Capturing 3D Stretchable Surfaces from Single Images in Closed Form *

Francesc Moreno-Noguer
Institut de Robòtica i
Informàtica Industrial (CSIC-UPC)
08028 Barcelona, Spain

Mathieu Salzmann
EECS & ICSI
UC Berkeley
CA 94720

Vincent Lepetit
EPFL - CVLab
1015 Lausanne
Switzerland

Pascal Fua
EPFL - CVLab
1015 Lausanne
Switzerland

Abstract

We present a closed-form solution to the problem of recovering the 3D shape of a non-rigid potentially stretchable surface from 3D-to-2D correspondences. In other words, we can reconstruct a surface from a single image without a priori knowledge of its deformations in that image.

State-of-the-art solutions to non-rigid 3D shape recovery rely on the fact that distances between neighboring surface points must be preserved and are therefore limited to inelastic surfaces. Here, we show that replacing the inextensibility constraints by shading ones removes this limitation while still allowing 3D reconstruction in closed-form.

We demonstrate our method and compare it to an earlier one using both synthetic and real data.

1. Introduction

Capturing the shape of deformable 3D surfaces from a single image remains an open problem with an endless list of potential applications in computer vision and graphics. The main challenge comes from the fact that monocular 3D shape recovery is severely under-constrained. A common approach to overcoming this is to introduce deformation models. They can be either physically-based [1, 3, 11, 12, 13, 18] or learned from training data [2, 4, 10, 17]. In all these methods, surface deformations are expressed in terms of the model parameters, which are first initialized and then refined by minimizing an image-based objective function. Since this function typically has many local minima, good initialization is both critical and difficult to achieve.

This problem has been addressed recently in [6, 14, 16]. These papers propose approaches to 3D shape recovery in a single input image given a reference image in which the shape is known, and correspondences between the input and reference images. They do not require any knowledge of the deformations other than the fact that the surface is inextensible and that the distances between neighboring surface points must remain constant. This is a valid assumption for inelastic materials such as paper or cardboard, but rules out the reconstruction of elastic materials that can stretch.

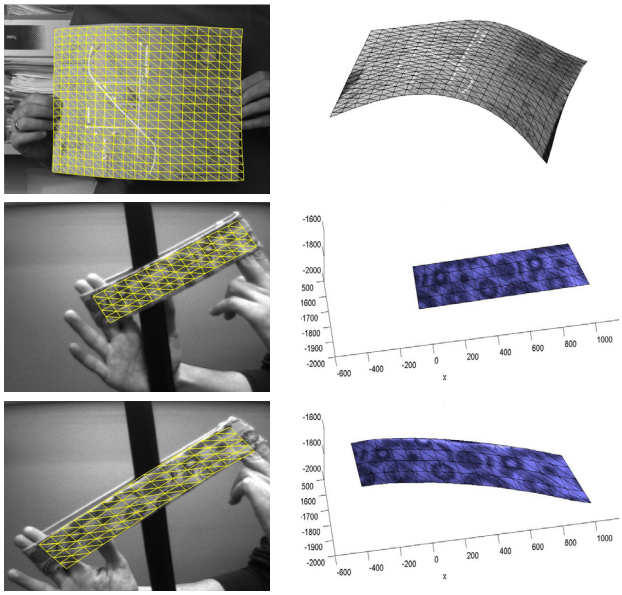


Figure 1. 3D Reconstruction of non-rigid inelastic (top) and elastic (center and botton) surfaces in closed form. In addition to correspondences, we use shading constraints and estimate the direction of the main light source.

In this paper, we show that replacing the constant-length constraints by shading constraints removes this limitation while allowing a closed-form solution to the 3D shape-recovery problem, as in [16]. More specifically, we rely on correspondences between the input and reference images and express the deformations as linear combinations of modes. In addition, we constrain the intensities of selected surface patches in the input and reference images to be related through a Lambertian reflectance model. This yields a set of cubic equations in terms of the mode weights and of the lighting parameters, which we solve through linearization. As shown in Fig. 1, this lets us reconstruct both elastic and inelastic objects in closed-form and simultaneously es-

*This work has been partially funded by the Spanish Ministry of Science and Innovation under projects 200850I055, DPI2008-06022, and Consolider Ingenio 2010 CSD2007-00018, by EU PACO-PLUS project FP6-2004-IST-4-27657, and by the Swiss National Science Foundation.

timate the light direction and intensity. Furthermore, using two different sources of image information –keypoint locations and shading– increases the robustness in ambiguous situations.

