BodyFusion: Real-time Capture of Human Motion and Surface Geometry Using a Single Depth Camera

Tao Yu[12], Kaiwen Guo[2], Feng Xu[2], Yuan Dong[2], Zhaoqi Su[2], Jianhui Zhao[1], Jianguo Li[3],
Qionghai Dai[2], Yebin Liu[2]
[1]Beihang University, Beijing, China
[2]Tsinghua University, Beijing, China
[3]Intel Labs China, Beijing, China

Abstract

We propose BodyFusion, a novel real-time geometry fusion method that can track and reconstruct non-rigid surface motion of a human performance using a single consumer-grade depth camera. To reduce the ambiguities of the non-rigid deformation parameterization on the surface graph nodes, we take advantage of the internal articulated motion prior for human performance and contribute a skeleton-embedded surface fusion (SSF) method. The key feature of our method is that it jointly solves for both the skeleton and graph-node deformations based on information of the attachments between the skeleton and the graph nodes. The attachments are also updated frame by frame based on the fused surface geometry and the computed deformations. Overall, our method enables increasingly denoised, detailed, and complete surface reconstruction as well as the updating of the skeleton and attachments as the temporal depth frames are fused. Experimental results show that our method exhibits substantially improved non-rigid motion fusion performance and tracking robustness compared with previous state-of-the-art fusion methods. We also contribute a dataset for the quantitative evaluation of fusion-based dynamic scene reconstruction algorithms using a single depth camera.

1. Introduction

Recently, volumetric depth fusion methods for dynamic scene reconstruction, such as DynamicFusion [27], VolumeDeform [18] and Fusion4D [11], have attracted considerable attentions from both academia and industry in the fields of computer vision and computer graphics. The dynamic fusion module in such a reconstruction method enables quality improvements over temporal reconstruction models in terms of both the accuracy and completeness of the surface geometry. This beneficial achievement further enables reconstruction systems to bypass the need for a complete model template and allows them to be operated in real time for VR/AR applications such as holoportation [28]. Among all of these works, fusion methods using a single depth camera [27, 18] are low in cost and easy to set up and therefore show more promise for popularization. However, current techniques based on a single depth camera are still restricted to handling slow and controlled motions because of their relative lack of observations (single view), computational resources (real time) and geometry priors (no template).

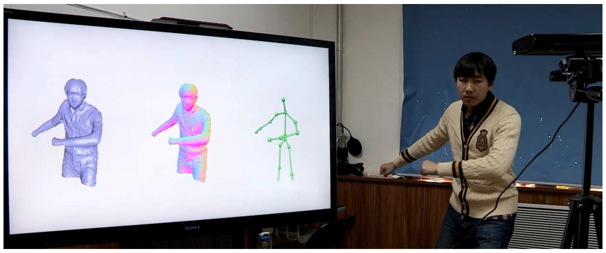

Figure 1: Our system and real-time reconstruction results.

Aiming at a more robust real-time fusion method for dynamic scenes, we make two observations. On the one hand, the human body is usually the core element of many non-rigid dynamic scene in which we are interested. On the other hand, as indicated by previous works [14], human motion largely follows articulated structures, and thus, articulated motions (in terms of the skeleton) can be extracted from non-rigid motion as a prior, and used to constrain any large non-rigid deformations to be consistent with the skeletal joints, thereby reducing the optimization space and the range of physically plausible deformations. Based on these observations, we propose BodyFusion, i.e., a skeleton-embedded surface fusion (SSF) approach, to improve the reconstruction quality of a dynamic human motion. Our SSF method jointly optimizes on the skeleton and the graph-nodes for dynamic surface fusion. Note that in our method, the articulated skeleton is automatically detected and introduced, without losing any of the single-view, real-time and template-free features of available systems or requiring any additional manual operations.

On one hand, integrating the articulated body prior into the dynamic fusion framework assists in the reconstruction of human motions; on the other hand, including a non-rigid surface registration method in a skeleton-based tracking technique improves the quality of fused geometry. Therefore, it is reasonable to combine both of these methods in a uniform framework. However, designing a real-time algorithm to take advantage of both merits of these two aspects is still an unstudied problem. Moreover, in skeleton-embedded surface tracking method the skin attachments need to be calculated only once for the first frame, whereas in SSF, the skin attachments need to be repeatedly updated over time because of the increasing updating of the surface. However, previous methods (*e.g.*, [2]) are not applicable to recalculate the attachments for each frame in real-time.

With the intent of solving the above problems, we have carefully designed our BodyFusion system, to achieve the fully automatic, real-time fusion of natural human motion and surface geometry using a single depth video input. Specifically, we make the following technical contributions in this paper.

- The BodyFusion system is presented based on the proposed skeleton-embedded surface fusion (SSF) approach, which outperforms other recent dynamic fusion techniques [27] and [18] in handling human body motions, and enables the more convenient, real-time generation of a full-body 3D self-portrait using a single depth camera.

- A cooperative deformation scheme with a novel binding term is proposed to bridge the skeleton deformation and the graph node deformation, and to combine the merits of these techniques for a better surface tracking performance.

- An efficient skin attachment updating scheme is presented that can be executed in less than 0.5 ms for a frame. The output attachments provide the high level semantic information to assist in the design of the smoothness term used in the cooperative deformation step.

- We contribute a dataset captured using a marker-based Mocap system for the quantitative evaluation of dynamic fusion algorithms.

2. Related Work

Skeleton-based surface reconstruction. Although a variety of methods focus on skeletal motion capture [34, 35, 42, 5], we restrict this review to research on the reconstruction of dynamic surface geometries and motion. Most skeleton-based surface reconstruction algorithms require a skeleton-embedded mesh template (scanned or using human SCAPE model [16]) for surface tracking [39, 13, 30]. The skeletal structure was then embedded in the template, and the attachments between the skeleton and the surface vertices are computed once before the tracking process. The use of such a skeleton representation greatly reduces the solution space for the surface vertices and enables the tracking of fast and natural motions [23]. Multiview RGB video [13], binocular video [44], multiview depth video [45] and single-view depth video [46, 3] can all serve as input to guide the surface tracking process. In this work, we use a skeleton prior but do not use a pre-scanned template model for dynamic surface reconstruction.

