3D Scanning Deformable Objects with a Single RGBD Sensor

Mingsong Dou*[1], Jonathan Taylor[2], Henry Fuchs[1], Andrew Fitzgibbon[2] and Shahram Izadi[2]

[1]Department of Computer Science, UNC-Chapel Hill
[2]Microsoft Research

Abstract

We present a 3D scanning system for deformable objects that uses only a single Kinect sensor. Our work allows considerable amount of nonrigid deformations during scanning, and achieves high quality results without heavily constraining user or camera motion. We do not rely on any prior shape knowledge, enabling general object scanning with freeform deformations. To deal with the drift problem when nonrigidly aligning the input sequence, we automatically detect loop closures, distribute the alignment error over the loop, and finally use a bundle adjustment algorithm to optimize for the latent 3D shape and nonrigid deformation parameters simultaneously. We demonstrate high quality scanning results in some challenging sequences, comparing with state of art nonrigid techniques, as well as ground truth data.

1. Introduction

With the availability of commodity depth cameras, 3D scanning has become mainstream, with applications for 3D printing, CAD, measurement and gaming. However, many existing 3D scanning systems (e.g. [14, 9]) use rigid alignment algorithms and thus require the object or scene being scanned to remain static. In many scenarios such as reconstructing humans, particularly children, and animals, or in-hand scanning of soft and deformable objects, for instance toys, nonrigid movement is *inevitable*.

Recent work on nonrigid scanning are constrained by one or more of the following: 1) they rely on specific user motion (e.g. [24, 11, 18]); 2) require multiple cameras (e.g. [5]); 3) capture a static pre-scan as template prior (e.g. [29]); or align partial static scans nonrigidly (e.g. [11]).

To address these issues we present a new 3D scanning system for arbitrary scenes, based on a single sensor, which allows for large deformations during acquisition. Further, our system avoids the need for any static capture, either as a template prior or for acquiring initial partial scans.

Figure 1. A mother holds an energetic baby while rotating in front of a Kinect camera. Our system registers scans with large deformations into a unified surface model.

Our work uses a Kinect sensor which gives a partial noisy scan of an object at each frame. Our goal is to combine all these scans into a complete high quality model (Fig. 1). Even for rigid alignment, the problem of *drift* occurs when aligning a sequence of partial scans consecutively, where the alignment error accumulates quickly and the scan does not close seamlessly. Drift is more serious in the case of nonrigid alignment (Fig. 2(c)). KinectFusion alleviates some level of drift by aligning the current frame with the fused model instead of the previous frame [14].

Many follow-up systems based on KinectFusion have specifically looked at scanning humans (e.g., for 3D printing or generating avatars) where the user rotates in front of the Kinect while maintaining a roughly rigid pose, e.g., [24, 11, 18, 27, 7]. This highlights the fundamental issue when scanning living things – they ultimately move.

To make this problem more tractable, some systems make strong assumptions about the nonrigid object being a human, using either parametric models [24, 7] or limiting the user to certain poses such as a 'T' shape [3]. We wish to avoid such scene assumptions. Li *et al.* [11] adapt a more general nonrigid registration framework which can support a wider range of poses, clothing or even multiple users. This system demonstrates compelling results but relies on a very specific type of user interaction: the user moves in roughly 45 degree increments, in front of Kinect, and at each step

*Most work is conducted at Microsoft Research

remains static, whilst the motorized sensor scans up and down. Each of these partial static scans are then nonrigidly registered and a global model reconstructed. Here the user is assumed to explicitly perform a loop closure at the end of sequence. For certain energetic subjects, such as children or animals, who do not follow instructions well, such a usage scenario may be constraining.

Zeng *et al.* [27] show that when using nonrigid alignment to an embedded deformation (ED) graph model [15] for *quasi-rigid* motion, drift is greatly alleviated, and loop-closure can be made *implicit*. However, for nonrigid motion, our experience (Fig. 6) shows that drift is still a serious problem even when scanning mildly deforming objects such as a turning head.

In this paper, we detect loop closures explicitly to handle severe drift without restricting user motions. However, dealing with such loop closures, is only one piece of the puzzle, as this only evenly distributes error over the loop instead of minimizing the alignment residual. Thus, our pipeline also performs a dense nonrigid bundle adjustment to simultaneously optimize the final shape and nonrigid parameters at each frame. We use loop closure to provide the initialization for the bundle adjustment step. Our experiments show that bundle adjustment gives improved data alignment and thus a high quality final model.

We will summarize previous work in the next section and describe our surface and deformation model in Sec. 2&3. From Sec. 4 through Sec. 6, we explain the preprocessing procedures for bundle adjustment, including partial scan extraction, coarse scan alignment, and loop closure detection. Then we illustrate our bundle adjustment algorithm in Sec. 7. Finally, we show results in Sec. 8.

1.1. Related Work

Dou *et al.* [5] designed a system to scan dynamic objects with eight-Kinect sensors, where drift is not a concern given that a relatively complete model is captured at each frame. Tong *et al.* [18] illustrated a full body scanning system with three Kinects. Their system uses a turntable to turn people around, but cannot handle large deformations. Other high-end multi-camera setups include [4, 20, 6, 21]. In our work we wish to move away from complex rigs, and support more lightweight and commodity consumer setups, using only a single off-the-shelf depth sensor.

More lightweight capture setups have been demonstrated, but either still require complex lighting, more than one camera, or cannot generate high quality results [8, 12, 10, 23, 19, 26, 25].

More severe deformations can be handled with template-based systems. For example, Zollhofer *et al.* [29] first acquire a template of the scene under near-rigid motion using Kinect fusion, and then adapt that template to non-rigid sequences. Even more specialized are systems based on human shape models [20, 24, 28]. The shape prior means they cannot scan general shapes, including even humans holding objects, or in unusual clothing. More general approaches either work on diverse (non-rigged) templates [8, 4, 12, 10], or use template-less spatio-temporal representations [13, 22, 17]. Instead our system discovers the *latent* surface model without the need for an initial rigid scan or statically captured template model. It also attempts to mitigate the drift inherent in non-template-based models.

