Tracking Complete Deformable Objects with Finite Elements

Stefanie Wuhrer
Saarland University
swuhrer@mmci.uni-saarland.de

Jochen Lang
University of Ottawa
jlang@site.uottawa.ca

Chang Shu
National Research Council of Canada
chang.shu@nrc-cnrc.gc.ca

Abstract

We present an approach to robustly track the geometry of an object that deforms over time from a set of input point clouds captured from a single viewpoint. The deformations we consider are caused by applying forces to known locations on the object's surface. Our method combines the use of prior information on the geometry of the object modeled by a smooth template and the use of a linear finite element method to predict the deformation. This allows the accurate reconstruction of both the observed and the unobserved sides of the object. We present tracking results for noisy low-quality point clouds acquired by a stereo camera and a Kinect sensor, and simulations with point clouds corrupted by different error terms. We show that our method is also applicable to large non-linear deformations.

1. Introduction

The accurate acquisition of physical objects has numerous applications in the entertainment industry and in inspection systems. While there exist commercial systems for digitizing rigid objects, the acquisition of deforming objects remains a challenge due to the complex changes in geometry over time. A rigid object can be scanned sequentially from multiple viewpoints to accurately capture the complete surface, whereas scanning the entire surface of a deforming object would require a complex and expensive physical setup involving multiple synchronized sensors, which may still be subject to occlusions.

Recently, several techniques were proposed that solve this problem by using a template shape as a geometric and topological prior for the reconstruction and by deforming the template to fit to the observed data [7, 25, 11, 5]. In some of these methods, the observed data comes from a set of single-view scans. The assumption of input from a single viewpoint is useful due to the simplicity of acquiring this type of data. Template-based tracking approaches are shown to lead to visually pleasing results for numerous examples. However, the deformation of the unobserved side of the object is generally only guided by a smoothness cost.

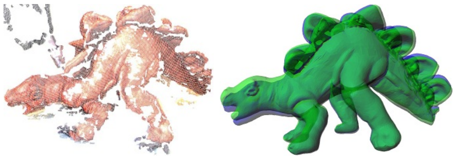


Figure 1. *Influence of FEM step after 26 frames. The left shows the input data and the right shows the results. The result using only the tracking step is visualized in green, and the result using tracking and FEM is visualized in blue.*

We combine a tracking-based approach with fitting a volumetric elastic model to improve the estimation of the unobserved side of the object. We employ a linear finite element method (FEM) to solve for physical deformations when a given force is applied. Our method proceeds in two steps: First, we use a tracking approach to deform the template model. Second, we use the offsets of the observed vertices of the template mesh found using the tracking step in a FEM to predict the offsets of the unobserved vertices. Hence, rather than smoothly deforming the unobserved side of the model, we deform the unobserved side by taking into account volumetric information. We repeatedly linearize the deformation in the FEM at its current deformation state. Note that our method allows for tracking data acquired using single, multiple, or moving viewpoints.

While deformable models have been introduced to computer vision and computer graphics 30 years ago [21], we combine modern non-rigid template-based tracking with a volumetric elastic model for completion of the deformation at the unobserved side only. Our major contributions are therefore: (a) the use of a FEM-based model to deform the unseen side leading to more physically plausible results than by using a smoothness cost in the template-based tracking, and (b) tracking linear and non-linear deformations by repeatedly linearizing the FEM model at its current deformation state.

2. Related Work

We review relevant work related to tracking surfaces and predicting shape deformations using finite element models.

Tracking. Computing the correspondence between deformed shapes has received considerable attention in recent years [24]. This review focuses on techniques that do not employ a priori skeletal models or marker positions, as we aim to minimize assumptions about the structure of the surface. None of the following approaches combine physics-based models with tracking approaches.

Template-based approaches. The following techniques solve the tracking problem using a template as shape prior. De Aguiar *et al.* [7] and Vlasic *et al.* [25] tracked multi-view stereo data of a human subject that is acquired using a set of cameras. Tung and Marsuyama [23] extended this type of approach by using 3D shape features to aid the tracking. Li *et al.* [11] proposed a generic data-driven technique for mesh tracking, where a template is deformed to each observed frame. A deformation graph is used to derive a coarse-to-fine strategy that decouples the complexity of the original mesh geometry from the representation of the deformation. Cagniart *et al.* [5] proposed an alternative where the template is decomposed into a set of patches. The template is then deformed to each observed frame using a data term that encourages inter-patch rigidity. We however combine a template with a FEM step.

Template-free approaches. The following techniques solve the tracking problem using prior information on the deformation of the object. Mitra *et al.* [15] modeled the surface tracking problem as a problem of finding a smooth space-time surface in four-dimensional space. Sharf *et al.* [19] used a similar concept to find a volumetric space-time solid. Wand *et al.* [26] used a probabilistic model based on Bayesian statistics to track a deformable model. Tevs *et al.* [22] extended this approach by first tracking a few stable landmarks and by subsequently computing a dense matching. Furukawa and Ponce [8] proposed a technique to track data from a multi-camera setup by computing the polyhedral mesh that captures the first frame and by deforming this mesh to the data in subsequent frames. Liao *et al.* [13] take images acquired using a single depth camera from different viewpoints while the object deforms and assemble them into a complete deformable model over time. Popa *et al.* [17] used a similar approach that is tailored to allow for topological consistency across the motion. Li *et al.* [12] avoid the use of a template model by initializing the tracking procedure with the visual hull of the object. Zheng *et al.* [27] track a deformable model using a skeleton-based approach, where the skeleton is computed using the data.