In the remainder of the paper, we first discuss related work. We then introduce our formulation and derive our systems of linear and cubic equations. Finally we compare our method against [16] both on synthetic and real data.

2. Related Work

Recovering the 3D geometry of a non-rigid surface from single images requires prior knowledge of its properties to turn an under-constrained problem into a tractable one.

Traditional shape-from-shading [8] and shape-from-texture [22] techniques do this by imposing surface smoothness and assuming that the surface either is Lambertian with known albedo or exhibits statistically homogeneous texture patterns. There has been many attempts over the years at relaxing these constraints but most state-of-the-art methods still require very strong assumptions that can only rarely be satisfied. [20, 23] are representative of current single-image approaches that refine both shape and illumination parameters. Even though these methods can return accurate estimates of both, their iterative nature means that, unlike our approach, they require good initial guesses. Furthermore, they are not designed to handle materials that can stretch. The idea of overcoming ambiguous situations by combining texture and shading cues was introduced in [21]. This approach, however, involves multiple iterative stages and explicitly penalizes stretching, which precludes accurate modeling of elastic surfaces.

Another approach to making the problem tractable is to introduce surface deformation models. Physically-based approaches introduce global models such as superquadrics [12], triangulated surfaces [7] or thin-plate splines [11]. Modal analysis [5, 13] has also been proposed to reduce the dimensionality of the problem. However, while these methods have been successful for retrieving smoothly deforming objects, they cannot capture the physics of complex deformations, which requires much more sophisticated and difficult to handle non-linear models [1, 18]. This has been recently addressed in a data-driven manner by using machine learning methods to build deformation models from collections of deformed shapes [2, 4, 10], or, for relatively small deformations, directly from sequences of images [17, 19].

In any event, in all the approaches discussed above, model parameters must first be initialized and then refined by minimizing an image-based objective function, which may have many local minima. In frame-to-frame tracking, the shape parameters found in a frame can serve as initial values for the following one, but this kind of approach still requires parameters to be specified in the first frame and cannot recover from a tracking failure. To avoid this, one must be able to recover the 3D shape without an initial estimate. This issue has been addressed in three recent papers [6, 14, 16] that all rely on the fact that distances between surface points must remain constant—a valid assumption for inelastic materials but not stretchable ones.

In short, our approach differs from previous techniques in that it can reconstruct a surface whether it stretches or not. Furthermore, the shape is recovered in closed-form, which implies that no initial estimate is needed.

3. Elastic and Inelastic Surface Reconstruction in Closed Form

In this section, we first use the formalism of [16] to show that the solution of our problem can be expressed as a linear combination of singular vectors corresponding to the small eigenvalues of a matrix. This matrix is derived from the point correspondences between the input and reference image. We then show that shading constraints can be expressed in terms of cubic polynomials involving the coefficients of the linear combination and the shading parameters. Finally, we solve the resulting system of cubic equations to compute both the ones and the others.

3.1. Initial Assumptions

We represent the surface as a triangulated 3D mesh whose shape is given by the vector $\mathbf{x} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_{n_v}^T]^T$ of its vertex coordinates, where $\mathbf{v}_i = [x_i, y_i, z_i]^T$. In the following, we assume that the mesh we use, like those of Fig. 1, has a rectangular topology and therefore that all mesh facets have one 90 degree angle. As will be discussed below, we could also use hexagonal meshes made of equilateral triangles, which can be used to model surfaces of arbitrary topology.

We seek to retrieve $\mathbf{x}$ in an input image, assuming that we are given

1. The shape of the mesh in a *reference configuration*, and n_c correspondences between a set of 3D points $\mathbf{p}_i$ on this mesh and 2D image locations $\mathbf{u}_i$.
2. An albedo value ρ_i for each point $\mathbf{p}_i$, which can be taken as the intensity of the corresponding pixel in the reference image if it was lit by a diffuse light source.
3. The internal calibration matrix $\mathbf{A}$ of the camera.
4. A mean shape $\mathbf{x}_0$ and a set of n_m deformation modes $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_{n_m}]$, representing the linear subspace of feasible mesh deformations.