2. Triangular Mesh Surface Model

Throughout this paper, we use a triangular mesh as our fundamental surface representation. We parameterize a triangle mesh by the set of 3D vertex locations $\mathcal{V} = \{\mathbf{v}_{m=1}^{M}\}$ and the set of triangle indices $\mathcal{T} \subset \{(i,j,k) : 1 \leq i,j,k \leq M\}$. We will also occasionally query the triangulation through the function $\mathcal{N}(m)$ which returns the indices of the vertices neighboring vertex m, or through the use of a variable $\tau \in \mathcal{T}$ representing a single triangle face.

We will often need to label a mesh using a subscript (*e.g.*, $\mathcal{V}_i$) in which case we will label the vertices with a corresponding superscript (*e.g.*, $\mathbf{v}_m^i$). Indeed, a point on the surface itself is parameterized using a surface coordinate $\mathbf{u} = (\tau, u, v)$ where $\tau \in \mathcal{T}$ is a triangle index and (u, v) is a barycentric coordinate in the unit triangle. The position of this coordinate can then be evaluated using a linear combination of the vertices in τ as

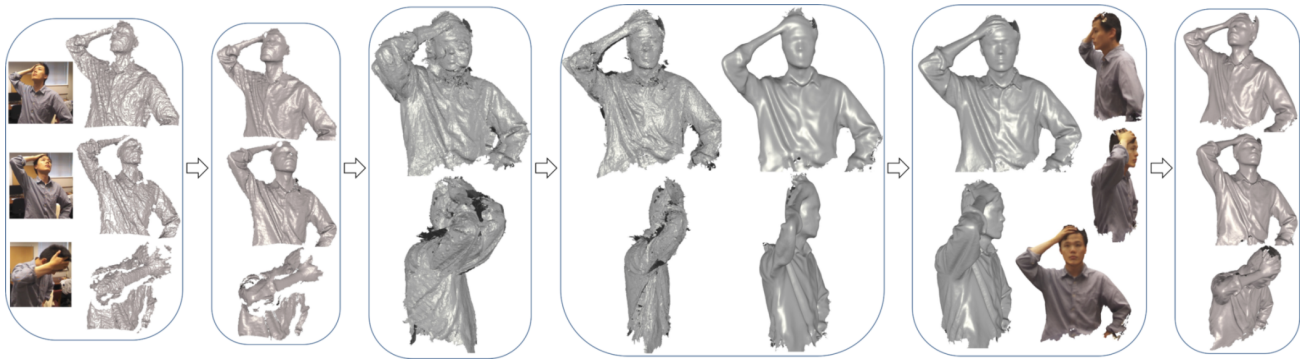
$$S(\mathbf{u}; \mathcal{V}) = u\mathbf{v}_{\tau_1} + v\mathbf{v}_{\tau_2} + (1-u-v)\mathbf{v}_{\tau_3} \qquad (1)$$

and its surface normal computed as (with $\|\!|x|\!\| := x/\|x\|$)

$$S^{\perp}(\mathbf{u}; \mathcal{V}) = \|\!|(\mathbf{v}_{\tau_2} - \mathbf{v}_{\tau_1}) \times (\mathbf{v}_{\tau_3} - \mathbf{v}_{\tau_1})|\!\| \qquad (2)$$

3. Embedded Deformation Model

In general, we will want to allow our meshes to deform, for example to allow our surface reconstruction to explain the data in a depth sequence. Our desire to keep our algorithm agnostic to object class led us to choose the embedded deformation (ED) model of [15] to parameterize the non-rigid deformations of a mesh $\mathcal{V}$. In this model, a set of K "ED nodes" are uniformly sampled throughout the mesh at a set of fixed locations $\{\mathbf{g}_k\}_{k=1}^{K} \subseteq \mathbb{R}^3$. Each vertex m is "skinned" to the deformation nodes by a set of fixed weights $\{w_{mk}\}_{k=1}^{K} \subseteq [0,1]$, where $w_{mk} = (\max(0, 1 - d(\mathbf{v}_m, \mathbf{g}_k)/d_{\max}))^2 / w_{\text{sum}}$ with $d(\mathbf{v}_m, \mathbf{g}_k)$ the geodesic distance between the two, $d_{\max}$ the distance of $\mathbf{v}_m$ to its c+1-th nearest ED node, and $\frac{1}{w_{\text{sum}}}$ the normalization weight. Note $\mathbf{v}_m$ is only influenced by its c nearest nodes ($c = 4$ in our experiments) since other nodes have weights 0. The weighted deformation of the vertices surrounding $\mathbf{g}_k$ is parameterized by a local affine transformation $A_k \in \mathbb{R}^{3\times3}$ and translation $\mathbf{t}_k \in \mathbb{R}^3$. In addition,



(a) input color and depth (b) partial scans (c) coarse-aligned scans (d) LC-aligned scans (e) LC-fused surface (f) BA-optimized surface (g) deformed surface

Figure 2. Scanning pipeline. The input sequence has around 400 frames which are fused into 40 partial scans (Sec. 4). Partial scans are consecutively placed in the reference pose to achieve the coarse alignment (Sec. 5). Next, loop closures are detected and alignment is refined (Sec. 6); all the LC-aligned scans are fused volumetrically to get the LC-fused surface which serves as the initial for the following bundle adjustment stage (Sec. 7). As a by-product of the system, the reconstructed model can be deformed back to each frame.

we follow [27, 11] in augmenting the deformation using a global rotation $R \in SO(3)$ and translation $T \in \mathbb{R}^3$. The precise location of vertex $\mathbf{v}_m$ deformed using the parameter set $G = \{R, T\} \cup \{A_k, \mathbf{t}_k\}_{k=1}^K$ is

$$ED(\mathbf{v}_m; G) = R \sum_{k=1}^{K} w_{mk} \left[A_k(\mathbf{v}_m - \mathbf{g}_k) + \mathbf{g}_k + \mathbf{t}_k\right] + T \tag{3}$$

and its associated surface normal is:

$$ED^{\perp}(\mathbf{n}_m; G) = R \left\| \sum_{k=1}^{K} w_{mk} A_k^{-T} \mathbf{n}_m \right\|. \tag{4}$$

In addition, we allow the former functional to be applied to an entire mesh at a time to produce a deformed mesh $ED(\mathcal{V}; G) := \{ED(\mathbf{v}_m; G)\}_{m=1}^M$.