Predicting Shape Deformations Using FEM. Several authors suggested learning the parameters of linear finite element models from a set of observations. We use such a method in combination with a tracking method to find an accurate tracking result of the observed and the unobserved side of the model. For a summary of linear finite element methods, refer to Bro-Nielsen [4]. Lang and Pai [10] used a surface displacement and contact force at one point along with range-flow data to estimate elastic constants of homogeneous isotropic materials. Becker and Teschner [2] presented an approach to estimate the elasticity parameters for isotropic materials using a linear finite element method. Syllebranque and Boivin [20] estimate the parameters of a quasi-static finite element simulation of a deformable solid object from a video sequence. Schnabel *et al.* [18] used finite element models to validate the non-rigid image registration of magnetic resonance images. Nguyen and Boyce [16] presented an approach to estimate the anisotropic material properties of the cornea. Bickel *et al.* [3] proposed a physics-based approach to predict shape deformations. Recently, Choi and Szymczak [6] used FEM to predict a consistent set of deformations from a sequence of coarse watertight meshes but their method does not apply to single-view tracking, which is the focus of our method.

3. Overview

The input to our method consists of a closed template T, a contact point on T and a force direction that leads to the deformation of the model, and a set of 3D video frames (point clouds) capturing the deformation. We assume that the template has roughly the shape of the object before the deformation, and is approximately aligned to the first frame.

We aim to track the deformation of the object by computing a deformed template shape for each of the captured frames. Let p_i denote the vertices of T, let $\vec{p_i}$ denote their position vectors, and let $\tilde{p}_i$ denote their homogeneous coordinates. We deform T by applying a 4×4 transformation matrix A_i to each $\tilde{p}_i$ of T. The transformation matrix A_i depends on six parameters: three parameters $\vec{t_i}$ describing a translation, two parameters $\vec{r_i}$ describing a rotation axis, and one parameter ϕ_i describing a rotation angle. That is, A_i describes a rigid transformation as

$$A_i = A_{trans}(\vec{p_i}) A_{rot}(\vec{r_i}, \phi_i) A_{trans}(\vec{t_i}) A_{trans}(-\vec{p_i}), \qquad (1)$$

where $A_{trans}(-\vec{p_i})$ expresses a point in a coordinate frame centered at p_i and $A_{rot}(\vec{r_i}, -\phi_i)$ expresses a point in a coordinate frame rotated by angle ϕ_i around axis $\vec{r_i}$. Expressing the transformation in a coordinate system centered at p_i has the advantage that differences between the transformations of neighboring vertices can be measured directly.

The main idea of the approach is to combine tracking the point cloud data and predicting the deformation on the unseen side of the model using a FEM. To track the data, we use an energy optimization approach that aims to find parameters $\vec{t_i}, \vec{r_i}$, and ϕ_i that lead to a smooth, intersection-free mesh that is close to the observed data. Afterwards, we displace the vertices of T that are not observed in the data using a linear FEM with the given contact point and force direction. Finally, we readjust the parameters $\vec{t_i}, \vec{r_i}$, and ϕ_i to take this displacement into account. When multiple frames $F_1, F_2 \ldots, F_n$ are recorded, we use the result of F_j to update the vertex positions of the template for F_{j+1}.

4. Tracking of Point Cloud Data

This section presents our energy optimization approach to find the transformation parameters $\vec{t_i}, \vec{r_i}$, and ϕ_i that lead to a mesh that is close to the point cloud data.

We aim to deform the template T to the observed data. When fitting T to the first frame, we start by deforming T using a global rigid transformation A to fit the observed data as much as possible. We consider all vertices p_i of T and compute the nearest neighbor $N(p_i)$ of the deformed point $A\tilde{p_i}$ in the point cloud data. In the following, $N(p_i)$ is represented in homogeneous coordinates.

To rigidly align T to F, we find A by minimizing

$$E_{initial} = \sum_i \omega_i \left\| A\tilde{p_i} - N(p_i) \right\|_2^2 \tag{2}$$

with respect to seven degrees of freedom (one for scaling, three for rotation, three for translation), where ω_i is a weight term. The scaling term accommodates slight errors in the calibration of the 3D scanner used to acquire the template shape and/or the calibration of the camera system used to acquire the frames. The weight ω_i is one if the angle between the normal of the deformed template T at p_i and the normal of F at $N(p_i)$ is less than a threshold α and if the distance between $A\tilde{p_i}$ and $N(p_i)$ is at most dr, otherwise it is zero. Here, r is the average edge length of the undeformed template T and d and α are parameters.