We also assume the surface to be Lambertian and illuminated by a single point light source, whose direction and intensity are unknown. In the results section we will show that this assumption can be relaxed in practice and that our algorithm still yields good results in the presence of an extended light source.

3.2. Linear Geometric Constraints

As in [16], we start by showing that $\mathbf{x}$ can be expressed as the solution to a linear system encoding the 3D-to-2D correspondences equations. We express each point $\mathbf{p}_i$ as a function of the barycentric coordinates of the triangular face it belongs to and write

$$\forall i \ , \ \mathbf{p}_i = \sum_{j=1}^{3} a_{ij} \mathbf{v}_j^{[i]} \ , \tag{1}$$

where the a_{ij} are the homogeneous barycentric coordinates and $\{\mathbf{v}_j^{[i]}\}_{j=\{1,2,3\}}$ are the vertices of the face containing the point $\mathbf{p}_i$. Without loss of generality, we express the 3D points $\mathbf{p}_i$ in the camera referential, and their 2D projections $\mathbf{u}_i = [u_i, v_i]^T$ as

$$\forall i \ , \ w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{A}\mathbf{p}_i = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^{3} a_{ij} \begin{bmatrix} x_j^{[i]} \\ y_j^{[i]} \\ z_j^{[i]} \end{bmatrix} \tag{2}$$

where the w_i are the scalar projective parameters, $[x_j^{[i]}, y_j^{[i]}, z_j^{[i]}]^T$ the 3D coordinates of each vertex $\mathbf{v}_j^{[i]}$, and f_u, f_v and (u_c, v_c) the focal lengths and principal point of the calibration matrix $\mathbf{A}$.

From the last row of Eq. 2, the projective parameters can be written as $w_i = \sum_{j=1}^{3} a_{ij} z_j^{[i]}$. When substituted back into the first two rows we get for each 3D-to-2D correspondence

$$\sum_{j=1}^{3} a_{ij} f_u x_j^{[i]} + a_{ij}(u_c - u_i) z_j^{[i]} \quad = \quad 0 \ , \tag{3}$$

$$\sum_{j=1}^{3} a_{ij} f_v y_j^{[i]} + a_{ij}(v_c - v_i) z_j^{[i]} \quad = \quad 0 \ . \tag{4}$$

These equations can be jointly expressed for all the n_c correspondences as a linear system

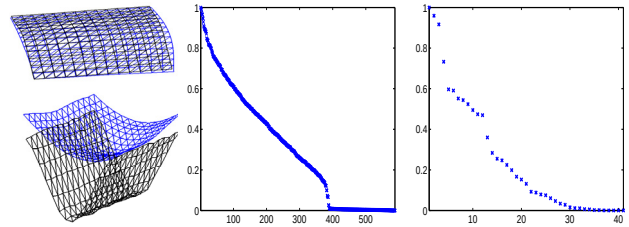
$$\mathbf{M}\mathbf{x} = \mathbf{0} \ , \tag{5}$$

where $\mathbf{M}$ is a $2n_c \times 3n_v$ matrix, made of the known coefficients of Eqs. 3 and 4.

As observed by [15] the matrix $\mathbf{M}$ is rank deficient even for a large number of correspondences n_c, that is, a solution $\mathbf{x}$ yielding a correct reprojection is not guaranteed to have a correct 3D shape. Fig. 2(a) illustrates this. One consequence of this is that matrix $\mathbf{M}$ has a large number of eigenvalues close to zero, as seen in Fig. 2(b). Therefore, additional constraints have to be introduced to reduce these ambiguities. This can be done by introducing deformation modes and representing the surface as a linear combination of $n_m << n_v$ basis shapes, which can be written as

$$\mathbf{x} = \mathbf{x}_0 + \sum_{i=1}^{n_m} \alpha_i \mathbf{q}_i = \mathbf{x}_0 + \mathbf{Q}\boldsymbol{\alpha} \ , \tag{6}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{n_m}]^T$ are the unknown weights of the basis shapes we want to recover. If we introduce this expression into Eq. 5, the linear system becomes