In general, we will want to find parameters that either exactly or approximately satisfy some constraints (*e.g.* $ED(\mathbf{v}_m; G) \approx \mathbf{p}_k \in \mathbb{R}^3$), and thus encode these constraints softly in an energy function $E(G)$ (*e.g.* $E(G) = \|\mathbf{p}_k - ED(\mathbf{v}_m; G)\|^2$). In order to prevent this model from using its tremendous amount of flexibility to deform in unreasonable ways, we follow the standard practice of regularizing the deformation by augmenting $E(G)$ with

$$E_{\text{rot}}(G) = \sum_{k=1}^{K} \|A_k^T A_k - \mathbf{I}\|_F + \sum_{k=1}^{K} (\det(A_k) - 1)^2, \tag{5}$$

that encourages local affine transformations to be rigid (reflection is eliminated by enforcing a positive determinant), and

$$E_{\text{smooth}}(G) = \sum_{k=1}^{K} \sum_{j \sim k} \|A_j(\mathbf{g}_k - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j - (\mathbf{g}_k + \mathbf{t}_k)\|^2, \tag{6}$$

that encourages neighboring affine transformations to be similar. For clarity, we use $E_{\text{reg}}(G) = \alpha E_{\text{rot}}(G) + E_{\text{smooth}}(G)$ in later equations, where $\alpha = 10$ in our experiments. In addition, rigidity is encouraged by penalizing the deformations at ED nodes,

$$E_{\text{rigid}}(G) = \sum_k \rho\left(\|A_k - \mathbf{I}\|_F\right) + \sum_k \rho\left(\|\mathbf{t}_k\|^2\right), \tag{7}$$

where $\rho(\cdot)$ is a robustness kernel function. We minimize this energy using standard nonlinear least squares optimization [15, 5, 10].

4. Extracting Partial Scans

The first phase of our algorithm begins by preprocessing an RGBD sequence into a set of high quality, but only partial, scans $\{\mathcal{V}_i\}_{i=1}^N$ of the object of interest. Each of these segments is reconstructed from a small contiguous set of F frames using the method of [5] to fuse the depth data into a triangular mesh. These short segments can be reliably reconstructed using standard methods, in contrast to larger sequences where camera and reconstruction drift generally leaves gross errors at loop closure boundaries. In addition, these segments compress the information contained in the full sequence, drastically reducing the computational complexity of fitting our surface model to the entire sequence as described in following sections.

To reconstruct the partial scan for segment i, we begin by iteratively fusing data from each frame $f \in \{1, ..., F\}$ into the reference frame which is set as the first frame. This is trivially accomplished for frame 1, so for frame $f \in \{2, ..., F\}$ we extract from the current volumetric representation of the reference frame, the reference mesh $\mathcal{V}_i^1$ and align it to frame f using an ED deformation with parameters G_i^f. Note that the parameters G_i^{f-1} can be used to initialize this optimization. We then observe the deformed mesh $ED(\mathcal{V}_i^1; G_i^f)$, and find a set of nearby points on $\mathcal{V}_i^f$ to establish a set of correspondences between $\mathcal{V}_i^f$ and $\mathcal{V}_i^1$. These correspondences can then be used to estimate a pa-

rameter set $\hat{G}_i^f$ that aligns $\mathcal{V}_i^f$ back to $\mathcal{V}_i^1$ in the reference frame [15] and that can be used to volumetrically fuse the data from frame f into the reference frame (where $\mathcal{V}_i^1$ lives). After completing this operation for all frames, a single surface $\mathcal{V}_i$ is extracted from the volumetric representation using marching cubes [5].

After this initial fusing, we have obtained a set of partially reconstructed segments $\{\mathcal{V}_i\}_{i=1}^N$, each of which is a partial scan of the object of interest at a different time and in a different pose. Examples of partial scans are shown in Figure 2(b). Ultimately, we want all segments $\{\mathcal{V}_i\}_{i=1}^N$ to be explained by a single complete mesh $\mathcal{V}$ (we call it the **latent mesh**) and a set of ED graphs $\{G_i\}_{i=1}^N$ that deforms $\{\mathcal{V}_i\}_{i=1}^N$ to $\mathcal{V}$. But it is not immediately clear where to get such a mesh, and how to get a good initial estimate of the deformation parameters required to achieve this. Instead, we proceed by deforming all segments into the reference pose, fusing the results together into a complete mesh, and using the deformations to provide a good initial guess for the parameters that minimize an appropriate energy.

5. Coarse Scan Alignment

In this section, we describe how we find deformation parameters G_i for each segment $\mathcal{V}_i$ so that a set of roughly aligned meshes $\{ED(\mathcal{V}_i; G_i)\}_{i=1}^N$ can be obtained in the reference pose (*i.e.* pose of $\mathcal{V}_1$). We first align each segment $\mathcal{V}_i$ to its immediate neighbor $\mathcal{V}_{i+1}$ yielding a parameter set $G_{i\to i+1}$ by using the technique in [5]. This is effortless as adjacent scans have similar poses and the $G_{i\to i+1}$ can be initialized using the parameters already estimated by [5] when aligning the first frame to the last frame of segment i.

