E'_{\text{reg}} = \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} \|\mathbf{D}x_{ij}\|_1$. We solve it using primal-dual internal point method [54]. The comparison results are shown in Fig. 22. Our L_0 solver reconstructs motion in joint regions more accurately and avoids artifacts.

5.4 Other Types of Depth Input

In addition to the data captured by a single consumer depth sensor, our technique is also applicable for other depth acquisition techniques such as structure light [46] and binocular cameras [31]. This provides the extensive practicalities and enables more appealing applications. Results are shown in the supplementary video, available online.

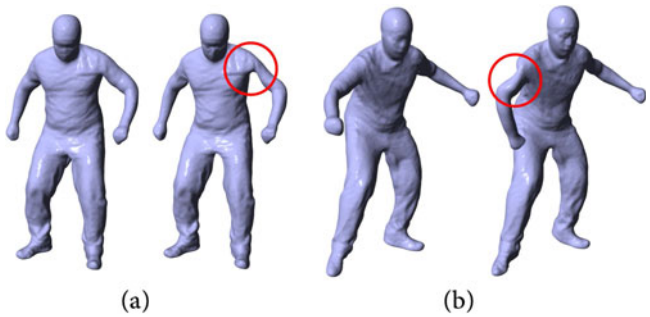


Fig. 22. Comparing L_1 minimization with the proposed L_0 minimization. Left images in (a) and (b) are our L_0 results and right ones are approximation of L_1.

5.5 Limitations

The proposed L_0-L_2 non-rigid tracking approach is still limited in tracking extremely fast motions. For instance, the supplementary video, available online, shows a failure case that the tracking cannot catch up the up-moving leg of a character. This is mainly because of the frangibility of the vertex-to-point matching in dealing with fast motions. Our method is also incapable of motions with serious or long term occlusions. However, it naturally supports multi-view depth input, which will effectively mitigate the occlusion challenge. Unlike [48] and [49], we propose a template-based motion tracking technique. The topology is predefined by the template, so we are not able to handle topology changes.

6 Discussion

We have presented a novel non-rigid motion tracking method using only a single consumer depth camera. Our method outperforms the state-of-the-art methods in terms of robustness and accuracy. The key contribution of our technique is the combined L_0-L_2 tracking strategy which takes advantage of the intrinsic properties of articulate motions to constrain the solution space. According to experiment results, our method outperforms four previous state-of-the-art non-rigid reconstruction algorithms and can robustly capture full body human motions using a single depth sensor without skeleton embedding.

Our L_0 regularization is performed on the result of non-rigid registration but is not limited to specific algorithms for getting the results, which means it can be flexibly applied to other non-rigid registration techniques for better reconstructions.

Acknowledgment

This work was supported by the National key foundation for exploring scientific instrument No. 2013YQ140517, the open funding project of state key laboratory of virtual reality technology and systems of Beihang University (Grant No. BUAA-VR-14KF-08), and NSFC (No. 61671268, 61522111, and 61531014). Feng Xu is the corresponding author.

References

[1] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1746–1753.

[2] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and C. Theobalt, "Markerless motion capture of interacting characters using multi-view image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1249–1256.

[3] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2088–2095.

[4] M. de La Gorce, D. J. Fleet, and N. Paragios, "Model-based 3D hand pose estimation from monocular video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1793–1805, Sep. 2011. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.33

[5] I. Baran and J. Popovic, "Automatic rigging and animation of 3D characters," *ACM Trans. Graph.*, vol. 26, no. 3, 2007, Art. no. 72.

[6] A. Jacobson, I. Baran, J. Popovi, and O. Sorkine, "Bounded biharmonic weights for real-time deformation," *ACM Trans. Graph.*, vol. 30, pp. 1–8, 2011.

[7] R. Szeliski and S. Lavallée, "Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines," *Int. J. Comput. Vis.*, vol. 18, no. 2, pp. 171–186, 1996.

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

[9] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Proc. 5th Eurographics Symp. Geometry Process.*, 2007, pp. 109–116.

[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, 2012, Art. no. 188.

[11] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai, "Robust non-rigid motion tracking and surface reconstruction using l0 regularization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3083–3091.

