SegmentedFusion: 3D human body reconstruction using stitched bounding boxes

Yao Shih-Hsuan
National Tsing Hua University
Taipei, Taiwan
s105062556@m105.nthu.edu.tw

Diego Thomas
Kyushu University
Fukuoka, Japan
thomas@ait.kyushu-u.ac.jp

Akihiro Sugimoto
National Institute of Informatics
Tokyo, Japan
sugimoto@nii.ac.jp

Shang-Hong Lai
National Tsing Hua University
Taipei, Taiwan
lai@cs.nthu.edu.tw

Rin-ichiro Taniguchi
Kyushu University
Fukuoka, Japan
rin@kyudai.jp

Abstract

This paper presents SegmentedFusion, a method possessing the capability of reconstructing non-rigid 3D models of a human body by using a single depth camera with skeleton information. Our method estimates a dense volumetric 6D motion field that warps the integrated model into the live frame by segmenting a human body into different parts and building a canonical space for each part. The key feature of this work is that a deformed and connected canonical volume for each part is created, and it is used to integrate data. The dense volumetric warp field of one volume is represented efficiently by blending a few rigid transformations. Overall, SegmentedFusion is able to scan a non-rigidly deformed human surface as well as to estimate the dense motion field by using a consumer-grade depth camera. The experimental results demonstrate that SegmentedFusion is robust against fast inter-frame motion and topological changes. Since our method does not require prior assumption, SegmentedFusion can be applied to a wide range of human motions.

1. Introduction

3D reconstruction using consumer-grade RGB-D cameras (also called RGB-D SLAM) has been extensively researched in the past decades in the computer vision, robotics, and computer graphics communities. The reconstructed 3D models can be used in many applications, such as animation, game, and AR/VR. In general the RGB-D SLAM procedure is composed of two main steps: (1) (non-rigid) motion tracking and (2) data integration.

A popular strategy used in many RGB-D SLAM techniques is to match points in the built 3D model with points on the input depth image to estimate camera pose and 3D model deformation. With a good estimate of the (non-rigid) transformation that aligns the 3D model to the input RGB-D image, depth measurements can be integrated into the built 3D model using the running average strategy. Volumetric depth fusion methods for static and dynamic scene reconstruction have been proposed in [15, 16], and have attracted considerable attentions from both academia and industry.

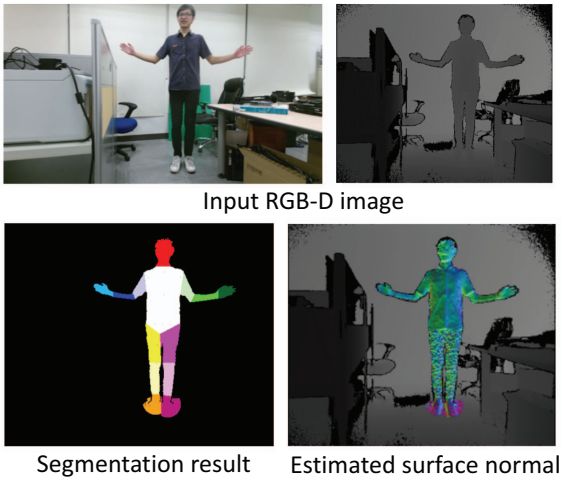

Figure 1. An illustration of inputs, intermediate results, and outputs of SegmentedFusion.

Recent work has used dense key-points [13, 14] or semi-dense key-points [9] to implement the tracking step and has achieved static scene reconstruction in real time. However, it is still extremely difficult to reconstruct the non-rigidly deforming scene because of the drastic increase in degree of freedom of the motion compared with the static scene. In [8, 15], warping strategies or energy functions are proposed to solve these issues. However, they are still restricted to limited topological change and slow motion.

Reconstruction techniques for a dynamically moving scene need to estimate a dense non-rigid warp field. By

2475-7888/18/$31.00 ©2018 IEEE
DOI 10.1109/3DV.2018.00031

matching points on the model with a live depth image, the transformation of a sparse set of control points is computed to obtain a dense warping field through interpolation [15]. However, this strategy is limited to well-controlled slow motion with no topological changes. Using skeleton information was proposed [22] to handle fast motions, but the problem of topological change was ignored. Motion-based segmentation to separate different objects was also proposed [17], where each object is tracked and reconstructed independently. We use this concept in our work to segment a human body into several body parts, to track each part's motion to build the warp field, and to fuse them into a 3D model independently. By deforming each body part correctly, our proposed method stitches all the parts and finally obtains a complete human body model. Our proposed method is capable of handling both fast motion and topological changes.

Our proposed method, SegmentedFusion, reconstructs non-rigid 3D models of the human body by using connected and deformed bounding boxes. We build connected bounding boxes, each of which bounds each body part, and we apply dense volumetric reconstruction to estimate the 3D model. The deformations of all the bounding boxes are initialized with the first input frame. Then, the deformations are tracked at run-time from the skeleton motion. Moreover, we use a refinement registration technique to reduce noises on the skeleton data and track the twist motion on the bone. Figure 1 is an illustration of our input, segmentation, and output.

The main contribution of this work is to propose connected and deforming bounding boxes representation for volumetric 3D reconstruction of a moving human body. By using skeleton information, we can estimate a dense volumetric model-to-frame warp field to fuse depth values of live frames into our canonical space. The dense warp field function is defined by blending the rigid transformations of adjacent bones. Through the interpolation, we find the 3D transformation of each voxel inside each bounding box. As a result, SegmentedFusion can handle fast human body motion and topological change as demonstrated in experiments.