To fit T to any frame, we deform T in a non-rigid fashion by changing $\vec{t_i}, \vec{r_i}$, and ϕ_i to minimize the energy

$$\begin{aligned}
E_{track} &= w_{data} E_{data} + w_{sm} E_{sm} + w_{md} E_{md} \text{ with} \\
E_{data} &= \textstyle\sum_i \omega_i \left\| A_i \tilde{p}_i - N\left(p_i\right) \right\|_2^2 \\
E_{sm} &= \textstyle\sum_i \frac{1}{|R_{sm}(p_i)|} \sum_{j \in R_{sm}(p_i)} \left(1 - \frac{\left\| p_i - p_j \right\|_2^2}{(s_{sm} r)^2} \right) \\
&\quad \left(\left\| \vec{t_i} - \vec{t_j} \right\|_2^2 + \left\| \vec{r_i} - \vec{r_j} \right\|_2^2 + (\phi_i - \phi_j)^2 \right) \\
E_{md} &= \textstyle\sum_i \frac{1}{|R_{md}(p_i)|} \sum_{j \in R_{md}(p_i)} \left(1 - \frac{\left\| A_i \tilde{p}_i - A_j \tilde{p}_j \right\|_2^2}{(s_{md} r)^2} \right) \\
&\quad \left(\frac{1}{\left\| A_i \tilde{p}_i - A_j \tilde{p}_j \right\|_2^2} \right)^2
\end{aligned} \tag{3}$$

where w_{data}, w_{sm} and w_{md} are weights for the individual energy terms, $R_{sm}(p_i)$ is the set of indices corresponding to points $A_j\tilde{p}_j$ that have geodesic distance at most $s_{sm}r$ of $A_i\tilde{p}_i$, $R_{md}(p_i)$ is the set of indices corresponding to points $A_j\tilde{p}_j$ that have Euclidean distance at most $s_{md}r$ from $A_i\tilde{p}_i$ and that have geodesic distance at least $s_{md}r$ from $A_i\tilde{p}_i$, and $|.|$ denotes the cardinality of a set. As before, r is the average edge length of T and s_{sm} and s_{md} are parameters. We associate the vertex p_i with the nearest neighbor $N(p_i)$ of the deformed point $A_i\tilde{p}_i$ in the point cloud data. The weight ω_i is set according to the angle between the normal vectors and the distance to the nearest neighbor as above. The transformation matrices A_i are computed according to Equation 1.

The data term of the energy drives the template mesh to the observed data. Using only this term results in an ill-posed problem as it defines at most three constrains per vertex to estimate the six parameters per vertex. We therefore add the smoothness term to act as a regularization term that encourages smooth transformations. A similar term was previously used by Allen *et al.* [1]. The last term, E_{md}, discourages self-intersections by penalizing close-by vertices that are far in the geodesic sense. Note that while this energy term does not guarantee that self-intersections are avoided, it acts as a repelling term between close-by vertices. Unlike previous tracking approaches, we use this energy term since self-intersections might cause problems in the FEM step.

We start by encouraging smooth transformations by setting $w_{sm} = 1000$. Similar to Li *et al.* [11], whenever the energy does not change by much, we relax the smoothness weight as $w_{sm} = w_{sm}/2$ to give more weight to the data term. We stop when the relative change in energy $\left(E_{track}^{(i-1)} - E_{track}^{(i)}\right)/E_{track}^{(i-1)}$, where i is the iteration number, is less than 0.0001 or when w_{sm} is smaller than 100. Throughout the optimization, we fix $w_{md} = 0.1$. For a fixed set of weights, we do not update the respective associated point $N\left(p_i\right)$ for any vertex p_i.

Recall that the distance to the nearest neighbor used in E_{data} is limited by the template resolution. To allow for larger deformations, we use a multi-resolution approach as follows. We compute a multi-resolution hierarchy of T by collapsing edges of the mesh according to Garland and Heckbert's geometry criterion [9]. We do not collapse edges if this would lead to a self-intersecting mesh. We perform the test whether an edge collapse leads to self-intersections greedily by performing the collapse and testing if self-intersections occur. In each resolution step, we halve the number of vertices. We stop the collapse when the base mesh contains about 1000 vertices or when no more valid edges can be found for the edge collapse operation. For the base mesh, we initialize all of the transformation parameters to either identity (i.e. $\vec{t_i}$ is the null vector and ϕ_i is zero) or to the result of the previous frame and minimize E_{track}. Once the energy is minimized for resolution level l, we consider the mesh of the next higher resolution level $l+1$. For the vertices of level $l+1$ that are present in level l, we initialize the transformation parameters to the result of the previous resolution. For the remaining vertices, we initialize the transformation parameters by minimizing the energy E_{sm} with respect to the indices i that are not present in resolution level l. After the transformations are initialized, we compute the resolution r of the current level and minimize E_{track}.

This optimization scheme leads to a set of transformation parameters $\vec{t_i}, \vec{r_i}$, and ϕ_i that align T with F while aiming to maintain a smooth and intersection-free mesh.

5. Displacing Unobserved Vertices Using FEM

Consider the situation after T was deformed to frame F_j using the approach outlined in the previous section, and denote the deformation of T by T_{F_j}. We call the vertices p_i in T_{F_j} that were deformed using valid data observations *observed vertices*, and we call the remaining vertices *unobserved vertices*. Unobserved vertices were deformed using smoothness assumptions and possibly unreliable data observations. This section describes how to displace the unobserved vertices using a linear FEM.