(a) View ambiguity (b) Eigenvalues of $\mathbf{M}$ (c) Eigenvalues of $\tilde{\mathbf{M}}$

Figure 2. (a) View ambiguity. The two plots correspond to the same configuration of the meshes seen from different viewpoints. (b) Eigenvalues of the matrix $\mathbf{M}$, for the black mesh of (a).(c) Eigenvalues of $\tilde{\mathbf{M}}$, after considering 40 deformation modes.

$$\begin{bmatrix} \mathbf{M}\mathbf{Q} & \mathbf{M}\mathbf{x}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ 1 \end{bmatrix} = \mathbf{0} \ . \tag{7}$$

A standard way of computing the deformation modes is to apply Principal Component Analysis to a large set of training shapes, and hence, each basis shape $\mathbf{q}_i$ is associated to a prior standard deviation value σ_i. We use this prior as a regularization term on the weights $\boldsymbol{\alpha}$ by minimizing $\mathbf{S}\boldsymbol{\alpha}$, where $\mathbf{S}$ is an $n_m \times n_m$ diagonal matrix with elements σ_i^{-1}. The shape is then obtained by solving

$$\tilde{\mathbf{M}} \begin{bmatrix} \boldsymbol{\alpha} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{Q} & \mathbf{M}\mathbf{x}_0 \\ \mathbf{S} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ 1 \end{bmatrix} = \mathbf{0} \tag{8}$$

in the least-squares sense. Note from Fig. 2(b) that this linear system has much fewer eigenvalues that are close to zero than the original one.

The solution of Eq. 8 belongs to the null space of $\tilde{\mathbf{M}}$. We take it to be

$$\begin{bmatrix} \boldsymbol{\alpha} \\ 1 \end{bmatrix} = \sum_{i=1}^{N} \beta_i \mathbf{m}_i \ , \tag{9}$$

where $\mathbf{m}_i$ are the right-singular vectors of $\tilde{\mathbf{M}}$ corresponding to its N smallest singular values. Picking the correct value for N is non-trivial since it amounts to deciding which of the singular values are small enough to be considered as being effectively zero. This is illustrated by Fig.2(c), where the number of small eigenvalues is around 10 and it is difficult to decide on an exact number. In practice, we run the algorithm for all $N \leq N_{max}$, where N_{max} is intentionally too large, and retain the result that yields the smallest average reprojection error. In all experimental results presented in this paper, we use $N_{max} = 15$. As will be discussed below, we chose the β_i by solving a set of cubic equations in closed form. And, since this is only a fraction of the overall computation, performing it several times only represents a small computational overhead.

3.3. Cubic Shading Constraints

Solving our shape reconstruction problem therefore amounts to picking the right β_i coefficients for the linear

combination of Eq. 9. In [16], this was done by choosing them so as to preserve the length of the mesh edges, which precludes the accurate modeling of a stretchable surface. Here, we remove this limitation by replacing length constraints by shading ones.

Let us first consider a single facet f with vertices $\{\mathbf{v}_i = [x_i, y_i, z_i]^T\}_{i=\{1,2,3\}}$ lit by a distant point light source, with unit direction $\mathbf{l} = [l_x, l_y, l_z]^T$ and intensity L. Let I be the observed intensity at a facet point of albedo ρ. Assuming a Lambertian reflectance model, we have

$$I = \rho L(\mathbf{l} \cdot \mathbf{n}) \ , \tag{10}$$

where $\mathbf{n}$ is the facet normal. In the following, we show that Eq. 10 can be written as a cubic equation in the unknowns β_i, L, l_x, l_y, and l_z. Since we can write such a constraint for each of our n_c correspondences, this yields a system of n_c cubic equations that we can solve using linearization.