2. Related Work

Here we focus on only non-rigid RGB-D SLAM methods that can be used for dynamic human body 3D reconstruction.

Several recently proposed methods focus on reconstructing a dynamic human body using a-priori available information such as template meshes. Using captured skeletal motion [3, 18, 21] or skeleton model representation, template-based 3D reconstruction methods have been proposed. Zollhöfer et al. [23] starts by scanning a template model as the prior. It then performs non-rigid registration to fit live data with the template and deforms the mesh in real time. However, this method requires a template beforehand, which makes it impractical for real-world applications. Furthermore, this method cannot handle topological change. The template-based methods make the model fixed, and aim at deforming the model to represent input data, instead of incrementally integrating the scene. They usually do not possess the function to refine the reconstructed model using input data.

Another approach to reconstruction is based on template-free techniques [8, 10, 15, 19]. DynamicFusion [15] proposes to estimate a dense volumetric warp field between the canonical frame and a live frame for dynamic scene reconstruction. By using a motion field, this system non-rigidly warps the canonical volume into the live frame, and updates the depth into the canonical model. However, DynamicFusion has the limitation that it cannot handle fast motion, splitting objects, and topological change because the model-update algorithm is based on mesh correspondences. VolumeDeform [10] improves DynamicFusion by using scale-invariant features. It uses a set of extracted sparse color features with a dense depth features for camera tracking and drastically reduces drifts that the standard model-to-depth alignment faces. Although VolumeDeform enables to handle faster motion than DynamicFusion, it still cannot properly recover all free motion. KillingFusion [19] was also proposed to handle fast inter-frame motion where the level set evolution instead of explicit correspondence search is used for non-rigid registration. Using signed distance fields for tracking and reconstruction, KillingFusion is capable of handling topological changes, but it may lose some high-frequency details.

In order to combine the advantages for the template-free and template-based approaches, BodyFusion [22] proposes a skeleton-embedded surface fusion. Using skeleton as a prior enables this method to deal with fast motion. BodyFusion estimates a deformation field based on attachments between graph nodes and bones, and updates the attachments with tracking. However, since the attachments are computed using the nearest-neighboring node, BodyFusion cannot deal with large topological change such as human hand motion. Human hand motion tends to change topology: connecting two hands (closed loop) to disconnecting them (open). Co-Fusion [17], on the other hand, handles this kind of problem by using motion or semantic cues to segment the scene into multiple objects and to reconstruct them separately. But it cannot provide good quality for the reconstruction.

In our work, we use skeleton as the prior instead of using a pre-scanned template model to handle fast motion. Also, similarly to Co-Fusion, our proposed method separates the human body into several body parts (volumes) while all the parts are connected to form the final whole human model stitched well. We use ICP as a refinement step and fuse up-to-date surface with all depth image data by saving the values of the truncated signed distance function (TSDF) [7]

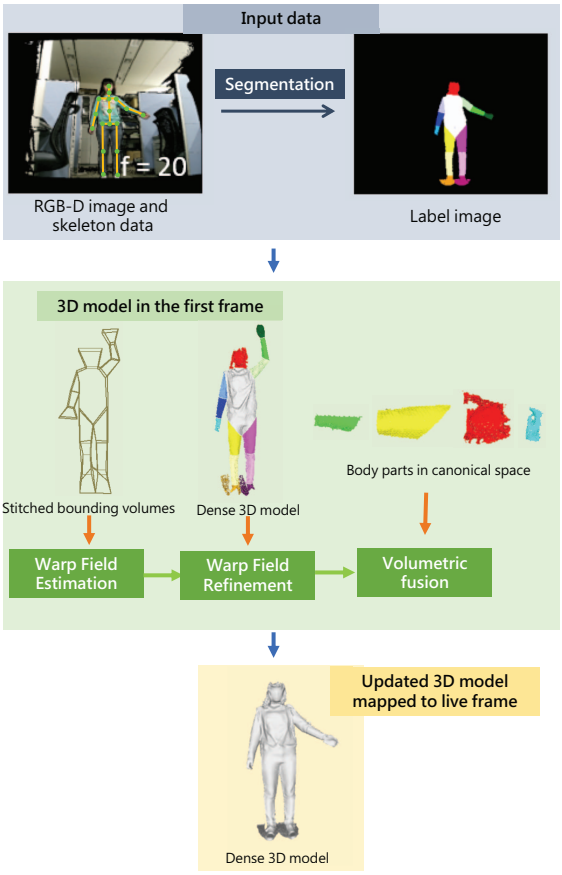

Figure 2. The pipeline of our proposed SegmentedFusion.

in the volumes.

3. Overview of Proposed Method

Our proposed method, called SegmentedFusion, aims to reconstruct the 3D model of the human body in dynamic scenes using a single depth camera and its associated skeleton data. Figure 2 illustrates the pipeline of our proposed SegmentedFusion.