To obtain an alignment of segment $\mathcal{V}_{i+1}$ back to the reference frame, it is helpful to assume that we have already obtained such an alignment for segment $\mathcal{V}_i$, which is trivial for $i = 1$. Then for each vertex $\mathbf{v}_m^i$ of mesh $\mathcal{V}_i$, we find the nearest surface point $\mathbf{v}_{\mu(m)}^{i+1}$ on $\mathcal{V}_{i+1}$ (closer than 1cm) to its deformed position $ED(\mathbf{v}_m^i, G_{i\to i+1})$. Similarly, the alignment parameter set G_i tells us that $\mathbf{v}_m^i$ should be located at $\tilde{\mathbf{v}}_m^i = ED(\mathbf{v}_m; G_i)$ in the reference frame. This process establishes a set of correspondences $\{\langle \mathbf{v}_{\mu(m)}^{i+1}, \tilde{\mathbf{v}}_m^i \rangle\}_{m=1}^M$ which provide constraints that can be used to estimate G_{i+1} using the standard ED alignment algorithm [15].

6. Error Redistribution

Naturally, the error in the propagation step accumulates, making the deformation parameter sets more and more unreliable as i increases. On the other hand, we assume that our sequence includes a loop closure and thus there should be some later segments that could match reasonably well to earlier segments. We would thus like to identify such pairs and establish rough constraints between them, in the form of correspondences, so that the deformations can be refined. To this end, we consider matching the aligned scan $ED(\mathcal{V}_i; G_i)$ against the aligned scans $\{ED(\mathcal{V}_j; G_j)\}_{j=1}^{i-K}$, where $K \geq 1$ restricts to frames with enough movement. To measure the overlap of a mesh $\mathcal{V}_j$ and a mesh $\mathcal{V}_i$, we define the overlap ratio

$$d(\mathcal{V}_i, \mathcal{V}_j) = \frac{1}{M_i} \sum_{m=1}^{M_i} I[\min_{m'} \|\mathbf{v}_m^i - \mathbf{v}_{m'}^j\| < \delta] \qquad (8)$$

as the proportion of vertices in $\mathcal{V}_i$ that have a neighboring vertex in $\mathcal{V}_j$ within δ (we use δ =4cm). We thus calculate $d_j^i = d(ED(\mathcal{V}_i; G_i), ED(\mathcal{V}_j; G_j))$ and consider as possible candidates, the set of scan indices $\mathcal{J}_i = \{j : d_j^i \geq r_1, |i-j| > K, d_j^i > d_{j-1}^i, d_j^i > d_{j+1}^i\}$, the indices whose aligned scan is at least K indices away with a 'peak' overlap ratio of at least r_1. For any scan index $j \in \mathcal{J}_i$, we then consider doing a more expensive, but more accurate, direct alignment of $\mathcal{V}_j$ to $\mathcal{V}_i$ using a set of ED parameters $G_{j\to i}$ [5]. If $d(\mathcal{V}_i, ED(\mathcal{V}_j, G_{ji})) \geq r_2$ we then find a set of correspondences $\mathcal{C}_{ij} \subseteq \{1, ..., M_i\} \times \{1, ..., M_j\}$ for which for any $(m, m') \in \mathcal{C}_{ij}$, we have that $\|\mathbf{v}_m^i - ED(\mathbf{v}_{m'}^j, G_{j\to i})\|$ is less than 1 cm. We set $\mathcal{C}_{ij} = \emptyset$ for any other pairs of frames that did not pass this test. In our experiment we let $r_1 = 30\%, r_2 = 50\%$.

With these loop closing correspondences extracted, we use Li *et al.*'s algorithm [11] to re-estimate ED graph parameters $\mathcal{G} = \{G_i\}_{i=1}^N$, by minimizing the energy;

$$\min_{\mathcal{G}} \lambda_{\text{corr}} E_{\text{corr}}(\mathcal{G}) + \lambda_{\text{reg}} \sum_i E_{\text{reg}}(G_i) + \lambda_{\text{rigid}} \sum_i E_{\text{rigid}}(G_i), \qquad (9)$$

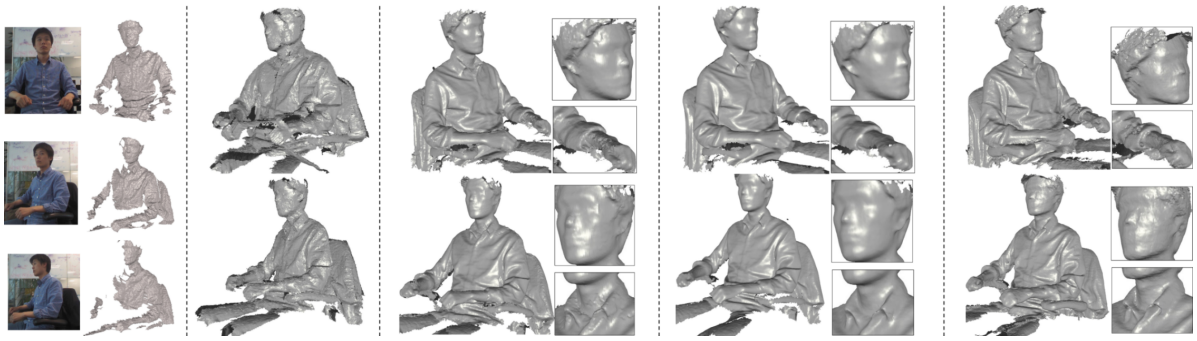
where

$$E_{\text{corr}}(\mathcal{G}) = \sum_{\substack{i=1 \\ j=1 \\ j\neq i}}^{N} \sum_{(m,m')\in\mathcal{C}_{ij}} \|ED(\mathbf{v}_m^i; G_i) - ED(\mathbf{v}_{m'}^j; G_j)\|^2 . \qquad (10)$$

After the set of deformation parameters $\mathcal{G}$ is estimated, we deform the scans accordingly and fuse them volumetrically to obtain a rough latent surface $\mathcal{V}$. Fig. 2(c,d)&3(b) show examples of scan alignment before and after loop closure.

7. Dense Nonrigid Bundle Adjustment

At this point, the above procedure has succeeded in giving us a rough surface representation of our object of interest, but the process has washed out the fine details that can be seen in the partial scans (see Fig. 2 and Fig. 8). This is largely a result of the commitment to a set of noisy correspondences used for error distribution. Eq. 9 does not aim to refine these correspondences, and thus misalignments are inevitable. As shown in Fig. 2 where large deformation exists, the misalignment is still visible where a loop closure has occurred, and the fused model looks flat and misses many details.



