Modeling Deformable Objects from a Single Depth Camera

Miao Liao† Qing Zhang† Huamin Wang‡ Ruigang Yang†
Minglun Gong*
University of Kentucky† Georgia Institute of Technology‡ Memorial University of Newfoundland*

Abstract

We propose a novel approach to reconstruct complete 3D deformable models over time by a single depth camera, provided that most parts of the models are observed by the camera at least once. The core of this algorithm is based on the assumption that the deformation is continuous and predictable in a short temporal interval. While the camera can only capture part of a whole surface at any time instant, partial surfaces reconstructed from different times are assembled together to form a complete 3D surface for each time instant, even when the shape is under severe deformation. A mesh warping algorithm based on linear mesh deformation is used to align different partial surfaces. A volumetric method is then used to combine partial surfaces, fix missing holes, and smooth alignment errors. Our experiment shows that this approach is able to reconstruct visually plausible 3D surface deformation results with a single camera.

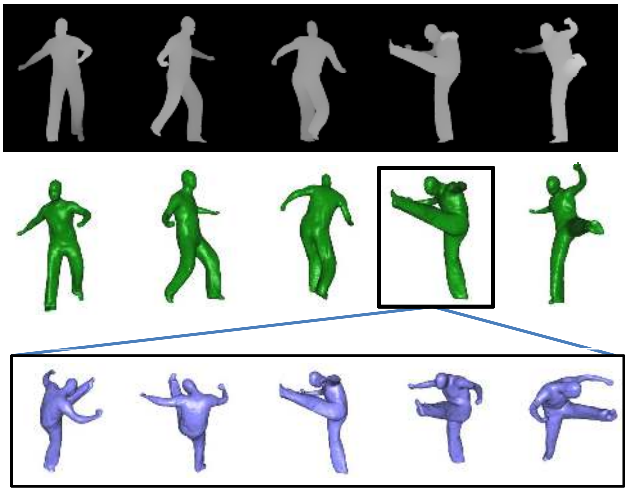

Figure 1. The input to our system is range data (1st row) captured by a single camera at different times, which is simulated with the data shared by [6]. And the output is a sequence of watertight 4D models (2nd row) reconstructed from a dynamic object. The 3rd row shows one 3D model in different views.

1. Introduction

Recent advances in camera self-calibration and stereoscopic vision have made it possible to create high-quality 3D models using a single hand-held camera (*e.g*., [23]) or even from community photo collections [10]. However, most of these techniques are limited to static objects. Typical treatment for dynamic scenes has been widely studied using an array of surrounding cameras, (*e.g*., [15, 33]). Compared to a single camera, a camera array is cumbersome, less affordable, and not practical to carry around for outdoor capturing activities. Using a single depth camera or stereo camera pair to capture a dynamic scene is a challenging problem since it can capture only the visible part of a dynamic object at each time instant. Fortunately, the underlying dynamic nature of the scene can be used to help provide more samples over time. In the simplest case, if the object is rigid or articulated, this *model-completion* task becomes the well-studied Structure-from-Motion (*e.g*., [28, 12]) problem using 3D point registration. Here, we would like to develop a similar technique for time-varying objects deforming arbitrarily but predictably, so that visible partial surfaces can be assembled together to complete a water-tight object surface.

Existing non-rigid Structure from Motion (SFM) techniques (e.g., [3, 29] can only handle small deformation or viewpoint changes. Encouraged by the recent development of full-frame range sensors and the rapid progress in stereo matching research, we expect that color+depth maps captured in the video rate will be practically available soon. The focus of this paper is how to fuse partial deformable surfaces over time to form a complete model. In general, this *deformable model completion* task is an ill-posed problem [29] — the occluded part can be in any shape at any instant. Fortunately most dynamic cases behave continuously in a short temporal interval as we observe in the real world, even though this may not be valid in rare cases when extreme deformation happens under an sudden impulse (such as the popping of a balloon). Under this assumption, we seek to produce a *visually* plausible model that is deforming naturally, and consistent with the input.

Our entire modeling pipeline can be separated into three

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

steps. In the first step, an image sequence is captured using a depth camera (or a stereo camera). Each captured depth map defines a partial surface of a deforming object at each time instant, and we use the image sequence to locate temporal point correspondences. Those correspondences are then used as anchor points in the second step to warp partial surfaces, so they become part of the same object surface at the same time instant. Extended from variational linear mesh deformation approaches [2], we propose a global deformation algorithm in order to warp all partial surfaces together to their destination positions in a single step. After that, partial surfaces are assembled together into a complete watertight surface using a volumetric method in the third step. All surfaces are also optimized in order to complete missing regions and remove remaining errors at the same time. Compared with the ground truth deformation data, our experiment shows that our approach can accurately recover the time-varying 3D shape sequence of a deforming object (as shown in Figure 1).