Similarly to [15, 22], we represent the 3D model of the human body using the volumetric TSDF. We warp the canonical model (*i.e.*, the model in the reference pose) to non-rigidly align the current 3D model to the input data, and then fuse depth measurements into the canonical model using the running average strategy. Our main contributions and differences from existing work are two fold:

- We represent the 3D model of the human body with multiple small bounding boxes placed around body parts. More precisely, using skelton information, we segment the human body into (almost) rigid parts and define around each body part, a bounding box to which the TSDF values are attached. This representation enables us to explicitly handle topological changes, as it is defined by the skeleton.
- All the bounding boxes (one for each body part) are deformed on-line so that (1) no gap exists between them and (2) no overlap exists between them. More precisely, the relative orientation of adjacent body parts defines the deformation of all the bounding boxes. Thanks to this, when a person moves, the movement of the skeleton consistently deforms all the bounding boxes corresponding to the body parts. These deformations allow us to compute the warping field of the complete canonical TSDF space to non-rigidly align the 3D model to input measurements and also to fuse them into the canonical 3D model. This approach has the advantage that the warping field can be estimated by optimizing on the bones' motion only (we can decrease the number of unknowns for the estimation).

We use data captured with a Kinect V2 which outputs color images, depth images, and skeleton information. We also use [4] to improve the accuracy of the joint detection on the color image.

Notation and preliminaries. For a point $\mathbf{u} \in \mathbb{R}^2$ and its depth value $D_t(\mathbf{u}) \in \mathbb{R}$ at frame t, the back-projection is expressed by

$$\pi^{-1}(\mathbf{u}, D_t(\mathbf{u})) = D_t(\mathbf{u}) K^{-1} \tilde{\mathbf{u}} \quad (\in \mathbb{R}^3), \tag{1}$$

where π is the perspective projection from $\mathbb{R}^3$ to $\mathbb{R}^2$ (namely, $\pi(\mathbf{p}) = (p_x/p_z, p_y/p_z)^\top$ for $\mathbf{p} = (p_x, p_y, p_z)^\top \in \mathbb{R}^3$), $\tilde{\mathbf{u}} = (\mathbf{u}^\top, 1)^\top$, and K is the camera matrix representing the intrinsic camera parameters.

Our proposed method uses one TSDF volume for each body part, which corresponds to a bounding box. For each body part i, we define the canonical space $\mathbf{S}_i \in \mathbb{R}^3$, and we encode in each voxel (in the volume) the TSDF value and the confidence of measurements corresponding to the 3D model, as in [15, 22]. Namely, $\mathcal{V}_i : \hat{\mathbf{S}}_i \to \mathbb{R}^2$, where $\hat{\mathbf{S}}_i \in \mathbb{N}^3$ is the discretized canonical volume of $\mathbf{S}_i$ (a volume is digitized into voxels).

We use the skeleton information K_t at frame t, and we have the set of 3D joints $\{J_t(i)\}_{i \in [1:21]}$ with their tree structure (Figure 3). The 3D position of the i-th joint at frame t is denoted by $\mathbf{J}_t(i) \in \mathbb{R}^3$ (we use the boldface for the 3D position while the normal-face for the index).

We build the corresponding bones graph (right of Figure 3). In this graph, the notation $i \to j$ means the bone from the i-th to the j-th joint. One bone corresponds to one body part (*i.e.*, one bounding box), except for the bone $1 \to 2$ that is the root bone. This is because the root bone has no parent, thus no (non-rigid) deformation can be computed for bone $1 \to 2$. Note also that we dropped bones $21 \to 3$, $21 \to 5$, $21 \to 9$, $1 \to 13$ and $1 \to 17$ in the tree structure to simplify the implementation. Each bone is indexed with the index of the starting joint. Namely, $\mathbf{B}_t(i)$ is the bone vector from joint $J_t(i)$ to its child joint $J_t(c(i))$:

$$\mathbf{B}_t(i) = \mathbf{J}_t(c(i)) - \mathbf{J}_t(i). \tag{2}$$

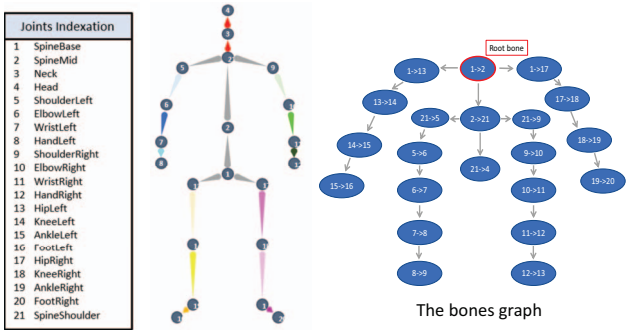

Figure 3. The Kinect's skeleton with joints (colors correspond to segmentation in body parts), and our proposed bones graph representation (right). We represent the skeleton in a tree structure, where each node is a bone that corresponds to a bounding box (except for the root node).

Note that $\mathbf{B}_t(p(i))$ denotes the parent bone of the i-th bone.

4. Detailed Description of Each Component

SegmentedFusion incrementally reconstructs the human body. It first warps the 3D model to fit the input depth-image, and then fuse depth measurements into the 3D model before processing the next input depth-image. SegmentedFusion is initialized with the first input depth image by segmenting the human body into multiple (almost) rigid parts and creating a canonical space for each body part. After estimating the warp-field, we compute the 3D transformation of each voxel, which is used to integrate the depth image into the set of TSDF volumes $\mathcal{V} = \{\mathcal{V}_i\}$.