3.3.1 Quadratic Representation of the Normal Vector

Let $\mathbf{v}_{12} = \mathbf{v}_1 - \mathbf{v}_2$ and $\mathbf{v}_{13} = \mathbf{v}_1 - \mathbf{v}_3$. The facet normal $\mathbf{n}$ can be computed as

$$\mathbf{n} = \frac{\mathbf{v}_{12} \times \mathbf{v}_{13}}{||\mathbf{v}_{12} \times \mathbf{v}_{13}||} = \frac{1}{2Area(f)}[\tilde{n}_x, \tilde{n}_y, \tilde{n}_z]^T \tag{11}$$

where $Area(f)$ is the area of the triangular facet f and

$$\begin{aligned} \tilde{n}_x &= y_2z_3 - y_2z_1 - y_1z_3 - z_2y_3 + z_2y_1 + z_1y_3 \ , \\ \tilde{n}_y &= z_2x_3 - z_2x_1 - z_1x_3 - x_2z_3 + x_2z_1 + x_1z_3 \ , \\ \tilde{n}_z &= x_2y_3 - x_2y_1 - x_1y_3 - y_2x_3 + y_2x_1 + y_1x_3 \ . \end{aligned} \tag{12}$$

The system of Eq. 12 is quadratic in the vertex coordinates, and therefore also in the β_i. More specifically, from Eq. 9 we can write $\alpha_i = \sum_{j=1}^{N} \beta_j \mathbf{m}_j^{[i]}$, where $\mathbf{m}_j^{[i]}$ is the $i-th$ element of the vector $\mathbf{m}_j$. From Eq. 6 we then have

$$\mathbf{x} = \mathbf{x}_0 + \sum_{i=1}^{n_m} \sum_{j=1}^{N} \beta_j \mathbf{m}_j^{[i]} \mathbf{q}_i \ , \tag{13}$$

which lets us write

$$\begin{aligned} \tilde{n}_k &= \gamma_0^{[k]} + \sum_{i=1}^{N} \gamma_i^{[k]} \beta_i + \sum_{i=1}^{N} \sum_{j=i}^{N} \gamma_{ij}^{[k]} \beta_i \beta_j \\ &= (\boldsymbol{\gamma}_{dir,k})^T \cdot \begin{bmatrix} \boldsymbol{\beta} \\ 1 \end{bmatrix} \ , \end{aligned} \tag{14}$$

where $k = \{x, y, z\}$ and the coefficients $\gamma_i^{[k]}$ and $\gamma_{ij}^{[k]}$ are generated by arranging the appropriate components of the vectors $\mathbf{x}_0$, $\{\mathbf{q}_i\}_{i=1,\ldots,n_m}$ and $\{\mathbf{m}_j\}_{j=1,\ldots,N}$, all of which are known. In the right part of Eq. 14, $\tilde{n}_k$ is written as the dot product of two vectors, $\boldsymbol{\gamma}_{dir,k}$ which is made of known coefficients and

$$\boldsymbol{\beta} = [\beta_1, .., \beta_N, \beta_1\beta_1, .., \beta_1\beta_N, \beta_2\beta_2, .., \beta_2\beta_N, .., \beta_N\beta_N]^T \ , \tag{15}$$

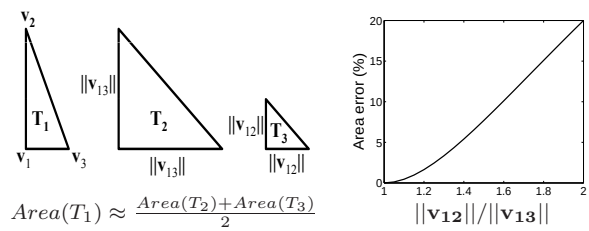


Figure 3. Approximating the area of a right triangle. **Left:** In order to be able to write the magnitude of the facets norm as a quadratic function of the coefficients β_i we have to approximate its area as the mean area of two isosceles triangles. Of course, the approximation is perfect for isosceles triangles and becomes poorer the bigger is the difference between the sides$||\mathbf{v_{12}}||$ and $||\mathbf{v_{13}}||$ of the triangle. **Right:** Error in the estimation of the triangle area, as a function of the ratio of the sides lengths.

which contains the unknown. In other words, the numerator of Eq. 11 can be written as quadratic polynomials in the β_i. By contrast, exactly computing the denominator would require evaluating square roots of 4-degree polynomials in the β_i coefficients, which would make it impossible to solve the resulting system of equations by simple linearization. We overcome this difficulty by replacing the exact value of $||\mathbf{v}_{12} \times \mathbf{v}_{13}||$ by an approximate one that depends on the fact that individual triangles have one 90 degree angle. This allows us to replace Eq. 11 by a pair of equations expressed in terms of a quadratic polynomial in the β_i's, as follows.