First, we discuss how to find the observed vertices. An easy way is to consider all vertices as observed whose associated points $N(p_i)$ in the point cloud data have a corresponding weight ω_i equal to one. However, this approach has the problem that many vertices corresponding to points in a hole of the data may pick the same associated point, which leads to some observed vertices with poor assignments. To remedy this problem, we wish to only consider an observation as valid if few vertices of the template share the same associated point. Note that it is not straight forward to define "few", since the number of template vertices that share the same associated data point depends on both the data and the template resolution. We use the following heuristic to make the approach independent of global resolution changes. For each data point the number of vertices on the template that chose it as nearest neighbor is counted, and this count is averaged over all data points that were chosen by at least one vertex of the template. A template vertex p_i is considered an observed vertex if ω_i is equal to one and if the count corresponding to $N(p_i)$ does not exceed twice the average count.

We aim to reposition the unobserved vertices of T_{F_j} using a finite element model. We use $T_{F_{j-1}}$ with $T_{F_0} = T$ as start position for the FEM step and an offset vector from the start position $T_{F_{j-1}}$ to the current frame T_{F_j}. A tetrahedral mesh is used to compute the FEM. The initial tetrahedral mesh of T is obtained by tetrahedralizing a simplified version of T. This simplification is necessary to make the algorithm more time and space efficient. The tetrahedral mesh of T_{F_j} is obtained by simply updating the vertex positions of $T_{F_{j-1}}$ using the deformed mesh T_{F_j} and the FEM deformation from frame F_{j-1} to frame F_j. Let $T^{tet}_{F_{j-1}}$ denote the tetrahedral mesh.

The FEM linearly relates the offsets of the vertices and the forces applied to the tetrahedral mesh using a stiffness matrix $K(E,\nu)$ that depends on the geometry of the tetrahedral mesh and on two elasticity parameters, the Young's modulus E and the Poisson ratio ν. Let $\vec{f}$ denote the vector of forces applied to the vertices of the tetrahedral mesh and let $\vec{u}$ denote the vector of offsets of the vertices of the tetrahedral mesh. Both $\vec{f}$ and $\vec{u}$ have dimension $3m$, where m is the number of vertices of the tetrahedral mesh. Then,

$$K(E,\nu)\vec{u} = \vec{f}. \tag{4}$$

Recall that we know reliable offsets for the observed, but not for the unobserved vertices. Furthermore, we know the contact point and the direction of the force at the contact point and we know that the force at all of the remaining vertices is zero. We can normalize the length of the force direction at the contact point, since changing the length of $\vec{f}$ only scales E. If the elasticity parameters are known, it is possible to use this information to compute the missing offsets and force vectors by rearranging the linear system of equations [4]. The mapping between the tetrahedral mesh and the surface mesh can then be used to modify the unobserved vertices of $T^{tet}_{F_{j-1}}$ according to the FEM prediction.

We use the approach of Becker and Teschner [2] to find the elasticity parameters. We aim to minimize the energy $F = (K(E,\nu)\vec{u} - \vec{f})^2$ with respect to E and ν, where we only consider the points with known offsets and forces. We achieve this by initializing E to $1e6$ and ν to 0.45 and by solving the optimization problem.

Once the FEM step is completed for frame F_j, it remains to adjust the transformation parameters $\vec{t_i}, \vec{r_i}$, and ϕ_i to capture the new deformation. We achieve this by minimizing

$$\begin{aligned} E_{def} &= w_{data}E_{target} + w_{sm}E_{sm} + w_{md}E_{md} \text{ with} \\ E_{target} &= \textstyle\sum_i \|A_i\tilde{p}_i - TP(p_i)\|_2^2, \end{aligned} \tag{5}$$

where $TP(p_i)$ (in homogeneous coordinates) is the position of the point corresponding to vertex p_i on the deformed tetrahedral mesh $T^{tet}_{F_{j-1}}$. Note that we only optimize the energy with respect to parameters that influence unobserved vertices of T_{F_j}. We set $w_{sm} = 100$ and $w_{md} = 0.1$.

6. Implementation Details

The implementation of the algorithm is in C++ and uses a quasi-Newton method [14] for all of the optimization steps. For each optimization step, at most 1000 iterations are used. The tetrahedralization is computed using tetgen (http://tetgen.berlios.de). When tetrahedralizing the model, we find a high quality tetrahedralization by restricting the radius-edge ratio of each tetrahedron to be at most two.

This section discusses implementation details. First, we discuss the parameter settings used in the experiments. Second, we analyze the influence of the different energy terms used in the tracking energy.

6.1. Discussion of Parameters

The parameters w_{data}, w_{sm}, and w_{md} used during tracking give the relative weights between the different energy terms, the parameters d and α control which data points influence the data term, and the parameters s_{sm} and s_{md} influence the neighborhoods considered for the smoothing and repelling terms, respectively.