4.1. Body part segmentation

Skeleton information crucially helps the 3D reconstruction of a moving person. This is because the human body is made of rigid bones and it is reasonable to assume that each body part corresponding to a bone is locally deformed in an almost rigid manner. The main idea in this work is to formulate non-rigid human-body reconstruction as a set of volumetric (almost) rigid 3D reconstruction of each body part. Then, the issue is how to deal with moving volumes representing body parts so that they have neither overlaps nor gaps between volumes even when a human freely moves. To tackle this problem, we propose a novel *deforming bounding box*, i.e., we allow each bounding box to be deformed depending on the skeleton motion.

We use the skeleton joints given by the Kinect V2 (middle of Figure 3) to segment the whole body into almost rigid body parts (Figure 4). More precisely, we use the 2D joint positions to find the contour of each body part. For each bone, we compute the 2D bone vector from two 2D joint positions. Two adjacent bone vectors give us a line that passes through the joint of the two bones and that bisects the angle between the two bones. We use the bisector line of every joint to separate the whole body into body parts.

Each body part attached to a bone $i \rightarrow j$ is segmented out by using the extreme joints $J(i)$ and $J(j)$ on the bone. We use two joints to define the limbs (arms, legs) while we use only one joint for the hands, the feet, and the head (we ignored joint 3 of the neck). The trunk part is what remains. Figure 4 illustrates the segmentation result.

Note that the bilateral filter [16, 20] is applied as the preprocessing on the raw depth image to reduce noise while preserving edges in the depth image. The segmentation algorithm, however, is not robust under all situations. The area near the feet is segmented incorrectly: the segmented feet is sometimes touching the floor. To reduce the influence caused by such an incorrect segment, we limit the size of each body part to be within a specific distance from the bone. That is why we see circles around the feet in Figure 4.

4.2. Stitched bounding boxes

The 3D model is represented with a set of volumes, each of which is uniformly discretized into voxels having TSDF values. A marching-cube algorithm [12] computes the 3D meshes from each body part given the current TSDF field. For a volume with size (X, Y, Z), the canonical space of a body part is simply the 3D space $\left([-\frac{X}{2} : \frac{X}{2}], [-\frac{Y}{2} : \frac{Y}{2}], [-\frac{Z}{2} : \frac{Z}{2}]\right) \subset \mathbb{R}^3$.

If we use a rigid bounding-box to represent each volume for a body part, then bounding-boxes may have overlaps or gaps along with human motion as shown in Figure 4. To prevent such a problem, we introduce the deforming bounding box, which tightly stitch all the bounding boxes together.

In their canonical space, all body parts are aligned and superimposed (as illustrated in Figure 4). Given the skeleton pose in the first input frame, we compute, for each body part, the rigid transformation $T_{\text{loc}\rightarrow\text{glo}}(i)$ that transforms the i-th body part from its canonical space to the coordinate system of the first frame.

$$T_{\text{loc}\rightarrow\text{glo}}(i) = \begin{bmatrix} R(i) & \mathbf{c}(i) \\ 0 & 1 \end{bmatrix}, \tag{3}$$

$$R(i) = [\mathbf{r}_1(i), \mathbf{r}_2(i), \mathbf{r}_3(i)] \in \text{SO}(3), \tag{4}$$

where $\mathbf{r}_1(i)$, $\mathbf{r}_2(i)$, $\mathbf{r}_3(i) \in \mathbb{R}^3$ are the normalized principal orthogonal vectors of the body part i, and $\mathbf{c}(i) \in \mathbb{R}^3$ is the 3D position of the center of the body part i. As illustrated in Figure 4, depending on the pose of the skeleton, the rigid transformation $T_{\text{loc}\rightarrow\text{glo}}(i)$ creates holes and overlaps, which will cause unpleasant visual artifacts after 3D reconstruction.

Each bone in our tree representation (right of Figure 3) has its unique parent (except for the root bone, which is not attached to any body part). Therefore, for the i-th body part we define an initial rotation matrix $R_{\text{init}}(i)$:

$$R_{\text{init}}(i) = \text{I}_3 + [\mathbf{s}]_\times + [\mathbf{s}]_\times^2 \frac{1}{1+\tau}, \tag{5}$$

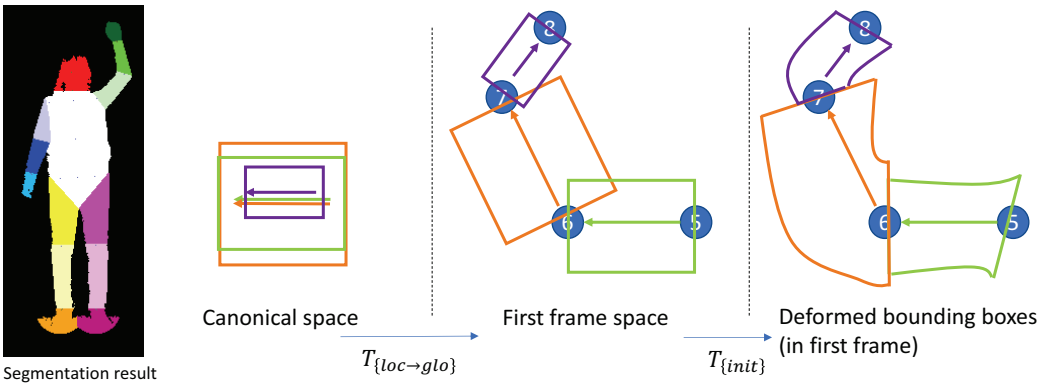

Figure 4. The results of the body part segmentation and illustration of the initial bounding box deformation. We took as an example three bones on the right arm: bone $5 \to 6$, bone $6 \to 7$ and bone $7 \to 8$. In their canonical space, all bone are aligned and their bounding boxes are superimposed. After applying the rigid transformations $T_{\text{loc}\to\text{glo}}$, all bounding boxes become aligned with the skeleton in the pose of the first input frame. But holes and overlaps appear between the rigid bounding boxes. Using the initial warp T_{init}, all bounding boxes are deformed so that they connect smoothly and tightly with the bounding box of their parent body part.