[12] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[13] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Comput. Vis. Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[14] M. Waschbüsch, S. Würmlin, D. Cotting, F. Sadlo, and M. Gross, "Scalable 3D video of dynamic scenes," *Visual Comput.*, vol. 21, no. 8–10, pp. 629–638, 2005.

[15] J. Starck and A. Hilton, "Surface capture for performance-based animation," *Comput. Graph. Appl.*, vol. 27, no. 3, pp. 21–31, 2007.

[16] C. Budd, P. Huang, M. Klaudiny, and A. Hilton, "Global non-rigid alignment of surface sequences," *Int. J. Comput. Vis.*, vol. 102, no. 1–3, pp. 256–270, Mar. 2013. [Online]. Available: http://dx.doi.org/10.1007/s11263-012-0553-4

[17] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur, "Markerless garment capture," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. no. 99.

[18] 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, Art. no. 98.

[19] C. Cagniart, E. Boyer, and S. Ilic, "Free-form mesh tracking: A patch-based approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 1339–1346.

[20] C. Cagniart, E. Boyer, and S. Ilic, "Probabilistic deformable surface tracking from multiple videos," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 326–339.

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

[22] J. Starck and A. Hilton, "Model-based human shape reconstruction from multiple views," *Comput. Vis. Image Understanding*, vol. 111, no. 2, pp. 179–194, 2008.

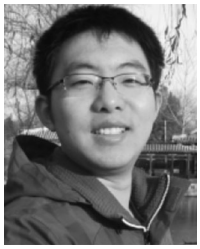
[23] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt, "Performance capture of interacting characters with handheld kinects," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 828–841.

[24] M. Dou, H. Fuchs, and J.-M. Frahm, "Scanning and tracking dynamic objects with commodity depth cameras," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2013, pp. 99–106.

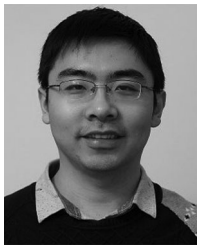
[25] M. Dou, et al., "Fusion4D: Real-time performance capture of challenging scenes," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, Art. no. 114.

[26] Y. Zhu, D. Huang, F. De La Torre, and S. Lucey, "Complex non-rigid motion 3D reconstruction by union of subspaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 1542–1549.