Non-rigid surface tracking. Most non-rigid surface tracking approaches require a mesh template. They deform the scanned mesh template in a frame-by-frame manner to approximate the input from general non-rigid targets. By solving for the deformation parameters of sparsely sampled nodes on a mesh template, Li et al. [19] tracked a low-resolution shape template and updated the geometric details based on the input depth sequence. Guo et al. [14] introduced the ℓ_0 regularizer to implicitly detect articulated body parts in the tracking of general non-rigid scenes. Their method achieved more robust and accurate tracking performance but suffered in terms of run-time efficiency. Using GPU-accelerated optimization, Zollhofer et al. [47] demonstrated the first real-time non-rigid tracking method. They used pre-scanned mesh templates to track the slow, non-rigid motions of multiple objects in real time.

By taking advantage of a shape prior for the target, non-rigid surface tracking can also be conducted with a focus on particular kinds of objects; examples include face tracking [43, 4, 6] and hand tracking [17, 31]. This line of research also shares some similarities with skeletal motion capture of the human body [34, 35, 42] when detailed geometry reconstruction is not considered.

Simultaneous tracking and reconstruction of dynamic scenes. To recover both the geometry and motion from a dynamic scene, most related methods need to integrate temporal information through non-rigid registration. In recent years, many non-rigid alignment methods have been proposed, *e.g.*, linear variational deformation in [22], embedded deformation in [36, 20], and subspace deformation in [40]. Some studies have narrowed their research focus to articulated motions alone. [7, 8] adopted a reduced deformable model and linear blend skinning to align partial scans such that a complete model could be generated from the entire sequence. In addition to articulated motions and integrated geometry, [29] also reconstructed a dynamic skeleton from detected rigid body parts. Besides the above works, variant other methods have been proposed, *e.g.*, reconstruction of 4D spatio-temporal surface [25, 37], incompressible flows [33], animation cartography [38], full body quasi-rigid motions [21], full body with larger motions [12] and dynamic reconstruction using 8 Kinects [12]. However, none of the above methods runs in real time.

To date, several non-rigid motion and geometry reconstruction methods have achieved real-time frame rates. Assuming general non-rigid motions, DynamicFusion [27] tracks dynamic scene motions and non-rigidly fuses a static model in a reference coordinate frame. [18] exploits SIFT features in color images for dynamic scene reconstruction. [15] can reconstruct the geometry, appearance and motion of dynamic scene using rgbd input, they utilize the reconstructed surface albedo to construct a shading-based scheme that can significantly improve the motion tracking performance. [11] uses 8 pairs of customized RGBD cameras to achieve the reconstruction of complex dynamic scenes in real time. The real-time methods described above assume general non-rigid deformation with regard to the scene motion. However, in most cases, these non-rigid motions include articulated motions, such as human body motion. Our proposed method explicitly uses this prior to constrain the frame-to-frame deformation, thereby improving the reconstruction quality for fast, natural body motion.

3. Overview

Similarly to DynamicFusion, BodyFusion processes a sequence in order, frame by frame. Fig. 2 illustrates the pipeline of our proposed skeleton-embedded surface fusion (SSF) method for each depth frame. The main differences from the DynamicFusion pipeline are the attachment update step and the cooperative deformation step, the purpose of which is to solve for both the skeleton deformation and the graph-node deformation. Note that, in the cooperative deformation step, a binding term is introduced to leverage the advantages of these two kinds of deformation techniques. Also note that the attachment update step and the cooperative deformation step are designed to assist each other, thereby leading to a better tracking performance and better depth integration. Specifically, the output attachments serve in the design of the smoothness term in the energy function for cooperative deformation, while the obtained deformation information is fed back to update the attachments.

For the first frame, our method automatically embeds a detected skeleton in the reference frame (Sec. 4.1). As illustrated in Fig. 2, we calculate the attachments, $i.e.$, the vertex-to-bone weights (Sec. 4.2), for the canonical model $\mathcal{C}_{t-1}$ given the deformation parameters of the skeleton and the graph nodes in the previous frame. For each new depth frame $\mathcal{D}_t$, the system solves for both the skeleton and graph-node deformation parameters (Sec. 4.3) using a novel joint optimization procedure. Here, the attachments $\mathcal{H}_{t-1}$ computed and fed into the joint deformation module to define the smoothness term. We implement an efficient Gauss-Newton method to solve the joint deformation problem to achieve real-time performance (Sec. 4.4). After registering the object to the current depth, we non-rigidly fuse the depth information into the reference frame (Sec. 4.5) to obtain a new canonical model $\mathcal{C}_t$. The results of this gradual geometry integration in the reference frame are used for processing in the next frame.

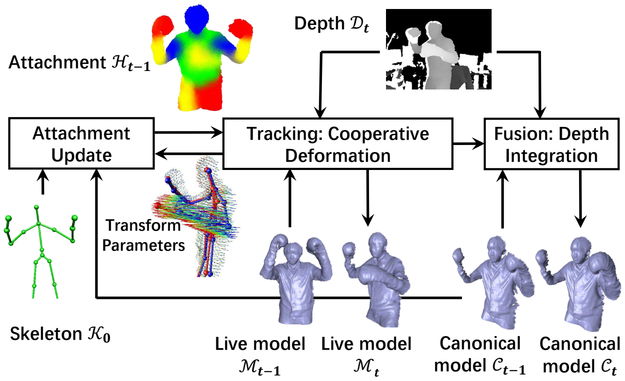
Attachment $\mathcal{H}_{t-1}$
Depth $\mathcal{D}_t$
Attachment Update
Tracking: Cooperative Deformation
Fusion: Depth Integration
Transform Parameters
Skeleton $\mathcal{K}_0$
Live model $\mathcal{M}_{t-1}$
Live model $\mathcal{M}_t$
Canonical model $\mathcal{C}_{t-1}$
Canonical model $\mathcal{C}_t$

Figure 2: The pipeline of our system.