Without loss of generality, let us number the vertices of the facet as $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$, with $\mathbf{v}_1$ being at the 90 degree angle. We therefore have

$$||\mathbf{v}_{12}||^2 + ||\mathbf{v}_{13}||^2 = ||\mathbf{v}_{23}||^2 \ . \tag{16}$$

Each $||\mathbf{v}_{ij}||^2$ is quadratic in terms of the vertex coordinates, and, given Eq. 13, also in the β_i's. Using the same notation as in Eq. 14, constraining the angle of a single facet to be 90 degrees can be written as

$$(\boldsymbol{\gamma}_{right})^T \cdot \begin{bmatrix} \boldsymbol{\beta} \\ 1 \end{bmatrix} = \mathbf{0} \ , \tag{17}$$

where $\boldsymbol{\gamma}_{right}$ is again computed by arranging specific elements of the known vectors $\mathbf{x}_0$, $\{\mathbf{q}_i\}_{i=1,\ldots,n_m}$ and $\{\mathbf{m}_j\}_{j=1,\ldots,N}$, according to the quadratic monomials generated when expanding the terms $||\mathbf{v}_{ij}||^2$.

Furthermore, the facet area of a right triangle is given by

$$Area(f) = \frac{||\mathbf{v}_{12}|| \cdot ||\mathbf{v}_{13}||}{2} \ . \tag{18}$$

Since directly using this would yield polynomials of degree higher than 2, we approximate it by

$$\begin{aligned} Area(f) &\approx \frac{1}{2}\left(\frac{||\mathbf{v}_{12}||^2}{2} + \frac{||\mathbf{v}_{13}||^2}{2}\right) \\ &= (\boldsymbol{\gamma}_{area})^T \cdot \begin{bmatrix} \boldsymbol{\beta} \\ 1 \end{bmatrix} \ , \end{aligned} \tag{19}$$

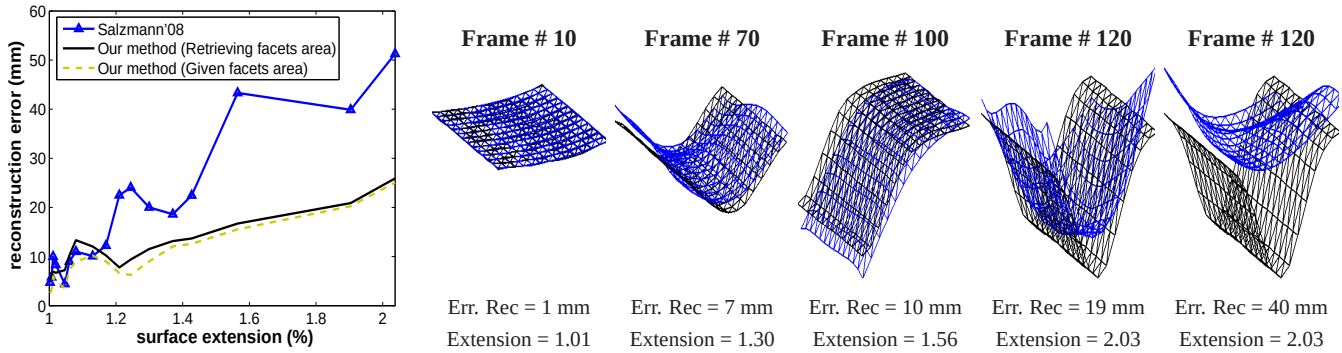


Figure 4. **Left:** Comparing the performance of our approach against the method of Salzmann et al. [16] in a synthetic experiment with a stretchable surface. We plot as well the results of our method if the true facet areas were given. The graph represents the mean 3D reconstruction error as a function of the surface extension. **Right:** Significance of the reconstruction error values. Black: true shapes. Blue: Shapes reconstructed using one the three methods and their associated error values.