(a) input color & depth (b) Before/After LC (c) LC-fused surface (d) BA-optimized surface (e) KinectFusion

Figure 3. Scanning a person with slight deformation. Before loop closure (LC), scans are poorly aligned. After LC, the surface is topologically correct but noisy. Bundle Adjustment (BA) removes spurious noise without further smoothing details such as the shirt collar.

To improve both the data alignment and recover the fine details we employ a bundle adjustment (BA) type technique to refine $\mathcal{V}$ as to explain all the data summarized in the partial scans $\{\mathcal{V}_i\}_{i=1}^N$. We parameterize the deformation that each partial scan $\mathcal{V}_i$ has to undergo to be explained by the reference $\mathcal{V}$ using a set of ED deformation parameters G_i. We then cast an energy $E(\mathcal{V})$ over the latent mesh $\mathcal{V}$ as a combination of the following terms.

7.1. Deformation Terms

For each data point $\mathbf{v}_m^i$ in segment $\mathcal{V}_i$, we expect that some ED graph G_i deforms it towards the latent mesh $\mathcal{V}$ and $ED(\mathbf{v}_m^i; G_i)$ gets explained by $\mathcal{V}$. We thus add an energy term designed to encourage the distance of $ED(\mathbf{v}_m^i; G_i)$ to the latent surface to be close, and for the normal to match. This term is

$$E_{\text{data}}(\mathcal{V}) = \sum_{i=1}^{N} \min_{G_i} \sum_{m=1}^{M_i} \min_{\mathbf{u}} \lambda_{\text{data}} E_{\text{point}}(\mathbf{v}_m^i; G_i, \mathbf{u}, \mathcal{V}) + \lambda_{\text{normal}} E_{\text{normal}}(\mathbf{n}_m^i; G_i, \mathbf{u}, \mathcal{V}) + \lambda_{\text{reg}} E_{\text{reg}}(G_i) + \lambda_{\text{rigid}} E_{\text{rigid}}(G_i)$$

where

$$E_{\text{point}}(\mathbf{v}; G, \mathbf{u}, \mathcal{V}) = \|ED(\mathbf{v}; G) - S(\mathbf{u}; \mathcal{V})\|^2 \qquad (11)$$

and

$$E_{\text{normal}}(\mathbf{n}; G, \mathbf{u}, \mathcal{V}) = \|ED(\mathbf{n}; G) - S^{\perp}(\mathbf{u}; \mathcal{V})\|^2 \,. \qquad (12)$$

$S(\mathbf{u}; \mathcal{V})$ and $S^{\perp}(\mathbf{u}; \mathcal{V})$ are corresponding point and normal of $ED(\mathbf{v}; G)$[1] in the latent surface $\mathcal{V}$, which we have explained in Section 2.

As we continue to use the ED deformation model, the terms $E_{\text{reg}}(G)$ and $E_{\text{rigid}}(G)$ continue to provide regularization for ED graphs.

[1]Note that we do not set ED graph G_i on the latent mesh $\mathcal{V}$ to deform $\mathcal{V}$ towards partial scan $\mathcal{V}_i$ and minimize $\sum_{i=1}^{M_i} \min_{\mathbf{u}} \|\mathbf{v}_m^i - S(\mathbf{u}; ED(\mathcal{V}; G))\|^2$, because this gives many unnecessary ED nodes as $\mathcal{V}$ is complete and $\mathcal{V}_i$ is partial.

7.2. Surface Regularization Terms

In addition, we regularize the latent mesh using the Laplacian regularizer

$$E_{\text{lap}}(\mathcal{V}) = \sum_{m=1}^{M} \|\mathbf{v}_m - \frac{1}{|\mathcal{N}(m)|} \sum_{m' \in \mathcal{N}(m)} \mathbf{v}_{m'}\|^2 \,, \qquad (13)$$

where $\mathcal{N}(m)$ is the set of indices of vertices that neighbor $\mathbf{v}_m$. This term attracts a vertex to the centroid of its neighbors, penalizing unevenness of the surface, but has the potential to shrink the surface by dragging the set of boundary vertices inwards. We thus also add a energy term encouraging isometry as

$$E_{\text{iso}}(\mathcal{V}) = \sum_{m \in \mathcal{B}} \sum_{m' \in \mathcal{N}(i)} |\|\mathbf{v}_{m'} - \mathbf{v}_m\|^2 - L_{mm'}^2|^2 \qquad (14)$$

where $\mathcal{B} \subseteq \{1, ..., M\}$ is the set of indices of such boundary vertices, and $L_{mm'}$ is the length $\|\mathbf{v}_{m'} - \mathbf{v}_m\|$ in the initial mesh.

7.3. Solving

Combining all of the above energy terms, we obtain the full energy

$$E(\mathcal{V}) = E_{\text{data}}(\mathcal{V}) + \lambda_{\text{lap}} E_{\text{lap}}(\mathcal{V}) + \lambda_{\text{iso}} E_{\text{iso}}(\mathcal{V}) \qquad (15)$$

that we seek to minimize. To deal with the inner minimizations, we follow the lead of [16, 29] of defining a set of latent variables, passing them through the sums, and rewriting the energy in terms of a lifted energy defined over these additional latent variables. In our case, we have the ED deformation parameter sets $\mathcal{G} = \{G_i\}_{i=1}^N$ and the surface coordinates $\mathcal{U} = \{\mathbf{u}_1^m\}_{m=1}^{M_1} \cup ... \cup \{\mathbf{u}_N^m\}_{m=1}^{M_N}$, which allows us to obtain a lifted energy $E'(\mathcal{V}, \mathcal{G}, \mathcal{U})$ such that

$$E(\mathcal{V}) = \min_{\mathcal{G}, \mathcal{U}} E'(\mathcal{V}, \mathcal{G}, \mathcal{U}) \leq E'(\mathcal{V}, \mathcal{G}', \mathcal{U}') \qquad (16)$$

for any $\mathcal{G}'$ and $\mathcal{U}'$. We can thus minimize our desired energy by minimizing this lifted energy and to this end, we notice