4. Method

In this section, we first introduce the method used to initialize the skeleton embedding. Then, we introduce the attachment updating scheme, followed by the formulation and solving of the joint optimization for both the skeleton and graph-node deformation parameters. Finally, we describe the method used to integrate the depth information into the reference volume.

4.1. Initialization

Our system uses a single depth camera. The input to our pipeline is a depth sequence $\mathcal{D} = [D_1, \ldots, D_n]$. For the first frame, following [27], we rigidly integrate the depth information into the reference volume, extract a triangular surface from the reference volume using the marching cube algorithm [24], uniformly sample deformation nodes on the surface and construct a node graph to describe the non-rigid deformation. To search for nearest-neighboring nodes, we also perform a dense k-NN field in the reference volume. Afterward, we use the automatically detected skeleton (for example, using the Kinect SDK) with 3D joint positions and embed it into the same reference volume; then, we calculate the initial skin attachments as detailed in Sec. 4.2. Note that we assume a fixed camera location and treat any camera motion as global rigid object motion.

4.2. Attachment Calculation

Given an incomplete mesh surface and an embedded skeleton, the skin attachment problem, or the skinning problem, is the specification of bone weights for the vertices, i.e., the extent to which each bone transform affects each

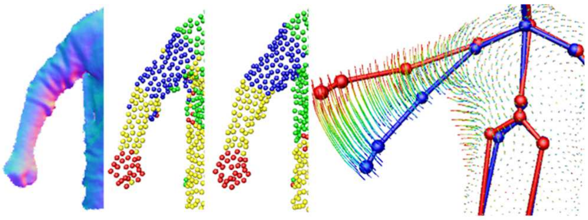

Figure 3: Motion criterion in attachment calculation. (a) Canonical surface normal. The most closely related bone for each node without using the motion criterion (b) and using the proposed motion criterion (c), with node color represented as its most closely related bone's color. Note that there are error node-bone assignments on the arm. (d) Motion field for calculation of (c). Red color represents the end position.

vertex. Most SST methods [39, 13] utilize fixed attachments for tracking throughout the entire sequence based on a complete-surface template, whereas in our case, the attachments must be updated frame by frame as the surface is gradually completed. Previous skinning method [2] formulates skin attachment calculation as a heat equilibrium problem on the mesh surface and uses the final vertex temperature as the skinning weight. However, it is a time-consuming process to solve such a heat equilibrium problem. Even using a GPU solution, [10] still takes several seconds to calculate the skin attachment, let alone the task of continuously updating the skin attachments, as is necessary in our real-time system. Therefore, we propose a simple but effective method to calculate skin attachment. The key to the success of this attachment strategy is that it leverages the deformation (motion) information from the tracking step (Sec. 4.3).

For each node n_i in the node graph, the skinning weights can be represented as a coefficient vector $\mathcal{H}_{n_\mathrm{i}} = [\eta_{i,1}, \dots, \eta_{i,m}]$. Here, m is the number of bones. For sake of efficiency, we first calculate the node-to-bone weights and then interpolate the vertex-to-bone weights using the node-to-bone weights of the k nearest-neighboring nodes. Overall, we use three criteria, i.e., distance, normal and motion, to find the most closely related bone for each node n_i. Specifically, for each node n_i, we check the nearest bone one by one until the node-bone pair satisfies the normal and motion criteria. The normal criterion requires that the bone-to-node direction for the selected bone is consistent with the surface normal of the node; it is formulated as $\langle \mathbf{e}_{ij}, \mathbf{n}_{n_\mathrm{i}} \rangle < \theta$. Here, $\mathbf{e}_{ij}$ is the normalized direction from the nearest point on the bone to n_i, and $\mathbf{n}_{n_\mathrm{i}}$ is the surface normal on node n_i. θ is a specified threshold and set to 0.5.

Ambiguity remains even with the use of both the distance and normal criteria, and attachment errors can still occur in surface regions with complex normal distributions, such as the winkles of loose clothes of the arm shown in Fig. 3. We therefore incorporate the motion information from the cooperative deformation step. Intuitively, if a node is located on a bone segment, then that node and that bone are closely related and usually share similar motion, as shown in Fig. 3(d). The motion criterion is thus defined as $|\mathbf{T}_{bj}\mathbf{x}_i - \mathbf{T}_{ni}\mathbf{x}_i| < t$, where $\mathbf{T}_{bj}$ is the accumulated motion of the jth bone, $\mathbf{T}_{ni}$ is the accumulated non-rigid motion of node n_i, and t is set to 0.015.

Based on the above criteria, we select the most closely related bone for each graph node and set the corresponding weight $\eta_{i,j}$ to a binary value of 1.0 if the jth bone is most closely related to n_i or 0.0 otherwise. Note that because of our hard constraints, a most closely related bone will not be found for every node; we therefore smooth the node attachments by averaging $\mathcal{H}_{n_\mathrm{i}}$ over the node graph using 8 neighbors. We perform this step iteratively until attachment weights have been assigned to all nodes. In practice, 2 iterations are sufficient. Moreover, we filter each node's attachment using temporal neighborhoods with the temporal window size set to 5.

Finally, we interpolate the vertex-to-bone weights as follows:

$$\mathcal{H}_{v_\mathrm{i}} = \frac{1}{Z} \sum_{k \in \mathcal{N}(v_\mathrm{i})} \lambda_{i,k} \mathcal{H}_{nk}, \tag{1}$$

where $\mathcal{H}_{v_\mathrm{i}}$ is the attachment of the ith vertex, the $\mathcal{N}(v_i)$ are the neighboring nodes of the ith vertex, Z is the normalization coefficient, and $\lambda_{i,k}$ is the spatial weight describing the influence of the kth node on v_i and is defined as $\lambda_{i,k} = \exp(-\|\mathbf{v}_i - \mathbf{x}_k\|_2^2/(2\sigma_k))$. Here, $\mathbf{v}_i$ and $\mathbf{x}_k$ are the 3D coordinates of the ith vertex and the kth node, and $\sigma_k = 0.025$ is the given influence radius of the kth node. As shown in Fig. 3(c), our simplified skinning method can generate reasonable attachment weights with a negligible run-time overhead of less than $0.5ms$.