Our work is closely related to the modeling and motion tracking techniques by Pekelny and Gotsman [22]. Similar to our setup, they also aim to build a complete model over time with a single depth camera. By assuming the deformation as articulated and piecewise rigid, Pekelny and Gotsman [22] can estimate each rigid transformation component and use them to merge partial surfaces over time using the Iterative Closest Point (ICP) method. Our method can deal with both rigid and non-rigid smooth deformations and does not require manual segmentation of different components.

To the best of our knowledge, we present the first method to generate a complete deformable model using a single depth camera. This is made possible by two main technical contributions: a global linear method to fuse all deformable meshes into a complete model and a volumetric method to refine the 4D model for hole-filling and smoothing. With the wider availability of depth sensors, we hope that our approach can eventually push the continued digitalization of our world toward dynamic scenes.

2. Related Work and Preliminary Results

Our proposed framework was motivated by the demonstrated success of Structure from Motion (SFM) techniques(e.g., [23, 10]), which was originally limited to static scenes. It has been recently extended to reconstruct dynamic non-rigid scenes by making extra assumptions about shape deformation. The motion of a non-rigid time-varying object can be decomposed into a rigid transformation and non-rigid deformation. Represented by a set of sparse feature points and their motions, shape deformation has been successfully reconstructed using different models, including a combination of several basic shapes [1, 3], Gaussian distributions [29],or based on Probabilistic Principal Components Analysis [30]. Different from existing techniques, our goal is to generate a complete 3D shape sequence of dynamic objects with large deformations and occlusions.

In order to obtain a complete dynamic model from dynamic scenes, a camera array system is usually deployed to capture objects from different views (e.g., [15, 33]). Surface reconstruction can then be done using either Multiview stereo algorithms [24, 9], or Shape from Silhouette techniques [16, 18, 4]. Unfortunately, missing regions caused by occlusions are still hardly avoidable no matter how many cameras are used in most real cases. How to intelligently fill in these missing regions remains an open problem for the dynamic reconstruction problem. In addition, a multi-camera array system is complicated and cumbersome to use, given the fact that it has to span a large area for enough capturing coverage.

Hole filling is also known as a common problem in the geometric modeling community. Many methods have been developed to address this issue (e.g., [5, 25, 14, 21]). Typically, they are focused on high-quality static models that are acquired using laser range scanner with relatively small missing parts. The problem we are trying to solve here is significantly more challenging. We allow 3D models acquired by time-of-flight (TOF) sensors as our input, since a laser range scanner can hardly capture dynamic scenes. Compared with those from range scanners, TOF sensors contain more noise, and they are *only* 50% complete at most (one depth map for each instant t).

Dynamic reconstruction from sparse cameras has become an active research topic recently in both graphics and vision, due to its usability in many future applications. In addition to Pekelny and Gotsman's work [22], deforming objects are modeled as a 4D hyper-surface in [31] and [19] with spatial-temporal smoothness. Missing regions can then be filled up by sampling over the 4D surface. Unfortunately, they were more focused on dynamic objects with minor deformation, and these techniques will have difficulty in reconstructing complete 3D models with our single-view setup.

3. Method

This work focuses on how to assemble surface patches captured at different time instants for the same dynamic object into a complete 4D space-time model. We assume that the individual surface patches have already been acquired using existing vision techniques [32, 8] or any commercial video-rate depth cameras.

Figure 2 shows the major steps of our algorithm. The inputs to our system are multiple color-depth image pairs captured at different time instants. In the initialization step, correspondences among salient features extracted from different frames are established by any tracking algorithms, typically SIFT in our experiment. In addition, the depth

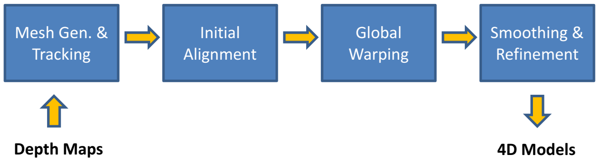

Figure 2. The flow chart of our overall algorithm.

maps are triangulated into 3D meshes. The second step estimates a rigid transformation for each frame to map surface patches from all frames into a reference global coordinate, where they roughly align with each other. A linear mesh deformation method is then applied in the next step to warp one mesh to another, while preserving local details, which is the core of our algorithm. We break down the presentation into several subsections, first introducing the basics for pairwise warping, then extending it to a global alignment scheme. The issue of severe occlusion is also discussed. Finally in the smoothing and refinement step, meshes are merged into a single dynamic 3D model by volumetric methods, with temporal coherence among models from different frames enforced.

3.1. Initial Alignment

Here we decompose the motion of a deformable object into a rigid part and non-rigid part. The goal is to separate a potentially large rigid translation and rotation from a relatively small surface deformation, preventing the later deformation estimation process being biased by the large rigid motion.

Since we do not need to precisely align the surface patches into the reference frame, and all the detailed warping will be handled by the next global alignment step, a rough rigid transformation is enough. The feature correspondences between frames are mapped to 3D point correspondences which could be used to estimate a rigid transformation by absolute orientation([13]). Although absolute orientation is used to estimate the transformation of a rigidly moving object, it can still give a rough estimate of the dominating rigid motion of the object when combined with RANSAC [7]. The transformation between two non-overlapping frames will be estimated by accumulating the pairwise ones between consecutive frames.