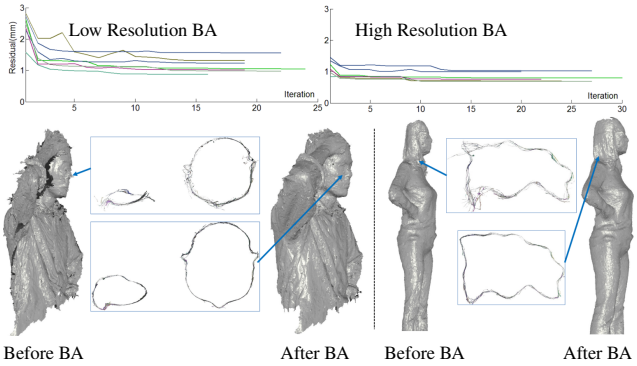


Figure 4. Top: Partial scan alignment residuals during bundle adjustment. Bottom: two examples of aligned scan before and after BA. The cross sections of scans are given in the middle.

that all terms are in a sum of squares form. We thus use the Levenberg–Marquardt algorithm implemented in Ceres [1] to minimize $E'(\mathcal{V},\mathcal{G},\mathcal{U})$. We initialize the latent mesh $\mathcal{V}$ using the coarse mesh recovered in the previous section, and $\mathcal{G}$ using the corresponding ED parameter sets and $\mathcal{U}$ by conducting a single closest point computation.

Note that even though surface normal $S^{\perp}(\mathbf{u};\cdot)$ is constant with respect to the barycentric coordinate $\mathbf{u}$ (an entire triangle on the latent surface shares the same normal vector), it does give constraints to the latent mesh and the ED graphs, which makes latent surface smooth and improves the alignment.

Note that some special care has to be taken to allow the Levenberg-Marquardt algorithm to interact with a surface coordinate variable $u \in \mathcal{U}$ [16, 2]. Such a variable has the atypical parameterization $\mathbf{u} = (\tau, u, v)$ where τ is discrete (a triangle ID), and (u, v) are real valued coordinates in the unit triangle. As typically the coordinate (u, v) will lie strictly within the unit triangle, τ remains constant locally and only the Jacobians with respect to (u, v) which are well defined are provided to the optimizer. When an update $(u, v) \leftarrow (u, v) + \delta(du, dv)$ is requested that would exit the unit triangle, the coordinate should first move the distance $\hat{\delta}$ to the edge of the triangle. The adjacent triangle τ' is then looked up, a new direction (du', dv') and step size $\delta' = \delta - \hat{\delta}$ computed, and finally the procedure is recursively called after updating $\tau \leftarrow \tau'$, $(du, dv) \leftarrow (du', dv')$ and $\delta \leftarrow \delta'$. Eventually the step size δ will be sufficiently small that an update does not need to leave a triangle.

8. Experiments

In the following experiments, we evaluate our method on a variety of RGBD sequences of various objects of interest. Each sequence is between 200 and 400 frames, and we fuse these volumetrically into 20 to 40 partial scans by fusing the data from each $F = 10$ frame subsegment. We set the size of the voxels in the fusion procedure to 2mm cubed

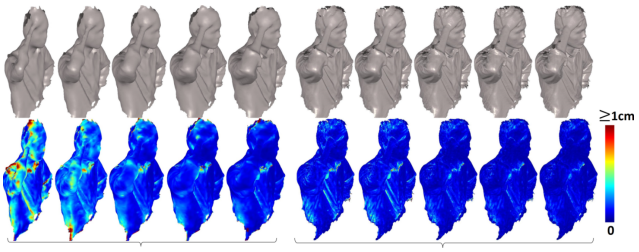


Figure 5. Bundle adjustment iterations. Top row: evolving surface model. Bottom: per-vertex residuals. Note the increase in detail of the right forearm and hand.

when scanning a close object and 3mm cubed for objects at a further distance. This results in partial scans with around 100,000 vertices. When conducting nonrigid alignment for both partial scan extraction and alignment, ED nodes are sampled as to remain roughly at 5cm (measured in geodesic distance) to their neighbors. This endows each ED graph with roughly 150 to 200 nodes depending on the dimension of the object of interest.

After detecting the loop closure constraints and performing error redistribution, the aligned partial scans are volumetrically fused to get an initial latent mesh for the final bundle adjustment stage. We perform bilaterial filtering on the volume data to ameliorate any misalignment. We also perform a simple remeshing to eliminate thin triangles on the initial latent mesh extracted with marching cubes, which makes the bundle adjustment numerically stable.

The bundle adjustment is the most expensive stage, given the huge amount of parameters to be optimized in Eq. 16: the roughly 5,000 graph nodes, 300,000 vertices of the latent mesh and three million surface coordinates. A limitation of this procedure is that the number of vertices on the latent mesh and its triangulation remain fixed throughout the bundle adjustment stage. Thus, if the initial mesh does not have the correct shape topology or has missing parts due to poor initial alignment, it is difficult for the bundle adjustment to recover the correct shape. To handle the above issues, we take a coarse-to-fine approach by running the bundle adjustment twice with different levels of detail.

In the first run, a low resolution latent mesh is used with an average distance between neighboring vertices of 1cm. The first run quickly converges and improves partial scan alignments $\mathcal{G}$ significantly, from which a better initial latent mesh can be built. In the second run, we use the full resolution mesh where the average distance between neighboring vertices is about 2mm. Initializing the parameters from the previous bundle adjustment, the vertices on the latent mesh do not need to move much along the tangent direction, so we constrain the vertex to only move as a displacement along the direction normal to the initial latent mesh, which reduces the number of parameters on the latent surface by nearly two thirds. That is, only one single displacement

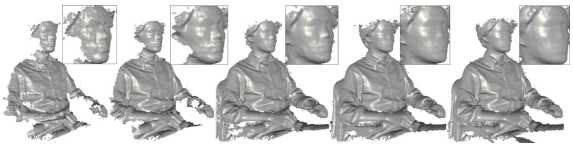


Figure 6. KinectFusion with nonrigid alignment. The accumulated surfaces after fusing 10, 30, 50, 70, 90 frames are shown. Note the nose gets blurred at the end.

parameter per vertex instead of three is required to parameterize full 3D position.