4.3. Tracking : Cooperative Deformation

In this subsection, we introduce a novel joint optimization procedure for both the graph-node and skeleton parameters. By exploiting the underlying articulated structure, our method constrains the solution space and rapidly converges to an accurate pose. The energy function of our cooperative deformation problem is defined as follows:

$$E_\mathrm{t} = \lambda_n E_\mathrm{nonrigid} + \lambda_s E_\mathrm{skeleton} + \lambda_g E_\mathrm{graph} + \lambda_b E_\mathrm{binding}, \tag{2}$$

where E_nonrigid and E_skeleton represent the errors of the data fitting driven by the node graph and the skeleton, respectively; E_graph denotes the as-rigid-as-possible spatial constraint enforced by the node graph; and E_binding enforces consistency between the graph-node and skeleton deformations.

Data terms E_nonrigid and E_skeleton. We use a point-toplane formulation for the data fitting terms [32], as follows:

$$\begin{aligned} E_{\text{nonrigid}} &= \sum_{(v_{\text{i}},u_{\text{i}})\in\mathcal{P}} \left|\hat{\mathbf{n}}_{v_{\text{i}}}\left(\hat{\mathbf{v}}_i - \mathbf{u}_i\right)\right|^2, \\ E_{\text{skeleton}} &= \sum_{(v_{\text{i}},u_{\text{i}})\in\mathcal{P}} \left|\widetilde{\mathbf{n}}_{v_{\text{i}}}\left(\widetilde{\mathbf{v}}_i - \mathbf{u}_i\right)\right|^2, \end{aligned} \tag{3}$$

where $\mathcal{P}$ represents the set of correspondences, $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{n}}_{v_{\text{i}}}$ represent vertex coordinates and its normal warped by the k nearest nodes (in our experiments, we set k to 4) and is defined as follows:

$$\hat{\mathbf{v}}_i = \sum_{j\in\mathcal{N}(v_{\text{i}})} \omega_{i,j}\mathbf{T}_{nj}\mathbf{v}_i, \quad \hat{\mathbf{n}}_{v_{\text{i}}} = \sum_{j\in\mathcal{N}(v_{\text{i}})} \omega_{i,j}\mathbf{T}_{nj}\mathbf{n}_{v_{\text{i}}}. \tag{4}$$

Here, $\mathbf{T}_{nj}$ is the $\mathbf{SE}(3)$ of the deformation associated with the jth node and $\omega_{i,j}$ is the weight with which the jth node influences v_i and is defined similarly to $\lambda_{i,k}$ in Eqn. 1. $\widetilde{\mathbf{v}}_i$ and $\widetilde{\mathbf{n}}_{v_{\text{i}}}$ are those warped by the skeleton, defined as:

$$\widetilde{\mathbf{v}}_i = \sum_{j\in\mathcal{B}} \eta_{i,j}\mathbf{T}_{bj}\mathbf{v}_i, \quad \widetilde{\mathbf{n}}_{v_{\text{i}}} = \sum_{j\in\mathcal{B}} \eta_{i,j}\mathbf{T}_{bj}\mathbf{n}_{v_{\text{i}}}, \tag{5}$$

where $\mathbf{T}_{bj}$ is the $\mathbf{SE}(3)$ of the deformation associated with the jth bone of the skeleton and is defined by the exponential map of a twist ζ_i, *e.i.*, $\mathbf{T}_i = \exp(\hat{\zeta}_i)$; $\eta_{i,j}$ is the attachment weight with which the jth bone influences v_i.

The kinematic chain is defined similar to [13] with a twist of 3 rotation degree of freedom for each bone. The transformation of a vertex can be represented by cascading transformations of each parent up to the root of the skeleton, please refer to [26] for details.

Smoothness term E_{graph}. E_{graph} is the local as-rigid-as-possible smoothness constraint imposed on neighboring graph nodes. Because of the single-view depth input, half of the object is invisible to the camera. This regularization term has the ability to drive the invisible regions to move with the observed regions by generating smooth deformations. However, enforcing a spatially uniform smoothness over the entire graph may produce deformation artifacts. As shown in Fig. 4(c), even when the motion in the current frame is considered for the detection and concentration of the deformations [27], bent surfaces on bone regions still appear.

In this paper, to take advantage of the intrinsic articulated motion components of non-rigid human motion, we propose a novel smoothness constraint based on the attachment information obtained in Sec. 4.2. As shown in Fig. 4(d), the attachments on the canonical model provide information of the part information, which imply regions of possibly discontinuous body motion. This attachment information can be converted into a smoothness term (as shown in Fig. 4(e)) and added into the optimization, without losing control over other detailed non-rigid surface deformations. The attachment-based smoothness function between node i and node j is defined as follows:

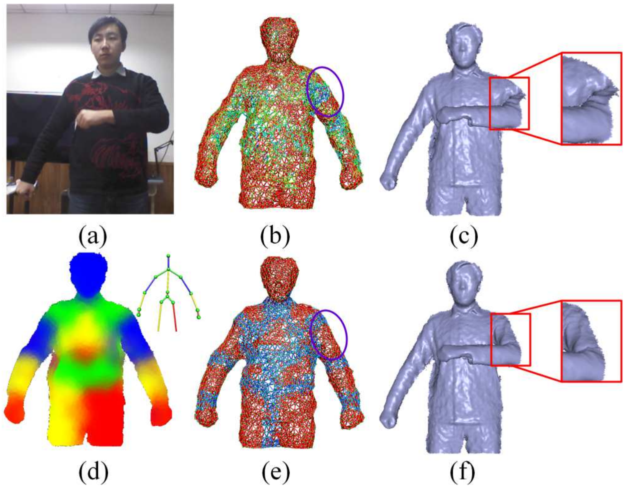

Figure 4: Smoothness term defined based on the attachment information: (a) reference color image; (b) the smoothness term used in DynamicFusion [27] in the canonical frame, computed from the motion energy in the previous frame and filtered using the Huber function; (c) the tracking result in the current frame obtained using the smoothness term (b); (d) our attachments computed as described in Sec 4.2; (e) our smoothness term based on the attachments (d); (f) our tracking result. The red, green and blue colors of the edges between nodes in (b) and (e) represent large, median and small smoothness value, respectively.