where I_3 is the 3×3 identity matrix, $\mathbf{s} = \mathbf{B}_1(i) \times \mathbf{B}_1(p(i))$, $\tau = \mathbf{B}_1(i) \cdot \mathbf{B}_1(p(i))$. $[\mathbf{s}]_\times$ is the skew-symmetric matrix defined by $\mathbf{s}$:

$$[\mathbf{s}]_\times \equiv \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix}. \tag{6}$$

For body part i, we also define the radial-based weight $w_i(\mathbf{p}) : \mathbb{R}^3 \to \mathbb{R}$ that defines the Gaussian distance between the 3D point $\mathbf{p}$ and the plane on the bounding box passing through the joint $J_t(c(i))$. Namely, with variance σ,

$$w_i(\mathbf{p}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{1}{2} \left(\frac{(\mathbf{p} - \mathbf{J}_1(c(i))) \cdot \mathbf{B}_1(i)}{\sigma} \right)^2 \right\}. \tag{7}$$

The stitching of the bounding box given the skeleton pose in the first frame is then obtained by applying the non-rigid transformation $\mathcal{W}_i^{\text{init}}(\mathbf{p})$ to all 3D points in the canonical space:

$$\mathcal{W}_i^{\text{init}}(\mathbf{p}) = \\ \mathcal{S}\left(w_i(\mathbf{p})\mathbf{q}_{\text{id}} + (1 - w_i(\mathbf{p}))\, \mathbf{q}_{\text{init}}(i)\right) T_{\text{loc}\to\text{glo}}(i), \tag{8}$$

where $\mathcal{S}$ converts a dual-quaternion to its corresponding transformation matrix in $\mathbb{SE}_3$, $\mathbf{q}_{\text{id}}$ is the dual-quaternion of the 4×4 identity transformation and $\mathbf{q}_{\text{init}}(i)$ is the unit dual-quaternion of the deformation transform $T_{\text{init}}(i)$, with

$$T_{\text{init}}(i) = \begin{bmatrix} R_{\text{init}}(i) & \mathbf{0} \\ 0 & 1 \end{bmatrix}. \tag{9}$$

Figure 5 illustrates the comparison between the results with and without using the stitching. The model without the stitching possesses unnatural deformation near the boundary. Thanks to this stitching, we can reconstruct a complete 3D model of the human body without any hole and with smooth connections between all the body parts. Also note that because of the initial stitching, the 3D models in their canonical space are actually deformed (but un-deformed after warping them into the initial frame coordinate system).

(a) (b)

Figure 5. The comparison of the mesh with and without initial warping. (a) Final lower arm models in canonical space. The left one use the initial deformation in warp field, and the right doesn't; (b) final whole models in the last frame without fusion step.

4.3. Warp field

For each voxel in a volume corresponding to a body part, we compute a warp function that non-rigidly transforms the voxel from its canonical space to its corresponding point in the current input depth image. The warping field $\mathcal{W}_i^t : \mathbf{S}_i \to \mathbb{SE}_3$ for a body part i at frame t is defined by combining the initial warp $\mathcal{W}_i^{\text{init}}$ with the current non-rigid motion tracking estimated using the input skeleton data. More precisely, at each new input frame at t, a body part i is first rigidly transformed to its pose in the first frame using $T_{\text{loc}\to\text{glo}}(i)$. Then the motions of each bone from the first frame to the current frame are computed using skeleton information. For the body part i the motion of the parent bone is composed with the initial warp to obtain a new transformation for the parent bone, with respect to the i-th body part. Then, the transformation of each voxel in the bounding box is obtained by applying the dual quaternion blending of the bone motion and the new transformation of

its parent bone. This initial deformation is further refined using ICP to obtain the final warp field $\mathcal{W}_i^t$. The concrete form is given in Equation 16.

4.3.1 Bone motion

We assume that the transformation of each body part is nearly rigid following the bone motion. Given the 3D skeleton information of two successive frames, the transformation of each body part from one frame to another can be computed. For the i-th body part, we use the bone vectors $\mathbf{B}_{t-1}(i)$ and $\mathbf{B}_t(i)$ in two successive frames to compute the bone's global transformation:

$$T_{t-1}^t(i) = \begin{bmatrix} R_{t-1}^t(i) & \mathbf{t}_{t-1}^t(i) \\ 0 & 1 \end{bmatrix}. \tag{10}$$

The rotation matrix $R_{t-1}^t(i)$ of the bone is computed with $\mathbf{B}_{t-1}(i) \times \mathbf{B}_t(i)$ and $\mathbf{B}_{t-1}(i) \cdot \mathbf{B}_t(i)$ (Equation 5). The translation of the bone, on the other hand, is given by

$$\mathbf{t}_t^{t-1}(i) = \mathbf{B}_t(i) - \mathbf{B}_{t-1}(i). \tag{11}$$

In this way, we obtain a 6 DOF transformation of one bone. Note that we have the fixed-scale assumption (*i.e.*, a bone is assumed to be able to neither stretch nor shrink).