where $\boldsymbol{\gamma}_{area}$ is known and computed as we did before for $\boldsymbol{\gamma}_{dir,k}$ and $\boldsymbol{\gamma}_{right}$. Fig. 3(left) illustrates the meaning of this approximation: The area of the right triangle is approximated as the mean area of two isosceles right triangles, one having two equal sides of length $||\mathbf{v}_{12}||$ and the other two equal sides of length $||\mathbf{v}_{13}||$. Note that we could use the same approximation for hexagonal meshes because equilateral triangles may be split into two isosceles right triangles. Fig. 3(right) shows the error produced by this approximation as a function of the ratio $||\mathbf{v}_{12}||/||\mathbf{v}_{13}||$. Note that this approximation is poorer if the stretching is produced just along one direction. For instance, as observed in the figure if one side is stretched to twice its initial length and the length of the other remains constant, the error of the approximated area will be around 20%. By contrast, if both sides are stretched more or less equally, the estimation error will be negligible. In Section 4 we will validate our approximation on experimental data and we will see that it is in fact very appropriate.

In short, imposing the constraints derived in Eqs. 14, 17, and 19 forces the normal of a facet to be of unit norm.

3.3.2 Integrating Lighting Unknowns

We are now in a position to integrate the expressions derived above for unit normals into the shading constraint of Eq. 10. Let $L\mathbf{l} = [L_x, L_y, L_z]^T$ be the lighting unknowns and $\tilde{I} = 2I/\rho$. Eq. 10 can be re-written as

$$Area(f)\tilde{I} = ([L_x, L_y, L_z] \cdot [\tilde{n}_x, \tilde{n}_y, \tilde{n}_z]^T) \tag{20}$$

If we expand this equation by considering the expressions of $\tilde{n}_x$, $\tilde{n}_y$, $\tilde{n}_z$, and $Area(f)$ derived in the previous section, we obtain

$$(\boldsymbol{\gamma}_{area})^T \begin{bmatrix} \boldsymbol{\beta} \\ 1 \end{bmatrix} \tilde{I} = (\boldsymbol{\gamma}_{dir,x})^T \cdot \begin{bmatrix} L_x\boldsymbol{\beta} \\ L_x \end{bmatrix} + (\boldsymbol{\gamma}_{dir,y})^T \cdot \begin{bmatrix} L_y\boldsymbol{\beta} \\ L_y \end{bmatrix} + (\boldsymbol{\gamma}_{dir,z})^T \cdot \begin{bmatrix} L_z\boldsymbol{\beta} \\ L_z \end{bmatrix} . \tag{21}$$

Note that we have one such equation for each 3D-to-2D correspondence. By grouping these equations for all n_c correspondences and introducing the right angle constraint of Eq. 17 for each of the n_f faces of the mesh, we obtain a system of the form

$$\begin{bmatrix} \mathbf{D} \\ \mathbf{R} \end{bmatrix} \mathbf{b} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} , \tag{22}$$

where $\mathbf{b} = [\boldsymbol{\beta}, L_x, L_y, L_z, \boldsymbol{\beta}L_x, \boldsymbol{\beta}L_y, \boldsymbol{\beta}L_z]^T$ is the vector of unknowns of size $2N(N+3)+3$, made of linear, quadratic and cubic terms that simultaneously contains the lighting and the geometric unknowns. $\mathbf{D}$ and $\mathbf{d}$ are an $n_c \times 2N(N+3)+3$ matrix and an n_c vector respectively, built from the known parameters $\tilde{I}$, $\boldsymbol{\gamma}_{dir,x}$, $\boldsymbol{\gamma}_{dir,y}$, $\boldsymbol{\gamma}_{dir,z}$ and $\boldsymbol{\gamma}_{area}$, for each correspondence. Finally, $\mathbf{R}$ is an $n_f \times N(N+1)/2$ matrix—expanded with zero columns to fit the dimension of $\mathbf{D}$—that accounts for the right angle constraints.

We use a simple linearization procedure to solve the system of Eq. 22, which means solving it as if it were a linear system where the quadratic and cubic terms are considered as new linear variables. Finally, the unknowns $\{\beta_1, \ldots, \beta_N\}$ and L_x, L_y and L_z are directly retrieved from the elements of $\mathbf{b}$ which were originally linear. The light intensity and direction are respectively computed as $L = ||[L_x, L_y, L_z]||$ and $\mathbf{l}^T = [L_x, L_y, L_z]/L$.