3.2. Warping Between Two Consecutive Frames

Different from constraining the problem by traditional epipolar geometry as in [27], we assume the object is under an arbitrary, non-linear deformation in long term, but linearly continuous locally in a short time. Therefore, surface patches can be warped to shapes in neighboring frames using linear mesh deformation, given sufficient feature point correspondences.

Let M be a polygon mesh defined by a pair (V, K) of vertices $V = \{\vec{v}_1, ..., \vec{v}_n\}$ and edges K, the 1-ring neighborhood of a vertex $\vec{v}_i$ is the set of its adjacent vertices $N_i = \{j|(i,j) \in K\}$ and the degree d_i denotes the number of vertices in N_i.

Given $\vec{u}_i$ on M_0 and $\vec{v}_i$ on M_1 be a correspondence pair representing the same feature point over a deforming object, the warping process on mesh M_1 from t_1 to t_0 changes $\vec{v}_i$ to $\vec{v}_i'$, which should be close to $\vec{u}_i$, implying that M_0 and M_1 will represent part of the same object. On the other hand, the warping process should minimize the mesh deformation as much as possible in order to maintain shape details. This can also be considered as a constraint to calculate warping over uncontrolled vertices in M_1.

In the mesh deformation community, surface shapes are usually described locally by Laplacian coordinates for vertices. The Laplacian coordinate $L(\vec{v}_i)$ for vertex $\vec{v}_i$ is defined by applying the Laplican-Beltrami operator over the vertex coordinate:

$$L(\vec{v}_i) = \frac{1}{\sum\limits_{j \in N_i} w_{ij}} \sum_{j \in N_i} w_{ij}(\vec{v}_i - \vec{v}_j) \tag{1}$$

$$w_{ij} = \tfrac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}) \tag{2}$$

in which α_{ij} and β_{ij} are two angles opposing to the edge (i,j) as in [17]. When the mesh is regular and nearly uniformly defined, Equation 1 can be simplified as:

$$L(v_i) = \vec{v}_i - \frac{1}{d_i} \sum_{j \in N_i} \vec{v}_j \tag{3}$$

Since the goal of a warping procedure is to move specified control points closer to their target positions and still maintain the mesh shape as much as possible, mathematically, this can formulated as a quadratic energy functional minimization problem as in [26]:

$$E(V') = \sum_{i \in V} \left\| L(\vec{v}_i) - L(\vec{v}_i') \right\|^2 + \sum_{i \in F} \left\| \vec{v}_i' - \vec{u}_i \right\|^2 \tag{4}$$

in which V' is the vertex position after warping, and F is the correspondence subset ($F \subseteq V$). The first sum measures the shape similarity before and after warping using Laplacian coordinates, whose least square solution is a linear system:

$$\mathbf{M}_L \mathbf{V}' {=} \mathbf{L}, \quad \mathbf{V}' {=} \begin{bmatrix} \vec{v}_1' \\ \vec{v}_2' \\ \ldots \\ \vec{v}_V' \end{bmatrix}, \quad \mathbf{L} {=} \begin{bmatrix} L(\vec{v}_1) \\ L(\vec{v}_2) \\ \ldots \\ L(\vec{v}_V) \end{bmatrix} \tag{5}$$

$\mathbf{M}_L$ is the laplacian matrix of the mesh. The second term gives the sum of squared differences over all control points, whose solution is given by:

$$\mathbf{M}_I \mathbf{V}' {=} \mathbf{U}, \quad \mathbf{U} = \begin{bmatrix} u_{i0} \\ u_{i1} \\ \ldots \\ \ldots \end{bmatrix} \tag{6}$$

Similar to an identity matrix, $\mathbf{M}_I$ is a non-square matrix composed of zeros and ones, in which a row stands for a control vertex and a column stands for a mesh vertex. Each row has exactly one non-zero entry if and only if that control vertex is the corresponding mesh vertex.

Stacking $\mathbf{M_L}$ and $\mathbf{M_I}$ together, we obtain an over-determined linear system in order to find the overall least square solution to Equation 4:

$$\begin{bmatrix} \mathbf{M}_L \\ \mathbf{M}_I \end{bmatrix} \mathbf{V}'= \begin{bmatrix} \mathbf{L} \\ \mathbf{U} \end{bmatrix} \tag{7}$$

Since X, Y and Z coordinates are independent in Equation 4, they can either be solved separately in three matrix systems, or simultaneously as a single system, in which case the matrix system will be three times larger.