$$\psi_{\text{reg}}(\mathcal{H}_{ni}, \mathcal{H}_{nj}) = \rho\left(\|\mathcal{H}_{ni} - \mathcal{H}_{nj}\|_2^2\right), \tag{6}$$

where $\rho(\cdot)$ is a Huber kernel with a threshold of 0.2. Therefore, this smoothness term can be formulated as

$$E_{\text{graph}} = \sum_i \sum_{j\in\mathcal{N}(i)} \psi_{\text{reg}}(\mathcal{H}_{ni}, \mathcal{H}_{nj})\|\mathbf{T}_{ni}\mathbf{x}_j - \mathbf{T}_{nj}\mathbf{x}_j\|_2^2. \tag{7}$$

Fig. 4(f) shows the natural reconstruction of an actor's left arm achieved using our proposed smoothness term.

Binding term E_{binding}. To guarantee consistent deformations driven by both the nodes and the skeleton, we propose a binding term, E_{binding}, as follows:

$$E_{\text{binding}} = \sum_{i=1}^{N} \|\mathbf{T}_{ni}\mathbf{x}_i - \widetilde{\mathbf{x}}_i\|_2^2, \tag{8}$$

where N is the number of nodes, $\mathbf{x}_i$ denotes the coordinates of the ith node, and $\widetilde{\mathbf{x}}_i$ represents the coordinates warped by the skeleton kinematics. This binding term enforces similar deformations of both the nodes and the skeleton.

Our SSF method contains two kinds of structures for deformation, i.e., the tree structure introduced by the skeleton deformation and the graph structure implied by the non-rigid surface deformation. These two kinds of structures have their own characteristics. In general, the skeleton tree structure enables easier capture of large motion but is more sensitive to erroneous and noisy feature correspondences on

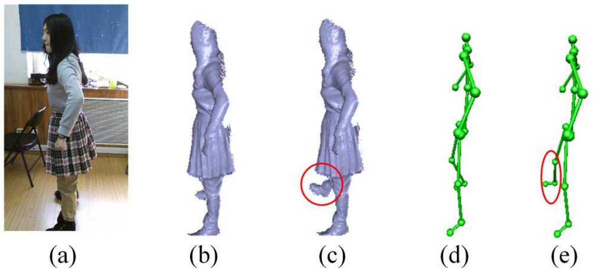

Figure 5: Evaluation of the binding term. (a) reference color image; (b,d) geometry integration and skeleton optimization result using the binding term; (c,e) geometry integration and skeleton optimization result without using the binding term.

the branches and end joints. Contrarily, the graph node structure is more tightly connected with uniformly spaced nodes, and therefore tend to be more robust, smooth and rigid. The binding term connects and take the advantages of both of them.

Without using the binding term, there will be no relationship between the skeleton deformation and the non-rigid deformation, thus the pipeline has to sequentially and independently perform these two deformations. From Fig. 5, we can see such pipeline will easily get failed when severe occlusion happens (e.g. small numbers of error correspondences drives the failure of skeleton tracking) due to the incorrect skeleton tracking results. In contrast, by including the binding term, the non-rigid deformation in our joint optimization can regularize the skeleton result, providing a reasonable reconstruction.

Eqn. 2 presents a non-linear least-squares problem. We initialize the unknowns using the values from the previous frame and simultaneously solve for both the skeleton and non-rigid embedded node deformation parameters in a projective ICP framework. For the first several ICP iterations, we use a relatively large λ_b value to track large-scale body motions. During the course of the ICP procedure, we gradually relax λ_b to improve the finer-scale fitting of details. In each iteration, we minimize the energy function (Eqn. 2) using the Gauss-Newton method. To achieve real-time performance, we implement an efficient Gauss-Newton solver on a GPU, as introduced in the next section.

4.4. Efficient Gauss-Newton Solver

In each step of the Gauss-Newton procedure, we linearize the energy function around the currently estimated deformation parameters using a first-order Taylor expansion. We use a twist representation for both the bone and node transformations. Therefore, the linearized transformations can be formulated as $\mathbf{T}_{ni} = \mathbf{I} + \hat{\zeta}_i$ and $\mathbf{T}_{bi} = \mathbf{I} + \theta_0\hat{\xi}_0 + \sum_{k\in\mathcal{K}_{\mathrm{i}}} \theta_k\hat{\xi}_k$, where both $\hat{\zeta}_i$ and $\hat{\xi}_i$ are 4×4 skew matrices [26]. After obtaining a fully linearized system, we solve the normal equation using the preconditioned conjugate gradient (PCG) method, in a manner similar to [47, 11].

Using the same method as in [11], we directly construct $\mathbf{J}^{\mathrm{T}}\mathbf{J}$ and $\mathbf{J}^{\mathrm{T}}\mathbf{f}$ on a GPU to avoid repeated calculations of non-zeros in the Jacobian matrix. After constructing the normal equation, we adopt the kernel-merged PCG method [41] to implement an efficient GPU linear solver and use a block-diagonal preconditioner to improve the speed of convergence.

4.5. Depth Integration

After cooperative optimizing the node and skeleton parameters, we non-rigidly integrate the current depth information into the reference volume and uniformly sampling the newly added surface to update the nodes [27].

However, this non-rigid integration method is subject to some ambiguities. If several voxels are warped to the same depth surface in the camera frame, then the TSDF [9] of all of these voxels will be updated. To resolve this ambiguity, we adopt the method presented in [11] and use a stricter strategy. If two voxels in the reference frame are warped to positions separated by a distance of larger than a given threshold $\epsilon = 0.02$, then we reject their TSDF integration. This method avoids the generation of erroneous surfaces due to voxel collisions.