[27] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Proc. Consum. Depth Cameras Comput. Vis.*, 2013, pp. 71–98.
[28] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3D pose estimation from a single depth image," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 731–738.
[29] Y. Chen, Z.-Q. Cheng, C. Lai, R. Martin, and G. Dang, "Realtime reconstruction of an animating human body from a single depth camera," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 8, pp. 2000–2011, Aug. 2016.
[30] M. Ye, H. Wang, N. Deng, X. Yang, and R. Yang, "Real-time human pose and shape estimation for virtual try-on using a single commodity depth camera," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 4, pp. 550–559, Apr. 2014.
[31] C. Wu, C. Stoll, L. Valgaerts, and C. Theobalt, "On-set performance capture of multiple actors with a stereo camera," *ACM Trans. Graph.*, vol. 32, no. 6, 2013, Art. no. 161.
[32] M. Ye and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2353–2360.
[33] Q. Zhang, B. Fu, M. Ye, and R. Yang, "Quality dynamic human body modeling using a single low-cost depth camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 676–683.
[34] F. Bogo, M. J. Black, M. Loper, and J. Romero, "Detailed full-body reconstructions of moving people from monocular RGB-D sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2300–2308.
[35] G. K. Tam, et al., "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 7, pp. 1199–1217, Jul. 2013.
[36] W. Chang and M. Zwicker, "Automatic registration for articulated shapes," *Comput. Graph. Forum*, vol. 27, no. 5, pp. 1459–1468, 2008.
[37] W. Chang and M. Zwicker, "Range scan registration using reduced deformable models," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 447–456, 2009.
[38] Y. Pekelny and C. Gotsman, "Articulated object reconstruction and markerless motion capture from depth video," *Comput. Graph. Forum*, vol. 27, no. 2, pp. 399–408, 2008.
[39] H. Li, R. W. Sumner, and M. Pauly, "Global correspondence optimization for non-rigid registration of depth scans," *Comput. Graph. Forum*, vol. 27, no. 5, pp. 1421–1430, 2008.
[40] M. Wand, et al., "Efficient reconstruction of nonrigid shape and motion from real-time 3D scanner data," *ACM Trans. Graph.*, vol. 28, no. 2, 2009, Art. no. 15.
[41] N. J. Mitra, S. Flory, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann, "Dynamic geometry registration," in *Proc. Symp. Geometry Process.*, 2007, pp. 173–182.
[42] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev, "3D self-portraits," *ACM Trans. Graph.*, vol. 32, no. 6, Nov. 2013, Art. no. 187.
[43] M. Bojsen-Hansen, H. Li, and C. Wojtan, "Tracking surfaces with evolving topology," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 53–1, 2012.
[44] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong, "Modeling deformable objects from a single depth camera," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 167–174.
[45] T. Popa, I. South-Dickinson, D. Bradley, A. Sheffer, and W. Heidrich, "Globally consistent space-time reconstruction," *Comput. Graph. Forum*, vol. 29, pp. 1633–1642, 2010.
[46] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, 2009, Art. no. 175.
[47] M. Zollhöfer, et al., "Real-time non-rigid reconstruction using an RGB-D camera," *ACM Trans. Graph.*, vol. 33, no. 4, 2014, Art. no. 156.
[48] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 343–352.
[49] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi, "3D scanning deformable objects with a single RGBD sensor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 493–501.
[50] Q. Zhou, S. Miller, and V. Koltun, "Elastic fragments for dense scene reconstruction," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 473–480.
[51] J. Gall, J. Potthoff, C. Schnörr, B. Rosenhahn, and H.-P. Seidel, "Interacting and annealing particle filters: Mathematics and a recipe for applications," *J. Math. Imag. Vis.*, vol. 28, no. 1, pp. 1–18, 2007.
[52] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, 2011, Art. no. 174.
[53] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3D reconstruction and tracking," in *Proc. 2nd Int. Conf. 3D Imag. Model. Process. Vis. Transmiss.*, 2012, pp. 524–530. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/3DIMPVT.2012.84
[54] K. A. McShane, C. L. Monma, and D. Shanno, "An implementation of a primal-dual interior point method for linear programming," *ORSA J. Comput.*, vol. 1, no. 2, pp. 70–83, 1989.



Kaiwen Guo received the BS degree in the Automation Department, Northeastern University, Shenyang, China, in 2011. He is currently working toward the PhD degree in the Automation Department, Tsinghua Univeristy.



Feng Xu received the BS degree in physics from Tsinghua University, Beijing, China, in 2007 and the PhD degree in automation from Tsinghua University, Beijing, China, in 2012. He is currently an assistant professor in the School of Software, Tsinghua University. His research interests include face animation, performance capture, and 3D reconstruction.

Yangang Wang received the BE degree in instrument measurement from Southeast University, Nanjing, China, in 2009 and the PhD degree in automation from Tsinghua University, Beijing, China, in 2014. He is currently an associate researcher with Microsoft Research Asia. His research interests include motion capture and animation, 3D reconstruction, and image processing.

Yebin Liu received the BE degree from the Beijing University of Posts and Telecommunications, China, in 2002 and the PhD degree from the Automation Department, Tsinghua University, Beijing, China, in 2009. He is currently an associate professor with Tsinghua University. He was a research fellow in the Computer Graphics Group of the Max Planck Institute for Informatik, Germany, in 2010. His research areas include computer vision, computer graphics, and computational photography. He is a member of the IEEE.

Qionghai Dai received the MS and PhD degrees in computer science and automation from Northeastern University, Shenyang, China, in 1994 and 1996, respectively. He is currently a professor in the Department of Automation and the director of the Broadband Networks and Digital Media Laboratory, Tsinghua University, Beijing. He has authored or co-authored more than 200 conference and journal papers and two books. His research interests include computational photography and microscopy, computer vison and graphics, intelligent signal processing. He is associate editor of the *Journal of Visual Communication and Image Representation*, the *IEEE Transactions on Neural Networks and Learning Systems*, and the *IEEE Transactions on Image Processing*. He is a senior member of the IEEE.