Equation 7 produces plausible applaudable results when the deformation is small, but if the shape undergoes large rotation or scaling, the Laplacian coordinate is not a good descriptor since it is well known as affine-variant. The Laplacian coordinate of a vertex is actually a vector in 3D space that originates from the centroid of its neighbors and ends at the vertex. For example, if a mesh performs rotation, the Laplacian coordinates of its vertices should also rotate with the same angle along the same axis. Unfortunately, this cannot be properly handled by Laplacian coordinates in Equation 4. In order to address this issue, we apply an explicit affine transformation together with the warping procedure to account for any large affine transformation; thus the first term in Equation 4 becomes:

$$\sum_{i \in V} \|T_i L(\vec{v}_i) - L(\vec{v}'_i)\|^2 \tag{8}$$

in which T_i is an estimated local affine transformation for $\vec{v}_i$. Since the Laplacian coordinate is determined by the 1-ring neighborhood of $\vec{v}'_i$, T_i can be calculated from the 1-ring neighborhood as well. By describing T_i as a function of V', the estimation of T_i can be implicitly contained into a single system with V' as the only unknowns. Unfortunately, this becomes a non-linear optimization problem since T_i is nonlinearly determined by V', which is known to be difficult to solve. Instead of using an exact solution, we adopted the method proposed in [26] to simply approximate T_i as a linear function of V' if the rotation angle is small. Specifically, we first define T_i in the homogeneous coordinates:

$$T_i = \begin{bmatrix} s & -h_3 & h_2 & t_x \\ h_3 & s & -h_1 & t_y \\ -h_2 & h_1 & s & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

By definition, an optimal T_i should minimize the following functional:

$$\sum_{k \in \{i\} \cup N_i} \|T_i \vec{v}_k - \vec{v}'_k\|^2 \tag{10}$$

Let $\mathbf{t}_i = (s, h_1, h_2, h_3, t_x, t_y, t_z)^T$ be the vector of the unknowns in T_i, Equation 10 can be rewritten as:

$$\|A_i \mathbf{t}_i - \mathbf{V}_{N_i}\|^2 \tag{11}$$

in which

$$A_i = \begin{bmatrix} v_{k_x} & 0 & v_{k_z} & -v_{k_y} & 1 & 0 & 0 \\ v_{k_y} & -v_{k_z} & 0 & v_{k_x} & 0 & 1 & 0 \\ v_{k_z} & v_{k_y} & -v_{k_x} & 0 & 0 & 0 & 1 \\ \vdots & & & & & & \end{bmatrix} \tag{12}$$

and

$$\mathbf{V}_{N_i} = \begin{bmatrix} v'_{k_x} \\ v'_{k_y} \\ v'_{k_z} \\ \vdots \end{bmatrix} \tag{13}$$

is a vertex vector of $\vec{v}'$ and its neighborhood. $\mathbf{t}_i$ can then be calculated from $\mathbf{V}_{N_i}$:

$$\mathbf{t}_i = (A_i^T A_i)^{-1} A_i^T \mathbf{V}_{N_i} \tag{14}$$

By creating a new matrix Δ_i using the coordinates in $L_(v_i)$ as follows:

$$\Delta_i = \begin{bmatrix} L_x(\vec{v}_i) & 0 & L_z(\vec{v}_i) & -L_y(\vec{v}_i) & 1 & 0 & 0 \\ L_y(\vec{v}_i) & -L_z(\vec{v}_i) & 0 & L_x(\vec{v}_i) & 0 & 1 & 0 \\ L_z(\vec{v}_i) & L_y(\vec{v}_i) & -L_x(\vec{v}_i) & 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

we can then calculate $T_i L(\vec{v}_i)$ as:

$$T_i L(\vec{v}_i) = \Delta_i \mathbf{t}_i = \Delta_i (A_i^T A_i)^{-1} A_i^T \mathbf{V}_{N_i} \tag{16}$$

$D_i = \Delta_i (A_i^T A_i)^{-1} A_i^T$ is solely defined on V, so it can be computed in advance, meaning that $T_i L(\vec{v}_i)$ is linear function of $\mathbf{V}_{N_i}$. By stacking all $T_i L(\vec{v}_i)$ together into a large vector $\mathbf{T}_L$, a large sparse matrix $\mathbf{M}_T$ can be constructed using submatrices $\{D_i\}$ such that:

$$\mathbf{T}_L = \mathbf{M}_T \mathbf{V}' \tag{17}$$

Replace the right hand side of equation 5 with $\mathbf{T}_L$:

$$\mathbf{M}_L \mathbf{V}' = \mathbf{M}_T \mathbf{V}' \tag{18}$$

or,

$$(\mathbf{M}_L - \mathbf{M}_T)\mathbf{V}' = 0 \tag{19}$$

Replacing the corresponding part in equation 7 with equation 20, we get the linear equation that can stitch two pieces of mesh with rotation and scaling between them.

$$\begin{bmatrix} (\mathbf{M}_L - \mathbf{M}_T) \\ \mathbf{M}_I \end{bmatrix} \mathbf{V}' = \begin{bmatrix} \mathbf{0} \\ \mathbf{U} \end{bmatrix} \tag{20}$$

3.3. Warping All Frames Simultaneously

A naive approach to obtain a complete 3D shape is to simply assemble two surface patches each time. For example, in order to create the shape surface at frame i, the surface patch at frame 1 is first combined with frame 2 into a new surface $1-2$ which is combined with frame 3, so on and so forth. By keeping doing so we combine all surface patches together into the desired shape at frame i. Since local temporal correspondences between two adjacent frames determine the warping procedure in each step, errors can be easily accumulated from frame to frame, causing misalignment between surface patches. Another reason we need a global method is that sequential warping cannot deal with occlusion. We will discuss this in detail in the occlusion handling section.