5. Results

In this section, we describe the overall performance of our system and details of its implementation, followed by the qualitatively comparisons and evaluations. We have captured more than 10 sequences with different actors/actress performing natural body motions like "*Dancing*", "*Marching*" and "*Body Building*", etc. We have also captured two sequences with marker suites (14 markers in total) under Vicon [1] system for quantitatively evaluation.

Fig. 6 shows some of our reconstruction results. The first row in this figure shows 6 sequential reconstructed models of a input sequence. Note that the geometry is continuously refined, with faithful tracking of the natural body motion. Our BodyFusion system enables convenient and real-time 3D self-portrait. As shown in the left bottom result of Fig. 6, the target person needs only to take a turn round then a full body 3D model can be obtained. Compared with [21], our system bypass the need of a motor tilt for data capture and tens of minutes for computation.

5.1. Performance

Our system is fully implemented on a GPU and runs at 32 ms per frame. The cooperative deformation requires 19 ms; the TSDF integration requires 5 ms; the precomputation of the k-NN field, the uniform sampling or updating of the nodes, the construction of the node graph and the attachment calculation collectively require 6 ms; the preprocessing of the depth information (including bilateral filtering and calculation of the depth normals) requires

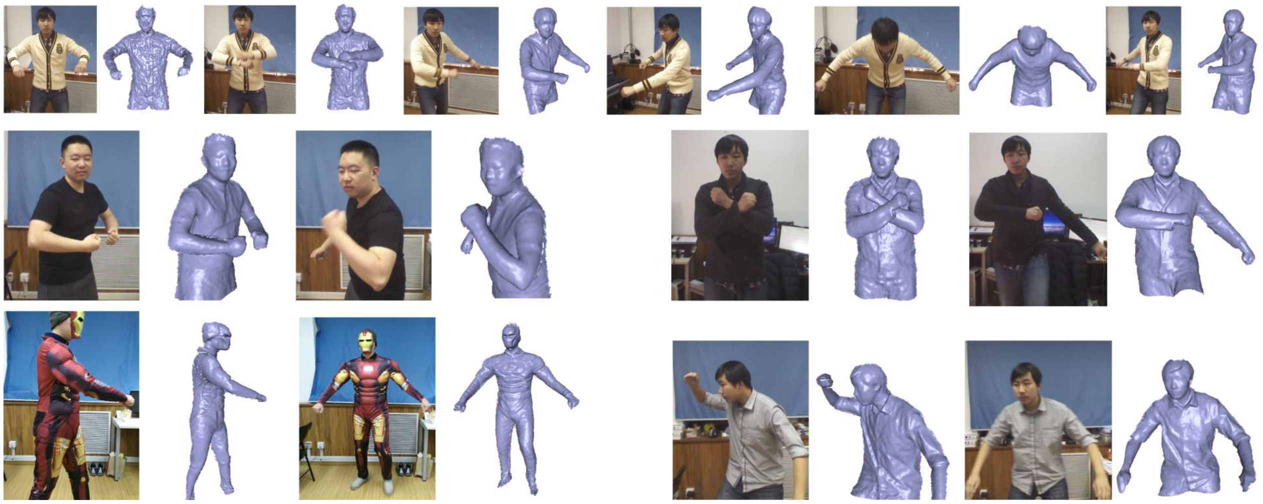

Figure 6: Selected results reconstructed by our system. The first row shows 6 sequential frames of a single sequence; the other rows show two of the reconstructed frames for each selected sequence.

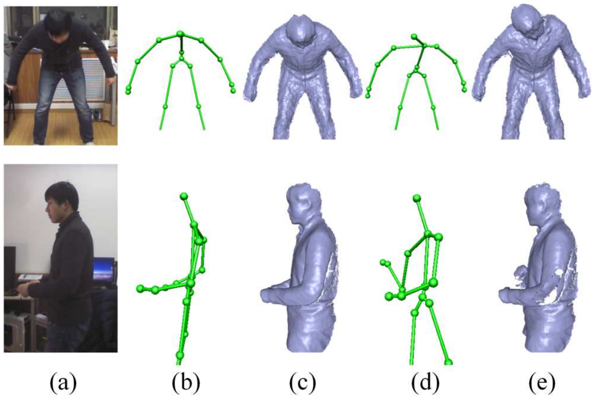

Figure 7: Comparison of our results with those produced using skeletons detected with the Kinect SDK followed with our cooperative deformation method. (a) image, (b) our optimized skeleton, (c) our fused results, (d) the skeletons detected with the Kinect SDK, (e) results using (d) as input to our cooperative deformation.

1 ms; and the rendering of the results requires 1 ms. For the cooperative deformation, we execute projective ICP 3 times. To solve the normal equation, we terminate the PCG procedure after 10 iterations. For all of our experiments, we set $\lambda_n = 1.0$, $\lambda_s = 1.0$, and $\lambda_g = 5.0$. For the first ICP iteration, we set $\lambda_b = 10.0$, and we halve this coefficient for each subsequent iteration. The voxel resolution is 4 mm. For each vertex, we choose the 4 nearest nodes to drive it, and for each node, we use the 8 nearest neighbors to construct the node graph.

5.2. Comparisons and Evaluations

We compare our method with two state-of-the-art methods, *i.e.*, DynamicFusion [27] and VolumeDeform [18]. Fig. 8 shows visual comparisons on 3 sequences. For each sequence, the first column presents the color images for reference, although these images are not used in our system. The 2nd-4th columns show the results of BodyFusion, DynamicFusion and VolumeDeform, respectively. From a comparison of the results, we can see that both DynamicFusion and VolumeDeform fail to reconstruct the body motions in these sequences, whereas our method generates faithful results.