To make the relative influence of the weights w_{data}, w_{sm}, and w_{md} invariant with respect to scaling, we

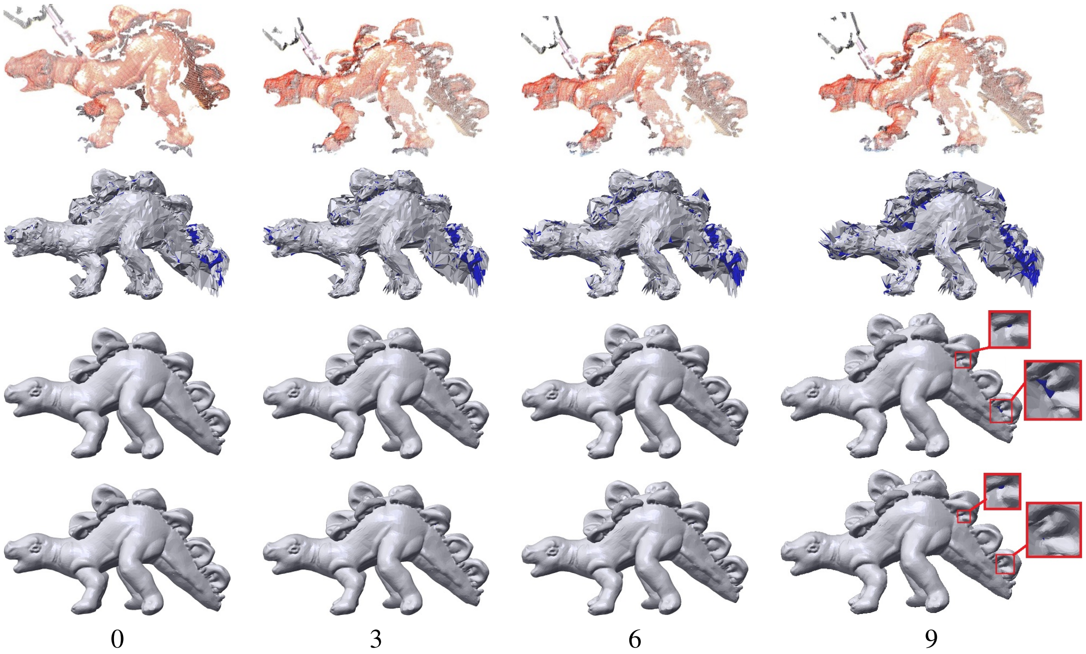


Figure 2. *Influence of energy terms. From top to bottom: data, result using E^{*}_{track}, result using E^{**}_{track}, result using E_{track}.*

pre-scale all of the input models, such that the length of the diagonal of the bounding box of the template model is one. This allows to set most of the parameters to one constant value for all experiments. The weight schedules used for w_{sm} and w_{md} are discussed in Sections 4 and 5. Furthermore, we set $w_{data} = 1, d = 5, \alpha = 60°$, and $s_{md} = 0.1$.

The parameter s_{sm} is the only parameter that is varied. This parameter gives the smoothing radius with respect to the resolution of the template mesh. It needs to be adjusted depending on the ratio between the mesh resolution and the mesh size. If the mesh resolution (measured as average edge length) is high compared to the size of the model, then s_{sm} can be set relatively low. If the mesh resolution is low compared to the size of the model, then s_{sm} needs to be set to a higher value. In our experiments, we set $s_{sm} = 3$ for synthetic data and $s_{sm} = 5$ for the dinosaur model.

6.2. Influence of Energy Terms

Figure 2 shows the results of the tracking step with different energy terms with the back side shown in blue. Specifically, results are shown for $E^{*}_{track} = w_{data}E_{data}$, $E^{**}_{track} = w_{data}E_{data} + w_{sm}E_{sm}$, and E_{track}. For these experiments, the weights are initialized and relaxed as discussed in Section 4. Note that when using E^{*}_{track}, the mesh already contains self-intersections after the first frame and the shape of the model is altered significantly after ten frames. The reason is that only using the data term results in an under-determined optimization problem. When using E^{**}_{track}, the overall shape of the model is preserved. However, after ten frames, the model contains self-intersections at the flaps on the tail of the model. With E_{track}, these problems are reduced.

7. Evaluation

This section discusses the datasets used in the experiments and shows a synthetic evaluation of the method as well as experiments based on real data. For all the experiments, the input models are pre-scaled, such that the length of the bounding box diagonal of the template model is one. This information on the scale of the models serves as reference for the numerical evaluations below.

7.1. Input Data

Synthetic Data. The synthetic datasets, shown in Fig. 3, are created using the buste and duck models from the AIM@Shape repository (http://shapes.aimatshape.net). We create synthetic deformations of the models by applying different finite element deformations to the models with GetFEM (http://download.gna.org/getfem/). The shapes are deformed using a linear FEM and using the incompressible non-linear Saint Venant-Kirchhoff model (StVK). The back sides of the deformed models are removed and the remaining front sides are used as input to the algorithm. The complete undeformed model is used as template. In our simulations, the head of the buste is pushed to the left and the base of the neck of the duck is pushed down and forward. Table 1 shows the deformation parameters.