We develop a global warping algorithm in order to warp surface patches in all frames altogether to the destination frame in a single step. This gives a single linear system with unknowns as the final positions of vertices in the destination frame. As an extension from the local warping algorithm, a global warping matrix system lists all Laplacian constraints as Diagonal sub-matrices as shown in Equation 21.

$$\begin{bmatrix} \mathbf{Q}_1 & & & & & & \\ & \mathbf{Q}_2 & & & & & \\ & & \ddots & & & & \\ & & & \mathbf{Q}_{d-1} & & & \\ & & & & \mathbf{Q}_{d+1} & & \\ & & & & & \ddots & \\ & & & & & & \mathbf{Q}_n \end{bmatrix} \begin{bmatrix} \mathbf{V'_1} \\ \mathbf{V'_2} \\ \vdots \\ \mathbf{V'_{d-1}} \\ \mathbf{V'_{d+1}} \\ \vdots \\ \mathbf{V'_n} \end{bmatrix} = 0 \tag{21}$$

in which $\mathbf{Q} = \mathbf{M_L} - \mathbf{M_T}$ is the Laplacian constraint, n is the number of frames, and d is the destination frame index. Feature correspondence constraints are imposed by adding more rows into equation 21. For example, if a correspondence exists from frame i to frame d, the following row will be added to the matrix system in equation 21 as a correspondence constraint:

$$\begin{bmatrix} 0 & \dots & 0 & \underset{k^{th}}{1} & 0 & \dots & 0 \end{bmatrix} \tag{22}$$

k is the index of the vertex into the vector of unknowns in equation 21. Accordingly, the position of the corresponding vertex in frame d should be added to the end of the vector on the right hand side of equation 21. If frame i has a correspondence point in frame j, which is also unknown, a different row will need to be added to the matrix:

$$\begin{bmatrix} 0 & \dots & \underset{k^{th}}{1} & \dots & \underset{h^{th}}{-1} & \dots & 0 \end{bmatrix} \tag{23}$$

Here, k and h indicate the matching vertices' position in the vector of unknowns, and a 0 should be added to the end of the right hand side vector ensuring that these two vertices be at the same 3D position after warping.

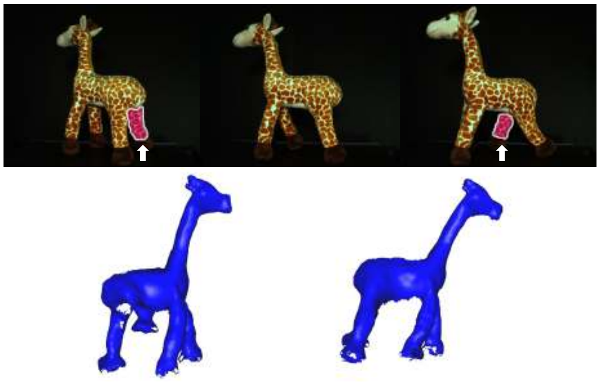

Figure 3. The need for occluded feature interpolation: 1^{st} row from left to right: frame 16, 17 and 18 of a walking giraffe toy. Note that one leg is completely occluded in frame 17. 2^{nd} row shows the reconstructed results of frame 17 without and with the occlusion handling. As can be seen in the left image, the occluded leg is largely distorted. After features are interpolated, it is corrected in the right image.

This global method is in spirit similar to *bundle adjustment*. However our formulation is linear while typical bundle adjustment is formulated as non-linear optimization that requires iterative methods.

3.4. Occlusion Handling

Since both the camera viewpoint is moving and the object is deforming, the least square doesn't necessarily yield the correct position. As illustrated in figure 3, to complete the 3D model of frame 17, the occluded leg needs to be recovered with the information from its neighboring frames 16 and 18. One possibility is that this leg is topologically connected to the visible surfaces, and will be pulled to a certain position under the laplacian constraint. This approach will result in incorrect warping (figure 3) when the leg itself is moving during these frames.

A more sophisticated approach is to use the tracked features to predict their occluded positions. The features are first extracted and stored for each frame. In the next step, we establish a global feature pool by searching the feature set of each frame for those that are visible in multiple frames. The global features are recorded along with their frame numbers and corresponding 3D positions. With those globally tracked features, the occluded part can be interpolated or extrapolated under the continuous motion assumption. Figure 3 shows the correct result by this method.