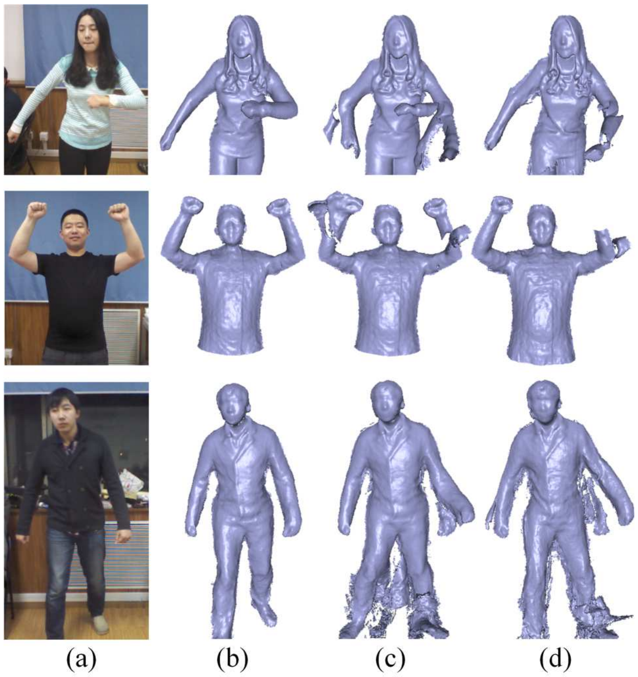

Figure 8: Visual comparisons of the results of our method (b), DynamicFusion [27] (c) and VolumeDeform [18] (d).

To qualitatively compare our reconstructed surfaces with those driven by the skeletal motions detected using the Kinect SDK, we set the skeletal parameters returned by the

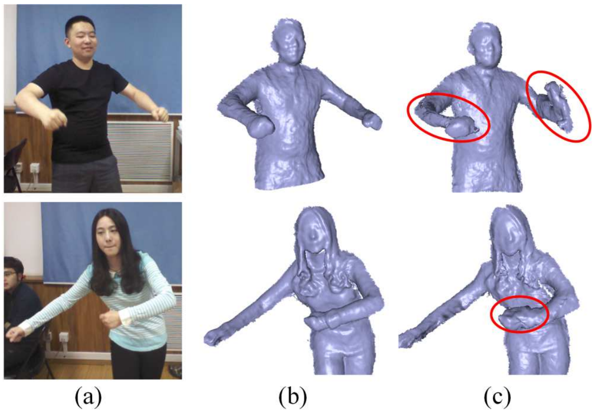

Figure 9: Comparison of the results obtained with (b) and without (c) non-rigid registration.

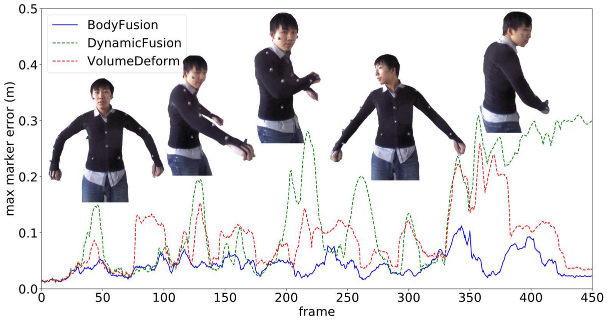

Figure 10: Numerical error curves of our method, DynamicFusion [27] and VolumeDeform [18].

Kinect SDK as the initial values for our joint optimization of each temporal frame. A comparison is presented in Fig. 7. From the figure, we can clearly see that the algorithm initialized using the Kinect SDK fails to reconstruct a good surface when occlusion occurs. The main reason is that the skeletal joint information provided by the Kinect SDK is noisy and temporal inconsistent, especially the very inaccurate joint rotation parameters.

We also evaluate the non-rigid registration part of the energy function (Eqn. 2) in Fig. 9. We eliminate terms related to non-rigid registration and use only E_{skeleton} for tracking. From these experiments, we can see that when non-rigid registration is not included, the system cannot achieve faithful fusion due to the lack of tracking of detailed non-rigid surface motion. Moreover, without a faithfully fused model, the tracking performance also suffers severely.

To quantitatively evaluate our tracking accuracy against the ground truth, we evaluate on two sequences simultaneously captured by the Vicon system and the Kinect. We synchronized the two systems by manually flashing the infrared LED. We first transform the marker coordinates from the Vicon frame into the canonical frame and then calculate the voxel index for each marker and compare the per-frame warped voxel positions with the Vicon-detected ground-truth marker positions. Fig. 10 presents the per-frame maximum error curves of BodyFusion, DynamicFusion [27] and VolumeDeform [18] on one of the sequence. For each depth frame, we calculate the maximum and the average errors of all the markers. We average for all the frames on the entire sequence. Tab. 1 lists the average maximum error and the average error on one of the sequence. From the numerical error curves and the average errors, we can see that our system generates the lowest tracking errors compared with DynamicFusion [27] and VolumeDeform [18].

BodyFusion		DynamicFusion		VolumeDeform	
max	avg	max	avg	max	avg
4.3 cm	2.2 cm	12.7 cm	4.4 cm	8.8 cm	3.7 cm

Table 1: Average numerical errors on the entire sequence.

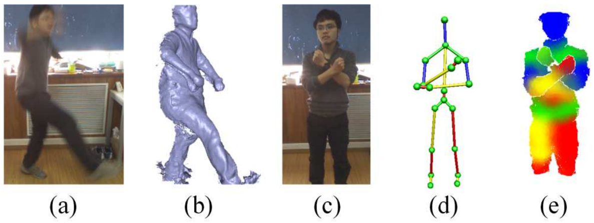

Figure 11: Failure cases of our system. Because of incorrect skeleton detection results, we cannot calculate the correct skin attachments.

6. Limitations and Conclusions

Our system is still limited with regard to the reconstruction of very fast motions because of the blurred depth and the ICP method. Fig. 11(a) and (b) presents a failure case. Moreover, because we currently rely on the detected skeleton in the first frame for skeleton embedding and attachment calculation, our system can fail if the initial pose is difficult to be accurately detected, as shown in Fig. 11(c),(d) and (e).