Fig. 5 illustrates the intermediate latent surfaces together with the alignment residual at each BA iteration; the alignment error is computed for each vertex on the latent surface as its average distance to the deformed partial scans. Fig. 4 plots the average alignment residuals during BA (including both accepted and rejected BA iterations) on various data sequences. The alignment error typically goes down from 3mm to less than 1mm. Examples of aligned scans before and after BA are also given in Fig. 4, where the cross sections of scans are shown to demonstrate the alignment quality and the bundle adjustment's ability to recover the true structure of the object.

8.1. Comparison with KinectFusion

Our system is designed for dynamically moving objects, but it still works in more restricted cases such as rigid scenes (*i.e.* scanning static objects). Fig. 7 shows the comparison in reconstruction quality between our method and KinectFusion on a static mannequin. To compare the two systems quantitatively, we first generate a 3D model of the mannequin which serves as the ground truth and then synthesize a sequence of depth maps and color images by moving a virtual camera around the 3D model. We run our algorithm and KinectFusion on the synthetic data. As shown in Fig. 7, both systems give appealing reconstructions which are authentic to ground truth. KinectFusion has an average reconstruction error of 0.94 mm v.s. 1.21 mm in our system. Our system has lower residuals on the side that is observed by the reference frame (1st row in Fig. 7, error map uses the same scale as Fig. 5) while it has higher residual on the other side (2nd row in Fig. 7) due to flexibility introduced by the nonrigid alignment. Naturally, we don't expect to outperform a method that exploits the rigidity of this scene, but we are satisfied that our system can get similar results without requiring such assumptions.

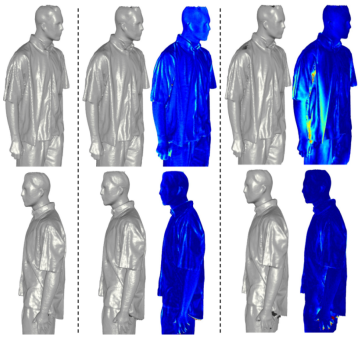


Figure 7. Rigid Scanning.

In contrast though, KinectFusion fails in dynamic cases. Fig. 3 shows the reconstruction results of KinectFusion on a sequence with slight head movement. Replacing ICP in KinectFusion with the nonrigid alignment algorithm [5] does not result in a reasonable reconstruction either. As shown in Figure 6, when non-rigidly fusing more than 30 frames, the drifting artifacts result in a blurred nose.

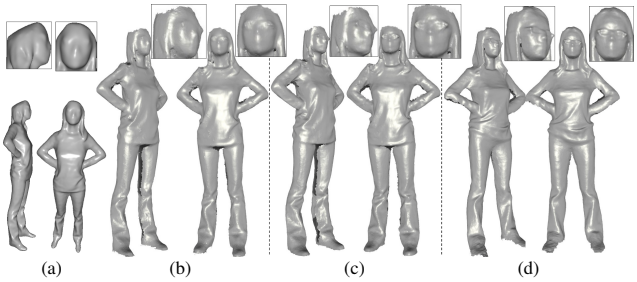


Figure 9. Comparison with 3D Self-Portraits. Scanning results of (a) shapifyme; (b) 3D self-portraits implemented by us; (c) BA-optimized 3D self-portraits; (d) our system.

8.2. Comparison with 3D Self-portraits

3D self-portraits [11] is among the first systems with the capability to scan a dynamic object with a single consumer sensor. We want to stress that our system handles continuously deforming objects while 3D self-portraits first reconstructs eight static scans and then non-rigidly fuses them later. The above difference prevents us from comparing the two system quantitatively, but we show side-by-side of the reconstructed models of the same person from the two systems in Fig. 9. The software Shapifyme which implements 3D self-portraits appears to heavily smooth the reconstructions, and our implementation of 3D self-portrait gives more detailed reconstructions. We then ran bundle adjustment algorithm of Sec. 7 on the eight scans, and we find that it improves the reconstruction further, showing another advantage of our approach. Compared with 3D self-portraits, our system allows continuous movement and recovers more facial details.

8.3. Synthetic sequence

We tested our system on the Saskia dataset [21] which contains dramatic deformations. The original sequence has a roughly complete model at each frame, and thus we synthesize one depth map and color image from each frame with a virtual camera rotating around the subject. Our reconstruction system results in a shape in a reference pose (*i.e.* the latent mesh $\mathcal{V}$) as shown on the left of Fig. 10. To measure alignment error, we then deform $\mathcal{V}$ to each frame and compute the distance from the frame data. To achieve this a backward ED graph $\tilde{G}_i$ from $\mathcal{V}$ to each partial scan $\mathcal{V}_i$ is first computed using correspondences. The deformations from partial scan $\mathcal{V}_i$ to the frames in segment i have already

Figure 8. Top: reconstruction after loop closure. Bottom: final reconstruction after bundle adjustment.

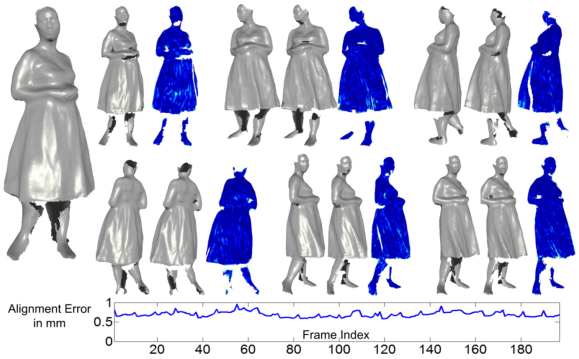


Figure 10. Alignment error in Saskia dataset. The first shape in each triple is the deformed reconstructed surface, the second is the ground truth, and the third shows the alignment error (same scale as Fig. 5). Per-frame alignment error is drawn at the bottom.

been computed as explained in Section 4, so we first deform $\mathcal{V}$ to each partial scan's pose and then to each frame's pose. The alignment error is then measured between the deformed reconstruction and the synthesized depth map. We draw the alignment error at each frame at the bottom of Fig. 10.