Stereo Data. We acquire the stereo observations of each frame using a commercial machine vision stereo camera (Point Grey Bumblebee 2) and its matching library. Fig. 4

Model	Number Vertices	Number Frames	λ $(\mathrm{N/mm}^2)$	μ $(\mathrm{N/mm}^2)$	Lagrange Multiplier
Buste	7470	10	$5.185e4$	$2.222e4$	$1e4$
Duck	1246	25	$5.000e4$	$1.000e3$	$1e4$

Table 1. *Information about the synthetic models.*

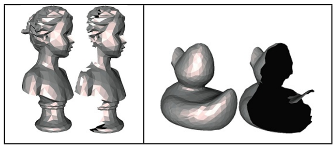


Figure 3. *Buste and duck models: model and front side.*

shows the template mesh along with a typical data frame containing noise that is typical for passive stereo, such as noise along the viewing direction and missing data due to occlusions. Occlusions are especially visible in the areas of the flaps. Furthermore, the input data contains points located on a tag at the tail of the model, which is not part of the template. We track a dinosaur plush toy based on stereo data while the head is pushed towards the floor.

Range Data. We acquire range data using a Kinect sensor and the point cloud library. With this setup, we acquire a similar deformation of the same dinosaur plush toy as with the stereo camera. The resolution of this type of data is low compared to our stereo data, as shown in Fig. 4.

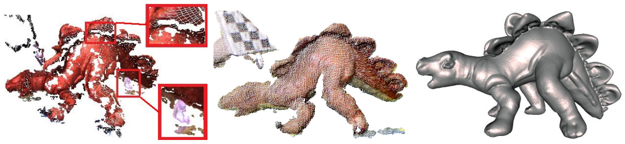


Figure 4. *Left to right: noisy stereo frame, Kinect frame, template.*

7.2. Evaluation of Robustness

Our first set of experiments aims to show that our approach is robust with respect to noise. The synthetic buste data generated with a linear FEM is used in this experiment. We consider three types of noise. First, we aim to model outliers in a way that simulates the outliers commonly present in stereo data. To model these outliers, we pick a viewpoint for the model. A vertex p_i is perturbed as $p_i + x\vec{v}_i$, where $\vec{v}_i$ is the unit vector pointing from p_i to the viewpoint and x is a uniformly distributed random number in the range $[-r, 4r]$, and r is the resolution of the model. Each vertex of the model is perturbed with probability 1/50. Second, we perturb the vertices of the input data by adding Gaussian noise in the vertices normal direction. The variance of the Gaussian is 75% of the bounding ball radius of the model. Third, we evaluate the influence of the resolution of the input data on the results by subdividing the input data using one step of Loop subdivision.

As the complete ground truth model is known, the per-vertex distance for a frame can be computed as the distance between the vertex position in the tracking result and its corresponding point in the ground truth model. The error for a frame can then be computed as the average or maximum over all per-vertex distances of this frame. The resulting errors are given in Fig. 5. The average distances are small compared to the size of the model and the increase in the distances caused by the presence of outliers, Gaussian noise, and an increase in resolution is insignificant.

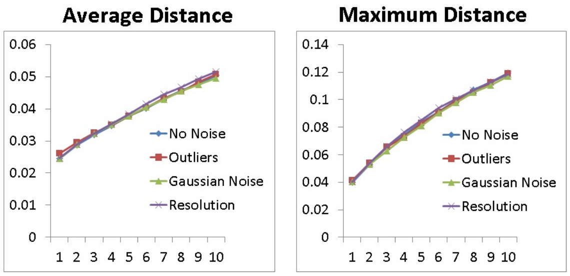


Figure 5. *Error measured as average and maximum distance over all vertices. X-axis: frame number. Y-axis: distances (for reference: normalized mesh diagonal of length one).*

7.3. Evaluation of FEM Correction

Our second set of experiments shows that the FEM step improves the shape of the unseen side of the model. We show that displacing the unobserved vertices using a linear FEM in every deformation step yields satisfactory results even for synthetic deformations that were generated using a non-linear FEM. We consider deformation sequences generated using linear FEM and the StVK model. For each deformation sequence, tracking is computed with and without FEM. We then evaluate the errors as above. However, instead of considering all vertices of the model for the error computation, we only consider vertices on the unseen side.

The first experiment uses the buste data. This dataset exhibits a deformation that affects the global shape of the model. We found that the FEM step yields an improvement of the average and maximum errors over the approach without the FEM step in most frames. Fig. 6 visualizes the distances to the ground truth for the two deformation sequences. The result without FEM has a large error in the back of the head, which is almost completely flattened, while the result with FEM has a significantly lower error in this area. Linear and non-linear deformations are tracked equally well. A similar experiment for a hand dataset is shown in the supplementary material.

The second experiment uses the duck data and is shown in Figure 7. The figure shows the result of applying a linear FEM deformation to the duck model and our result. The global linear FEM aims to linearize the global deformation that is observed when applying a non-linear FEM to the model. Observe that applying a linear FEM leads to unrealistic artifacts at the back of the head of the duck because the applied forces cause a rotation of the head. Recall that our

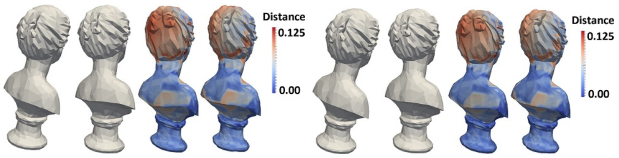
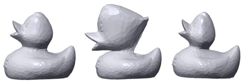


Figure 6. *Left: linear FEM dataset. Right: StVK dataset. Left to right: ground truth start and end positions, result without FEM, and result with FEM. Color coding shows distance to ground truth.*



	Volume	Area
Ground Truth	0.060784	1.027610
Global FEM	0.075405	1.267665
Our Method	0.056388	0.988828

Figure 7. *Duck results (end position). Left to right: ground truth, result computed using a global linear FEM, our result (with FEM). Volume and area are measured on the model with normalized size.*

method uses a linear FEM to predict the unobserved side of the model. However, since our method linearizes the deformation locally at each frame, no unrealistic artifacts occur. Furthermore, our approach is significantly closer to the ground truth than using a global linear FEM in terms of mesh volume and area.