In this paper, we propose a novel real-time geometry fusion method that can track and fuse non-rigid human motions using a single consumer-grade depth camera. Our main contribution is the skeleton-embedded surface fusion (SSF) approach that performs joint skeleton tracking with non-rigid surface deformation, which enables the production of not only more natural body motions but also surface geometries with fine details. We believe that our system represents a further step toward the wider adoption of consumer-level depth cameras to reconstruct dynamic scenes in real-time. Moreover, we believe that our SSF approach will open the door for the study of leveraging the high-level semantic information in real-time non-rigid surface fusion and dynamic scene reconstruction.

Acknowledgements. This work was supported by the National key foundation for exploring scientific instrument No. 2013YQ140517, NKBRP No.2014CB744201, and the NSFC grant No.61671268, No.61522111, No.61531014, No.61233005.

References

[1] https://www.vicon.com/.

[2] I. Baran and J. Popović. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3), July 2007.

[3] F. Bogo, M. J. Black, M. Loper, and J. Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2300–2308, Dec. 2015.

[4] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM Trans. Graph.*, 32(4):40:1–40:10, July 2013.

[5] K. Buys, C. Cagniart, A. Baksheev, T. D. Laet, J. D. Schutter, and C. Pantofaru. An adaptable system for rgb-d based human body detection and pose estimation. *J. Visual Communication and Image Representation*, 25:39–52, 2014.

[6] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. *ACM Trans. Graph.*, 32(4):41:1–41:10, July 2013.

[7] W. Chang and M. Zwicker. Range scan registration using reduced deformable models. *Comput. Graph. Forum*, 28(2):447–456, 2009.

[8] W. Chang and M. Zwicker. Global registration of dynamic range scans for articulated model reconstruction. *ACM Trans. Graph.*, 30(3):26:1–26:15, May 2011.

[9] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA, 1996. ACM.

[10] O. Dionne and M. de Lasa. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 173–180, New York, NY, USA, 2013. ACM.

[11] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph.*, 35(4):114:1–114:13, July 2016.

[12] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi. *3D scanning deformable objects with a single RGBD sensor*, volume 07-12-June-2015, pages 493–501. IEEE Computer Society, United States, 10 2015.

[13] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1746–1753, 2009.

[14] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3083–3091, Dec. 2015.

[15] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu. Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Trans. Graph.*, 36(3):32:1–32:13, June 2017.

[16] D. Hirshberg, M. Loper, E. Rachlin, and M. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *European Conference on Computer Vision (ECCV)*, LNCS 7577, Part IV, pages 242–255. Springer-Verlag, Oct. 2012.

[17] N. K. Iason Oikonomidis and A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 101.1–101.11, 2011.

[18] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision (ECCV)*, October 2016.

[19] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, pages 175:1–175:10, New York, NY, USA, 2009. ACM.

[20] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *In Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1421–1430, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[21] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3d self-portraits. *ACM Trans. Graph.*, 32(6):187:1–187:9, Nov. 2013.

[22] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong. Modeling deformable objects from a single depth camera. In *The IEEE International Conference on Computer Vision (ICCV)*, 2009.

[23] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and C. Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1249–1256. IEEE, 2011.

[24] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA, 1987. ACM.

[25] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann. Dynamic geometry registration. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 173–182, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[26] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1994.

[27] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[28] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 741–754, New York, NY, USA, 2016. ACM.

[29] Y. Pekelny and C. Gotsman. Articulated Object Reconstruction and Markerless Motion Capture from Depth Video. In *Comput. Graph. Forum*, 2008.

[30] G. Pons-Moll, A. Baak, T. Helten, M. Müller, H. Seidel, and B. Rosenhahn. Multisensor-fusion for 3d full-body human motion capture. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 663–670, 2010.

[31] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1113, 2014.

[32] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.

[33] A. Sharf, D. A. Alcantara, T. Lewiner, C. Greif, A. Sheffer, N. Amenta, and D. Cohen-Or. Space-time surface reconstruction using incompressible flow. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pages 110:1–110:10, New York, NY, USA, 2008. ACM.

[34] J. Shotton, A. Fitzgibbon, , A. Blake, A. Kipman, M. Finocchio, R. Moore, and T. Sharp. Real-time human pose recognition in parts from a single depth image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011.

[35] C. Stoll, N. Hasler, J. Gall, H. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 951–958, 2011.

[36] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. In *SIGGRAPH*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[37] J. Süßmuth, M. Winter, and G. Greiner. Reconstructing animated meshes from time-varying point clouds. In *Comput. Graph. Forum*, volume 27, pages 1469–1476, 2008.

[38] A. Tevs, A. Berner, M. Wand, I. Ihrke, M. Bokeloh, J. Kerber, and H.-P. Seidel. Animation cartography&mdash;intrinsic reconstruction of shape and motion. *ACM Trans. Graph.*, 31(2):12:1–12:15, Apr. 2012.

[39] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 97:1–97:9, New York, NY, USA, 2008. ACM.

[40] 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.*, 28(2):15:1–15:15, May 2009.

[41] D. Weber, J. Bender, M. Schnoes, A. Stork, and D. W. Fellner. Efficient GPU Data Structures and Methods to Solve Sparse Linear Systems in Dynamics Applications. *Comput. Graph. Forum*, 2013.

[42] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph.*, 31(6):188:1–188:12, Nov. 2012.

[43] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. In *SIGGRAPH*, SIGGRAPH '11, pages 77:1–77:10, New York, NY, USA, 2011. ACM.

[44] C. Wu, C. Stoll, L. Valgaerts, and C. Theobalt. On-set performance capture of multiple actors with a stereo camera. *ACM Trans. Graph.*, 32(6):161:1–161:11, Nov. 2013.

[45] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt. Performance capture of interacting characters with handheld kinects. In *European Conference on Computer Vision (ECCV)*. 2012.

[46] M. Ye, Y. Shen, C. Du, Z. Pan, and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. *IEEE Transactions on Pattern Analysis & Machine Intelligence (TPAMI)*, 38(8):1517–1532, 2016.

[47] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Trans. Graph.*, 33(4):156:1–156:12, July 2014.