4.3.2 Deformation around joints

Given the skeleton pose in two successive frames, we can estimate the transformation of each body part from one frame to another using the bones' motion. The whole human body model, however, is not well stitched because the vertices near the border between two body parts are influenced by the transformation associated with the two bones. In order to keep consistent stitching of the different body parts through any body motion, we need to interpolate the motions between adjacent bones. Therefore, we use the dual-quaternion blending [11] to represent the surface motion:

$$\mathcal{D}_i^t(\mathbf{p}) = w_i(\mathbf{p})\ \mathbf{q}_i^t + (1 - w_i(\mathbf{p}))\ \mathbf{q}_{p(i)}^t, \tag{12}$$

where $\mathbf{q}_i^t \in \mathbb{R}^8$ is the unit dual-quaternion of the i-th bone's motion $T_{t-1}^t(i)...T_2^3(i)T_1^2(i)$, and $w_i(\mathbf{p})$ is the radial-based weight of point $\mathbf{p}$. Note that $p(i)$ is the parent of the i-th bone.

4.3.3 Refinement

The bones' motions alone are insufficient to track the twist motion. Our goal here is to refine the bones' motions to track even such motions. Our refinement procedure takes two point clouds as its input: one is the segmented input 3D data and the other is our reconstructed surface composed of multiple body parts. We then define an energy function to be minimized for each body part for the refinement. We deform our surface and use the Iterative-Closest-Point (ICP [2]) for this minimization. This is because ICP corrects wrong estimates of translation caused by the noise in the skeleton data. We optimize the deformation parameters of all body parts turn by turn and run several iterations of the full optimization. At each iteration, each body part and its corresponding segment in the input 3D cloud of point are centered around their own gravity-center. After the optimization, for each body part we obtain our complete bone motion by combining the obtained rotation with the translation that is defined as the displacement between the gravity-centers of the two (segmented) point clouds.

The energy function is defined by the distance between the two point clouds and a regularization term to penalize non-smooth motion in the spatio-temporal domain: with a weight λ,

$$E = E_{\text{data}} + \lambda E_{\text{reg}}. \tag{13}$$

Our data term is the cost of the point-to-plane registration error:

$$E_{\text{data}} = \sum_k \left\| \hat{\mathbf{n}}_k^\top \left(\mathbf{u}_k - \hat{\mathbf{v}}_k \right) \right\|, \tag{14}$$

where $\hat{\mathbf{v}}_k = \mathcal{W}_i^t(\mathbf{v}_k)\mathbf{v}_k$ is the prediction of point $\mathbf{v}_k$ in a body part i from the canonical space to the live frame at t, $\hat{\mathbf{n}}_k$ is its corresponding normal vector, and $\mathbf{u}_k$ is the closest 3D point to $\hat{\mathbf{v}}_k$. Given two point clouds, we use the k-nearest neighbors algorithm (k-NN) [6] to find the correspondences and to estimate the distance.

Since the body parts are connected, the neighboring body parts should have similar motion. Also, the difference between the estimated rotation and initial one should be small to avoid matching the mesh to another body part. Accordingly, the regularization term is defined as follows:

$$E_{\text{reg}} = \sum_j \left\| \hat{\mathbf{q}}_j - \mathbf{q}_{p(j)} \right\|_2 + \left\| \hat{\mathbf{q}}_j - \mathbf{q}_j \right\|_2, \tag{15}$$

where $\mathbf{q}_j$ denotes the dual-quaternion of the initial rotation of the bone j, and $\hat{\mathbf{q}}_j$ is its refined one. The translation part of $\hat{\mathbf{q}}_j$ is replaced by the displacement vector between the gravity-centers of the two (segmented) input point clouds that correspond to the j-th body part.

Finally, we obtain the dense warp field and compute the deformation of all points from the first frame to another with tracking body motion an camera motion through transformation parameters for each body part. The dense warp field for the body part i at frame t is given by

$$\mathcal{W}_i^t(\mathbf{p}) = \\ \mathcal{S}\left(w_i(\mathbf{p})\hat{\mathbf{q}}_i^t + (1 - w_i(\mathbf{p}))\, \hat{\mathbf{q}}_{p(i)}^t \circ \mathbf{q}_{\text{init}}(i) \right) T_{\text{loc}\to\text{glo}}(i), \tag{16}$$

where $\mathbf{q}_{\text{init}(i)}$ is the unit dual-quaternion of the transformation $T_{\text{init}}(i)$, $\hat{\mathbf{q}}_i^t$ is the dual-quaternion of the refined transformation at frame t, and $\circ$ denotes the multiplication of two dual-quaternions.