7.4. Evaluation on Stereo and Range Data

In the following two experiments, we aim to track the dinosaur plush toy based on either stereo data or range data while the head of the toy is being pushed towards the floor. Fig. 8 shows the observed side of the deformed toy. The figure shows the data, the deformed template, and the signed distance between the deformed template and the data. In the visualization of the signed distance, points that do not have a valid nearest neighbor in the data are shown in red. Note that results of similar quality are obtained for data with significantly different resolution and noise levels.

Second, for the stereo dataset, we analyze the influence of the FEM step on the final result. Recall that by design, the FEM step influences the unobserved side of the model in each frame. Fig. 9 shows the unobserved side at the tail of the deformed model after 11 frames with and without the FEM step. Note that the FEM step reduces the amount of crumpling at the tail. This results in a better initialization for the next frame. Fig. 1 shows the results with and without the FEM step after 26 frames. Note that the result with FEM undergoes a larger overall deformation than the result without FEM. This discrepancy is due to the improved initialization caused by FEM in each frame.

7.5. Limitations

Our algorithm has some limitations. We employ a non-rigid iterative closest point algorithm to fit the template to the data. Hence, if the initial alignment is poor or if there are large occluded areas in the data, the algorithm cannot deform the template accurately. Fig. 10 illustrates an example, where one flap of the dino model is pushed towards the spine. The initial alignment of the template and the data is poor and large parts of the flap are occluded. Due to these issues, our algorithm does not correctly track the movement of the flap towards the spine.

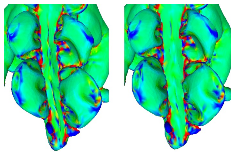


Figure 9. *Result color-coded with Gaussian curvature. Left: result using tracking only. Right: result with FEM. Note the improvements along the spine.*

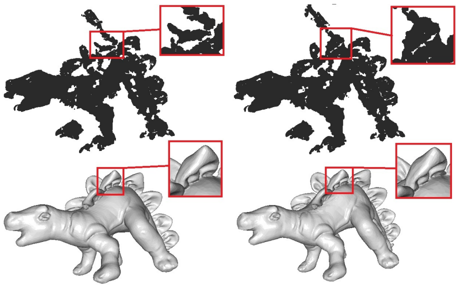


Figure 10. *Problems occur in areas with occlusion.*

While our method discourages self-intersections of the mesh using energy terms, it does not guarantee that self-intersections are avoided. If self-intersections occur, the quality of the updated FEM mesh may be low. Finally, our tracking method requires a number of input parameters, however, for all our examples, we use a fixed set of parameters except for s_{sm}, which we set based on mesh resolution.

8. Conclusions

We proposed an approach to track the geometry of a surface over time from a set of input point clouds captured from a single viewpoint. We combine the use of a template and the use of a linear finite element method to track the model. By linearizing the deformation at each frame, we show that we can accurately track surfaces that deform in a non-linear fashion. We demonstrate the robustness of our approach with respect to noise using a synthetic evaluation and using real data captured with a stereo setup and with a Kinect.

Acknowledgements

We thank Motahareh Teki for help with the pointcloud acquisition. This work has partially been funded by NSERC Canada and the Cluster of Excellence *MMCI*.

References

[1] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from

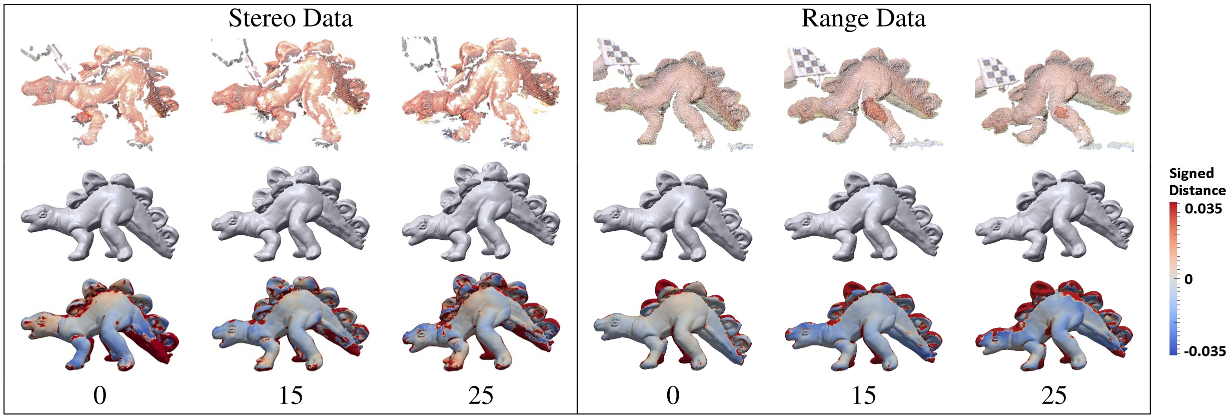


Figure 8. *Observed side of dinosaur. From top to bottom: data, result, and signed distance between result and data.*

range scans. *TOG 22(3)* (2003) 587–594.