3.5. Smoothing and Refinement

After the warping procedure, surface patches are aligned together to cover the shape of an object at the same time instant. In this section, they will be merged together to form a single surface. Instead of manipulating meshes directly, we choose to use an Eulerian approach by volumetric representation for several major reasons. First of all, a volumetric representation can easily handle topological changes among different meshes. Secondly, a volumetric representation can fix missing holes and misalignments, which are often caused by image noise, correspondence errors, occlusions or other errors. Last but not least, an Eulerian approach is straightforward to implement and it does not require the complicated re-meshing process.

We first define a distance function $d(\vec{x})$, which gives the minimum absolute distance from $\vec{x}$ to any surface patches. Let ϕ be a signed distance function representing the final steady shape we would like to achieve, the level set formulation is:

$$\frac{\partial \phi}{\partial t} = \left(\nabla d \cdot \frac{\nabla \phi}{|\nabla \phi|} + d \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi| \tag{24}$$

Intuitively, ϕ will first be smoothed by a mean curvature flow. Once it gets close to surface patches d, the smoothing effect will be gradually reduced and it will cover all surface patches. As an example, in Figure 6, the holes by invisibility are filled up with this method. Details of this technique can be found in [20].

4. Experimental Results

We have tested our algorithms on both synthetic data and real data. The synthetic data is generated with the 4D models shared by [6]. As in figure 5, for each frame, the renderer outputs the depth map and tracked 3D points specified in advance. 400 out of 20000 vertices are specified as tracking points, which are uniformly distributed on the object surface. Since this data set has no color features to track, we generate correspondences directly.

Figure 5 shows results with correct correspondences. The complete 3D model is recovered. We further evaluated the performance of our algorithms under imperfect tracking by perturbing the original matching by some amount. Specifically, 1-pixel perturbation means matching a pixel to one that is randomly selected from the 1-ring neighbor pixels of its true correspondence. 3-pixel, 5-pixel, and 10-pixel perturbations are performed in the similar way. One pixel distance is approximately 10mm in the real world. So, 1-pixel perturbation is roughly a 1% perturbation in the real 3D space. Figure 4 shows the reconstructed results under the perturbations. It can be seen that when the amount of perturbation increases, the fine details are lost, nevertheless the overall shape is always well recovered. Table 1 shows the errors between the reconstructed model of frame 1 and the ground truth.

The real data is captured by a SwissRanger depth camera combined with a point grey flea color camera that provides texture information. The depth camera can produce 176×144 depth map at video rate. However, the quality of the depth map drops quite significantly for dynamic scenes. So we manually animate the toy giraffe and use temporal averaging to improve the signal-to-noise ratio of the depth map. SIFT features are extracted for each frame and features are tracked across different frames by searching in the pool of the extracted ones. The depth camera and color camera are almost coaxial so the mapping between their images can be approximated by a homography. There are about 200 features that are reliably tracked. Results in figure 6 show that the occlusion can be well handled by our algorithms. Figure 7 is a comparison of the models from frame 5 before and after hole-filling and smoothing. Figure 8 shows the reconstruction of a T-shirt worn by a person who turned around 360 degrees in front of our capture device, while moving his hands up and down.

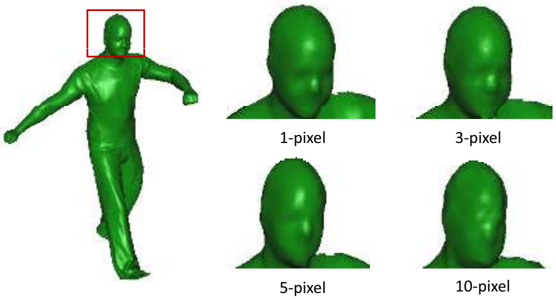

Figure 4. The comparison of results from different perturbations. As perturbation amount increases, details are lost.

	DimX	DimY	DimZ	Max Dist	Avg Dist
1-p	844.4	1273.7	1814.0	13.60	1.927
3-p	844.4	1273.7	1814.0	14.78	2.278
5-p	844.4	1273.7	1814.0	16.95	2.646
10-p	844.4	1273.7	1814.0	20.02	3.487

Table 1. The errors between reconstructed model of frame 1 and the ground truth. From row 2 to row 5: 1-pixel, 3-pixel, 5-pixel and 10-pixel perturbation. DimX, DimY, and DimZ are the size of the model in x,y,z dimensions. Max Dist and Avg Dist are the maximum and average distance between the result and ground truth. Details are lost when noise increases.

5. Conclusion

We have developed a novel approach to reconstruct complete 3D surface deformation over time by a single camera. The deformable surface patches are stitched together by mesh deformation in a global manner, and merged into a complete model by a volumetric method. Test on both synthetic and real data demonstrated that our approach works well with even large deformation. we believe our approach will help to simplify the difficult task of creating time-varying models for dynamic objects.