4.4. Fusion

Using the depth image with the skeleton data, we define a canonical bounding-box with the per-frame volumetric warp field for each body part. We non-rigidly integrate live depth data into the volumes in the canonical coordinate. The voxel ratio parameter decides the resolution of the built 3D model, and trades the computational efficiency against the mesh quality. For a body part i, we sample a voxel $\mathbf{x} \in \mathsf{S}_\mathsf{i}$ and store in the voxel, its TSDF value $v(\mathbf{x})$ and weight value $\omega(\mathbf{x})$:

$$\mathcal{V}_i(\mathbf{x}) = [v(\mathbf{x}), \omega(\mathbf{x})]. \tag{17}$$

We used the standard TSDF and the running average technique to fuse depth measurements into our canonical volumetric model. The marching cube algorithm [12] is used to identify the zero-level set surface in the triangulated representation as the output from the TSDF volumetric representation. We remark that our method uses a volume for each body part locally to efficiently use the memory resource. In contrast, the existing methods set a large volume to cover the whole human body or scene, and a lot of voxels in the volume are useless in practice because they are empty. We can expect that our method consumes significantly less amount of memory.

5. Experimental Results

We demonstrate the effectiveness of our proposed method on several dynamic scene scenarios. In our experiments, we used the parameter values as follows: the variance in the radial-based weight $w_i = 0.02$ for all i, the weight in the energy function $\lambda = 0.5$, the number of iterations in minimization is 15, and the voxel size is 0.3mm. Note that all the 3D model results below are snapshotted from MeshLab system [5].

We captured more than 10 sequences with different human motions using a Kinect V2. They include a variety of challenging motion such as fast inter-frame motion or motion with topological change. On average, one sequence persists around 2 seconds, producing 34 frames. In Figure 8, for example, the person raises her hand from bottom to the upper position in about 1 second, which is considered a fast motion (this cannot be handled by the state-of-the-art methods [17] and [10]). We remark that we used [4] to obtain the skeleton data by aligning the color images to the depth data captured with Kinect SDK. This is because the skeleton tracking by the Kinect is easily failed when fast inter-frame motion or occluded regions arises.

Figure 6 shows some examples of reconstruction results obtained with our proposed method. We see that our method successfully reconstructed the human body in motion. Although we observe some large deformation of the body through all sequences, we observe neither any holes between different body parts, nor significant visual artifacts. This is because our estimated warp field efficiently stitched

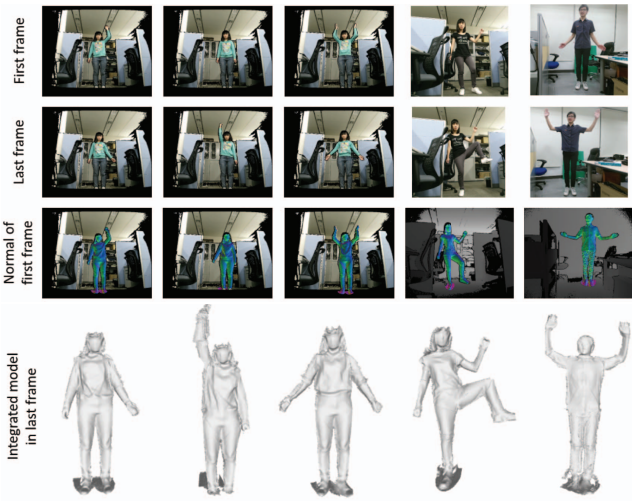

Figure 6. The reconstruction results obtained with SegmentedFusion, demonstrating that our proposed method can handle a variety of human motion.

Figure 7. Results obtained with our method in different motion scenarios:a scenario with slow body and camera motion (1st row), a scenario with large frame-to-frame motion (2nd row), and a scenario with motion in topological change (3rd row).

all the body parts together in any body pose. Moreover, our non-rigid motion tracking using initialization with skeleton motion followed by refinement with ICP allowed precise motion estimation. This enabled us to accurately fuse depth measurements into the canonical models and reconstruct detailed 3D models.

Figure 7 gives us a close look at reconstruction results for specific motion scenarios, showing the effectiveness of our proposed method. The first row shows the results in a simpler case of slow body and camera motion. We see that our method correctly tracks the deformation of the body and succeeds in reconstructing a detailed and smooth 3D model. The second row shows the case of fast body motion. Thanks to segmenting the human body into multiple parts

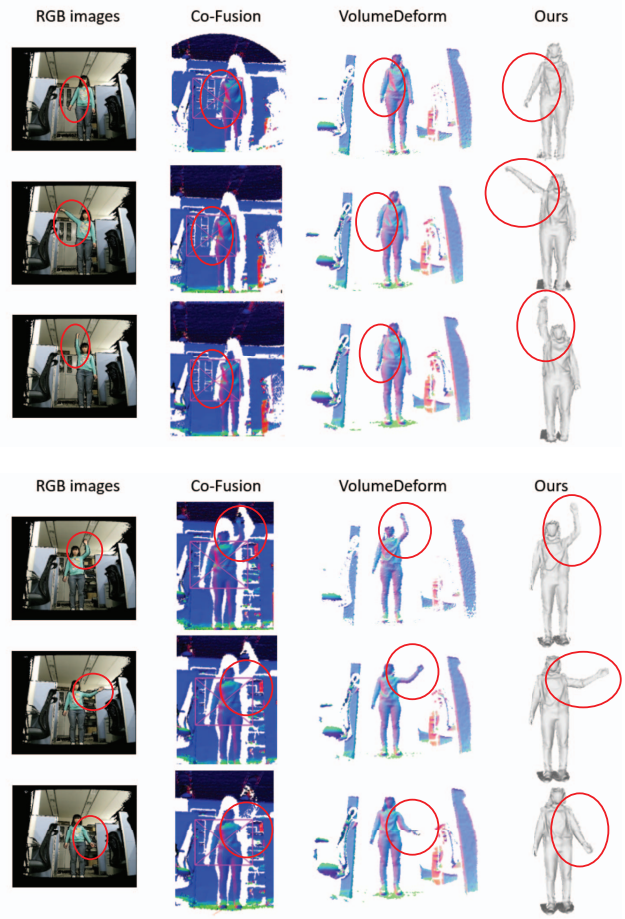

Figure 8. Comparison of the results obtained with our method and with Co-fusion [17], VolumeDeform [10] for two cases with fast motion. Co-Fusion fails in tracking the fast human motion and builds an uncompleted 3D model, while VolumeDeform fails in building a 3D model when motion with topological change appears. In contrast, our method successfully reconstructs a 3D model in any motion.