[2] Becker M., Teschner M.: Robust and efficient estimation of elasticity parameters using the linear finite element method. In *SimVis* (2007) 15–28.

[3] Bickel B., Bächer M., Otaduy M., Matusik W., Pfister H., Gross M.: Capture and modeling of non-linear heterogeneous soft tissue. *TOG 29(3)* (2009) 89:1–9.

[4] Bro-Nielsen M.: Finite element modeling in surgery simulation. *Proc. of the IEEE 86* (1998) 490–503.

[5] Cagniart C., Boyer E., Ilic S.: Free-from mesh tracking: a patch-based approach. In *CVPR* (2010) 1339–1346.

[6] Choi J., Szymczak A.: Fitting Solid Meshes to Animated Surfaces using Linear Elasticity. *TOG 28(1)* (2009) 6:1–10.

[7] de Aguiar E., Stoll C., Theobalt C., Ahmed N., Seidel H.-P., Thrun S.: Performance capture from sparse multi-view video. *TOG 27(3)* (2008) 98:1–10.

[8] Furukawa Y., Ponce J.: Dense 3d motion capture from synchronized video streams. In *CVPR* (2008) 1–8.

[9] Garland M., Heckbert P. S.: Surface simplification using quadric error metrics. In *Int. Conf. Comput. Graph. Interact. Techniques* (1997) 209–216.

[10] Lang J., Pai D.: Estimation of elastic constants from 3d range-flow. In *3DIM* (2001) 331–338.

[11] Li H., Adams B., Guibas L. J., Pauly M.: Robust single-view geometry and motion reconstruction. *TOG 28(5)* (2009) 175:1–10.

[12] Li H., Luo L., Vlasic D., Peers P., Popovic J., Pauly M., Rusinkiewicz S.: Temporally coherent completion of dynamic shapes. *TOG 31(1)* (2012) 2:1–11.

[13] Liao M., Zhang Q., Wang H., Yang R., Gong M.: Modeling deformable objects from a single depth camera. In *ICCV* (2009) 167–174.

[14] Liu D. C., Nocedal J.: On the limited memory method for large scale optimization. *Math. Prog. 45* (1989) 503–528.

[15] Mitra N. J., Flory S., Ovsjanikov M., Gelfand N., Guibas L., Pottmann H.: Dynamic geometry registration. In *SGP* (2007) 173–182.

[16] Nguyen T., Boyce B.: An inverse finite element method for determining the anisotropic properties of the cornea. *Biomech. Model. in Mechanobiol. 10(3)* (2011) 323–337.

[17] Popa T., South-Dickinson I., Bradley D., Sheffer A., Heidrich W.: Globally Consistent Space-Time Reconstruction. *SGP* (2010) 1633–1642.

[18] Schnabel J., Tanner C., Castellano-Smith A., Degenhard A., Leach M., Hose R., Hill D., Hawkes D.: Validation of nonrigid image registration using finite-element methods: Application to breast MR images. *IEEE Trans. Med. Imag. 22(2)* (2003) 238–247.

[19] Sharf A., Alcantara D., Lewiner T., Greif C., Sheffer A., Amenta N., Cohen-Or D.: Space-time surface reconstruction using incompressible flow. *TOG 27(5)* (2008) 110:1–10.

[20] Syllebranque C., Boivin S.: Estimation of mechanical parameters of deformable solids from videos. *Vis. Comp. 24* (2008) 963–972.

[21] Terzopoulos D.: Multi-level Reconstruction of Visual Surfaces: Variational Principles and Finite Element Representations. MIT TR 671 (1982).

[22] Tevs A., Berner A., Wand M., Ihrke I., Bokeloh M., Kerber J., Seidel H.-P.: Animation Cartography - Intrinsic Reconstruction of Shape and Motion. *TOG 31(2)* (2012) 12:1–15.

[23] Tung T., Matsuyama T.: Dynamic surface matching by geodesic mapping for 3d animation transfer. In *CVPR* (2010) 1402–1409.

[24] van Kaick O., Zhang H., Hamarneh G., Cohen-Or D.: A survey on shape correspondence. *Eurograph. STAR* (2010) 1–24.

[25] Vlasic D., Baran I., Matusik W., Popovic J.: Articulated Mesh Animation from Multi-view Silhouettes. *TOG 27(3)* (2008) 97:1–10.

[26] Wand M., Jenke P., Huang Q., Bokeloh M., Guibas L., Schilling A.: Reconstruction of deforming geometry from time-varying point clouds. *SGP* (2007) 49–58.

[27] Zheng Q., Sharf A., Tagliasacchi A., Chen B., Zhang H., Sheffer A., Cohen-Or D.: Consensus skeleton for non-rigid space-time registration. *CGF 29(2)* (2010) 635–644.