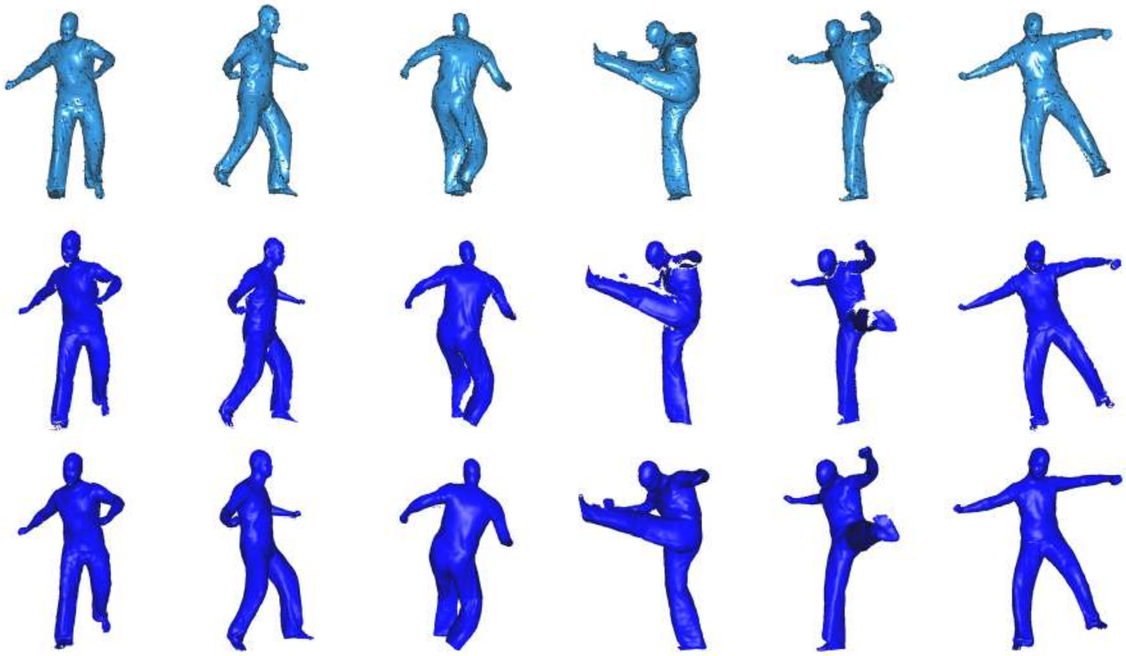

Figure 5. Six frames (frame 1, 10, 17, 24, 29 and 35 out of all 38 frames) are shown in this figure. 1^{st} row shows the rendered models. The black dots indicate the tracked features. $2^{n}d$ row is the partial meshes constructed from depth maps. The 3^{rd} row shows the reconstructed 3D models by our algorithm.

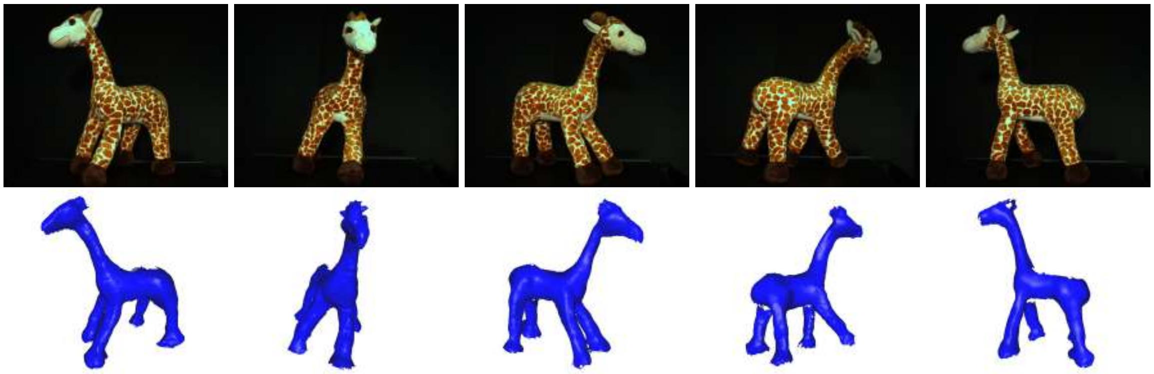

Figure 6. Frame 3,5,7,11 and 17 out of total 18 frames are shown as an example here. 1^{st} row shows the color images of the toy giraffe. 2^{nd} row shows the recovered models.