attached to the skeleton bones, the reconstructed 3D model efficiently adapts to fast body motion. The third row, on the other hand, shows that our method properly handles topological changes in motion. This is because the topology of the human body is identical with that of the skeleton, and, thus, our 3D model can easily adapt to topological changes by following the motion of the skeleton. The videos of these reconstruction results are available at [1] for better visualization.

We compared our method with Co-Fusion (the most closest work) [17] and VolumeDeform [10], which is shown in Figure 8. We used the publicly available code for results by Co-Fusion while the results by VolumeDeform were obtained by the author's experiment on our data (the authors of VolumeDeform kindly run their code on our data). Note that since the dataset used in [10] is not publicly available, we could not perform quantitative comparisons on the results in

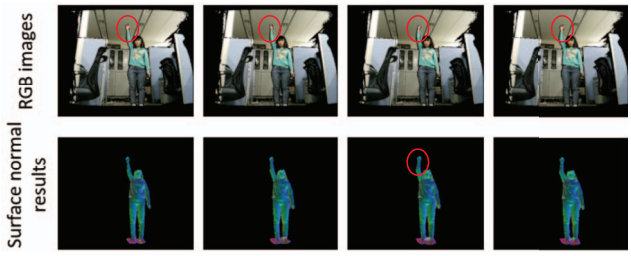

Figure 9. A failure case example: some twist motions could not be reconstructed by our method.

[10]. Overall, our method outperforms the other methods. If we closely look at Figure 8, we see that though Co-Fusion successfully segments different motions, it failed in reconstructing the whole 3D model. We also see that VolumeDeform fails in reconstruction when motion brings topological change (indeed, we observe topological change between the right arm and the trunk in (a)). Our proposed method, in contrast, always tracks the deformation of the human body and successfully reconstructs a complete dense 3D model of the body even when topological change appears. We note that Co-Fusion needs several static images to build the background model while our method does not.

5.1. Limitations

Although our proposed method can handle fast motions and topological changes, we observed through our experiments the following limitations. (1) We cannot reconstruct twist motions of the arms (Figure 9). This is because (a) skeleton information is not accurate enough and (b) sufficient salient geometric features are not available for the ICP algorithm. Using visual features is a possible way to overcome this problem, and we leave this task for future work. (2) Our proposed method fails in the case where large occlusions exist such as when crouching. This is because the segmentation of the body easily fails due to occlusions. We need a more robust method for human body segmentation. (3) In the upper part of Figure 8 we observe some gaps appearing in the armpit. Even though we deform the bounding boxes to minimize these gaps, the weights that blend the transformation of parent and child bones produce some collapse of the 3D model. Introducing weights for better deformation blending is required.

6. Conclusion

This paper presented SegmentedFusion, a system that reconstructs non-rigid human model by using a single depth camera with skeleton data. Our system segments the human body into multiple nearly rigid parts and builds the canonical bounding box for each part. Our system uses only bone motions to estimate a volumetric 6D motion field of each body part that warps the integrated model into the live frame. Our experiments show that SegmentedFusion is able to effectively handle fast motions and topological changes.

References

[1] http://limu.ait.kyushu-u.ac.jp/e/member/member0042.html. 8

[2] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992. 6

[3] K. Buys, C. Cagniart, A. Baksheev, T. De Laet, J. De Schutter, and C. Pantofaru. An adaptable system for rgb-d based human body detection and pose estimation. *Journal of visual communication and image representation*, 25(1):39–52, 2014. 2

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, volume 1, page 7, 2017. 3, 7

[5] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In V. Scarano, R. D. Chiara, and U. Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008. 7

[6] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967. 6

[7] 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*, pages 303–312. ACM, 1996. 2

[8] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 35(4):114, 2016. 1, 2

[9] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014. 1

[10] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision*, pages 362–379. Springer, 2016. 2, 7, 8

[11] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan. Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics (TOG)*, 27(4):105, 2008. 6

[12] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987. 4, 7

[13] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1

[14] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[15] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015. 1, 2, 3

[16] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 1, 4

[17] M. Rünz and L. Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4471–4478. IEEE, 2017. 2, 7, 8

[18] L. Sigal, M. Isard, H. Haussecker, and M. J. Black. Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. *International journal of computer vision*, 98(1):15–48, 2012. 2

[19] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 7, 2017. 2

[20] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998. 4

[21] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Transactions on Graphics (TOG)*, 31(6):188, 2012. 2

[22] T. Yu, K. Guo, F. Xu, Y. Dong, Z. Su, J. Zhao, J. Li, Q. Dai, and Y. Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. 2, 3

[23] 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 Transactions on Graphics (TOG)*, 33(4):156, 2014. 2