The Saskia sequence poses a particular challenge as the topology changes when the dress touches the legs. This introduces some artifacts on the legs in the reconstructed latent mesh $\mathcal{V}$ and also gives some problems in the deformed latent mesh in each frame's pose.

8.4. Scanned examples

Fig. 3 shows a sequence with small deformations. The loop closure technique described in Sec. 6 reconstructs a reasonable model, but some artifacts exist due to misalignment. Our bundle adjustment technique in Sec. 7, however, improves the reconstruction. Another example with considerable deformations is shown in Fig. 2, where the loop closure gives a problematic alignment of the partial scans and a poor reconstruction (*e.g.* the arm is unrealistically thin). During bundle adjustment, the arm gradually expands as optimization iterations are performed until it is a realistic size (see Fig. 5).

We tested our system on several situations including full body scans and upper body scans. We also tried to scan objects other than human beings. Fig. 8 shows some scan examples. In all the scans that we performed, the Kinect sensor is mounted on a tripod, and we let people turn around freely in front or, in the case of an object, be rotated by the "director" of the scene.

9. Conclusions

We have presented a system which merges a sequence of images from a single range sensor into a unified 3D model, without requiring an initial template. In contrast to previous systems, a wider range of deformations can be handled, including wriggling children. Some limitations remain, however. First, although complex scene topologies can be handled, the topology is restricted to be constant throughout the sequence, and if the coarse-scale reconstruction does not correctly choose the topology, it cannot currently change at the fine scale.

The computational cost is also high. We run our experiments on a desktop PC with 8-core 3.0G Hz Intel Xeon CPU and 64G memory. For a sequence with 400 frames, the partial scan preprocessing stage takes around 30 seconds per frame, the initial alignment and loop closure detection takes about 1 hour, and the final bundle adjustment up to 5 hours. However, these results are using only lightly optimized implementations, and if we were to assume the user intends to 3D print a "shelfie", the 3D printing process will itself take a considerable time. Even if the goal is to upload the model for use in a game, an overnight process remains valuable.

References

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 6

[2] T. J. Cashman and A. W. Fitzgibbon. What shape are dolphins? building 3d morphable models from 2d images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):232–244, 2013. 6

[3] Y. Cui, W. Chang, T. Nöll, and D. Stricker. Kinectavatar: fully automatic body capture using a single kinect. In *Computer Vision-ACCV 2012 Workshops*, pages 133–147. Springer, 2013. 1

[4] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM TOG (Proc. SIGGRAPH)*, 27:1–10, 2008. 2

[5] M. Dou, H. Fuchs, and J.-M. Frahm. Scanning and tracking dynamic objects with commodity depth cameras. In *Proc. ISMAR*, pages 99–106. IEEE, 2013. 1, 2, 3, 4, 7

[6] 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 *IEEE Conf. CVPR, 2009*. 2

[7] T. Helten, A. Baak, G. Bharaj, M. Muller, H.-P. Seidel, and C. Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *Proc. 3DV*, pages 279–286, 2013. 1

[8] C. Hernández, G. Vogiatzis, G. J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *Proc. ICCV*, pages 1–8. IEEE, 2007. 2

[9] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011. 1

[10] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2009)*, 28(5), December 2009. 2, 3

[11] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3d self-portraits. *ACM Trans. Graph.*, 32(6):187, 2013. 1, 3, 4, 7

[12] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong. Modeling deformable objects from a single depth camera. In *Proc. ICCV*, pages 167–174. IEEE, 2009. 2

[13] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. J. Guibas, and H. Pottmann. Dynamic geometry registration. In *Proc. SGP*, pages 173–182, 2007. 2

[14] 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

[15] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. In *SIGGRAPH*, 2007. 2, 3, 4

[16] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014. 5, 6

[17] A. Tevs, A. Berner, M. Wand, I. Ihrke, M. Bokeloh, J. Kerber, and H.-P. Seidel. Animation cartography—intrinsic reconstruction of shape and motion. *ACM TOG*, 31(2):12, 2012. 2

[18] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3D full human bodies using Kinects. *TVCG*, 18(4):643–650, 2012. 1, 2

[19] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM TOG (Proc. SIGGRAPH Asia)*, 31(6):187, November 2012. 2

[20] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. *ACM TOG (Proc. SIGGRAPH)*, 2008. 2

[21] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Transactions on Graphics (TOG)*, 28(5):174, 2009. 2, 7

[22] 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 TOG*, 28:15, 2009. 2

[23] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM TOG*, 30(4):77, 2011. 2

[24] A. Weiss, D. A. Hirshberg, and M. J. Black. Home 3d body scans from noisy image and range data. In *Int. Conf. on Computer Vision (ICCV)*, 2011. 1, 2

[25] C. Wu, C. Stoll, L. Valgaerts, and C. Theobalt. On-set performance capture of multiple actors with a stereo camera. *ACM TOG*, 32(6):161, 2013. 2

[26] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt. Performance capture of interacting characters with handheld kinects. In *Proc. ECCV*, pages 828–841. Springer, 2012. 2

[27] M. Zeng, J. Zheng, X. Cheng, and X. Liu. Templateless quasi-rigid shape modeling with implicit loop-closure. In *Computer Vision and Pattern Recognition (CVPR)*, pages 145–152. IEEE, 2013. 1, 2, 3

[28] Q. Zhang, B. Fu, M. Ye, and R. Yang. Quality dynamic human body modeling using a single low-cost depth camera. In *Computer Vision and Pattern Recognition (CVPR)*, pages 676–683. IEEE, 2014. 2

[29] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (TOG)*, 33(4), 2014. 1, 2, 5