References

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of SIGGRAPH*, 1999. 2

[2] M. Botsch and O. Sorkine. On linear variational surface deformation methods. In *Transcation on visualization and Computer Graphics*. IEEE, 2008. 2

[3] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, 2000. 1, 2

[4] G. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *CVPR*, 2000. 2

[5] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996. 2

[6] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *Siggraph*. ACM, 2008. 1, 6

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Comm. of the ACM*, 1981. 3

[8] P. Fong and F. Buron. Sensing deforming and moving objects with commercial off the shelf hardware. In *Computer Vision and Pattern Recognition*. IEEE, 2005. 2

[9] M. Goesele, B. Curless, and S. Seitz. Multi-view stereo revisited. In *Proceedings of CVPR*, 2006. 2

[10] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-View Stereo for Community Photo Collections. In *ICCV*, 2007. 1, 2

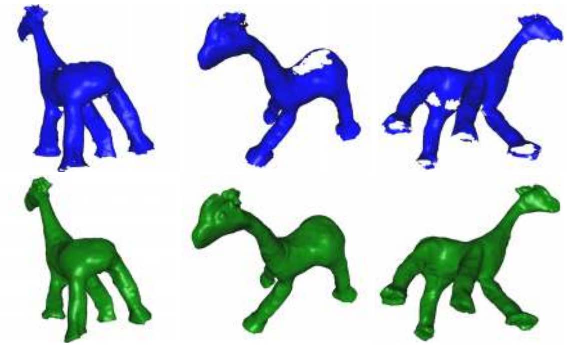

Figure 7. Three different views of the reconstructed model from frame 5 of the real data. The watertight model after smoothing is shown on the 2^{nd} row.

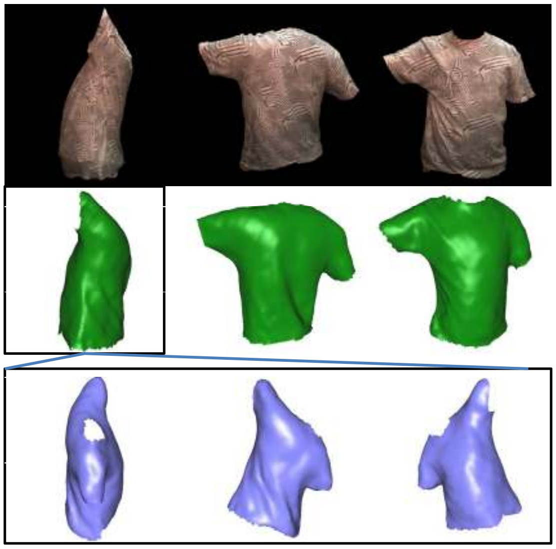

Figure 8. 1^{st} and 2^{nd} rows, from left to right: Frame 5, 7, 14 out of total 14 frames of a deforming shirt. 3^{rd} row shows 3 different views of the 3D model of frame 5.

[11] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *Proceedings of ICCV*, 2007.

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1

[13] B. K. Horn. Colosed-form solution of absolute orientation using unit quaternions. In *Journal of the Optical Society of America*, 1987. 3

[14] T. Ju. Robust repair of polygonal models. *ACM Transactions on Graphics*, 23(3):888–895, 2004. 2

[15] T. Kanade, P. Rander, S. Vedula, and H. Saito. Virtualized reality: Digitizing a 3d time-varying event as is and in real time. In *Mixed Reality, Merging Real and Virtual Worlds*, pages 41–57. 1999. 1, 2

[16] A. Laurentini. The Visual Hull Concept for Silhouette Based Image Understanding. *IEEE PAMI*, 16(2):150–162, February 1994. 2

[17] D. M., M. M., S. P., and B. H. Implicit Fairing of Irregular Mesher Using Diffusion and Curvature Flow. In *In Proceedings of Siggraph*, pages 317–324, 1999. 3

[18] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-Based Visual Hulls. In *Proceedings of SIGGRAPH 2000*, 2000. 2

[19] N. J. Mitra, S. Flory, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann. Dynamic geometry registration. In *Eurographics Symposium on Geometry Processing*, 2007. 2

[20] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002. 6

[21] S. Park, X. Guo, H. Shin, and H. Qin. Shape and appearance repair for incomplete point surfaces. In *Proceedings of ICCV*, pages 1260–1267, 2005. 2

[22] Y. Pekelny and C. Gotsman. Articulated object reconstruction and markerless motion capture from depth video. In *Eurographics*, 2008. 2

[23] M. Pollefeys, R. Koch, and L. V. Gool. Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters. In *ICCV*, 1998. 1, 2

[24] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of CVPR*, pages 519–526, 2006. 2

[25] A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. *ACM Transactions on Graphics*, 23(2):878–887, 2004. 2

[26] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rossl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2004. 3, 4

[27] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method. In *Technical Report CMU-CS-91-105 Carnegie Mellon University*, 1991. 3

[28] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: A Factorization Approach. *IJCV*, 9(2):137–154, 1992. 1

[29] L. Torresani, A. Hertzmann, and C. Bregler. Learning non-rigid 3d shape from 2d motion. In *In proceedings of NIPS*, 2003. 1, 2

[30] L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE PAMI*, To appear. 2

[31] M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *Eurographics Symposium on Geometry Processing*, 2007. 2

[32] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. spacetime faces: High resolution capture for modeling and animation. In *Transaction on Graphics*. ACM, 2004. 2

[33] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics,*, 23(3):600–608, 2004. 1, 2