4. Results

In this section, we use both synthetic and real data to show that we can correctly retrieve the 3D shape of both inelastic and stretchable surfaces, which is in contrast to earlier techniques.

In all our experiments, we used the same deformation modes, automatically generated by performing Principal Component Analysis on a database of synthetically deformed meshes. The only parameters that change from one experiment to the next are the mesh sizes.

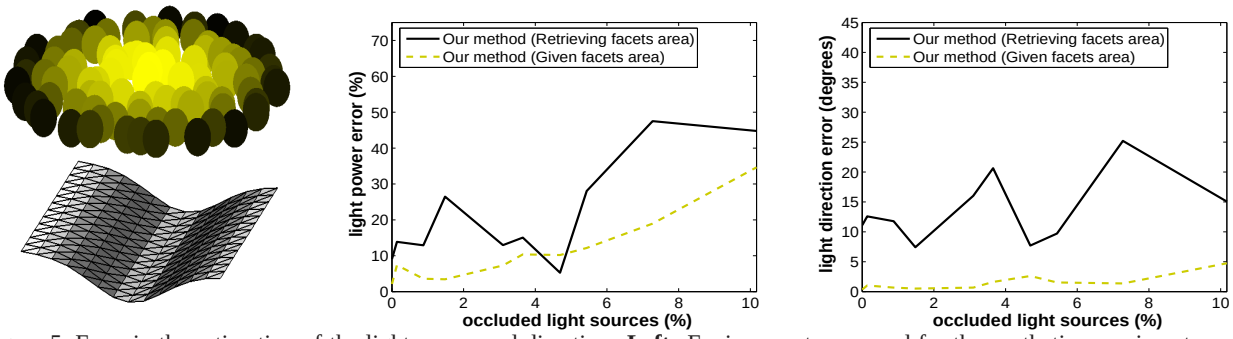


Figure 5. Error in the estimation of the light power and direction. **Left:** Environment map used for the synthetic experiment, made of a large number of point light sources with different intensity. **Center** and **Right:** Errors in the estimation of the light power and the mean direction, as a function of the percentage of occluded light sources.

4.1. Results on synthetic data

We synthesized a 120-frame sequence of a deforming 3D mesh with 14×14 vertices. Its initial configuration was a 100×100mm rectangle, which we used as a reference and progressively deformed according to a sinusoidal wave with translating phase and increasing amplitude, as shown in Fig. 4. Note that surface stretching increases with wave amplitude and that the area in the last frame is about twice that in the reference frame.

We then synthetically produced 100 random 3D-to-2D correspondences per frame, in a 640×480 image acquired using a virtual camera with focal length $f_u = f_v = 800$ and principal point at $(u_c, v_c) = (320, 240)$. A Gaussian noise with a 5 pixels standard deviation was added to the 2D point coordinates. We also computed the intensity of each image point assuming a Lambertian reflectance model and illuminating the surface using the lighting environment map of Fig. 5(left), which was made of 90 point light sources of different intensities distributed on the upper hemisphere. The effects of the cast and attached shadows were considered when computing the intensity to show that our approach can tolerate light sources that are *not* true point light sources.

The experiment was repeated 60 times per frame. Each time, we computed the 3D shape using [16], our method as described in Section 3, and a variant in which we use the correct value for the facet area $Area(f)$ in Eq. 19, known for synthetic data. The purpose of introducing this variant is to gauge the error resulting from replacing the true value of $Area(f)$ by its approximation, as discussed in Section 3.3.1. The graph of Fig. 4(left) summarizes the results of these experiments. We plot the mean reconstruction error as a function of the surface extension, which is the ratio between the true area of the surfaces and the area of the initial planar mesh. Observe that our method clearly outperforms [16], especially for large amounts of stretching. Furthermore, the difference in reconstruction error between the method using the true area, and our actual implementation is almost negligible. Fig. 4(right), shows a few frames with the ground truth meshes in black and the mesh reconstructed with one of the three methods in blue to help the reader to visualize what these error numbers truly represent.

However, as shown in Fig. 5, using the approximate facet areas instead of the true ones, has a more significant impact on the recovered lighting parameters. This was to be expected since object pose –and hence reconstruction errors– are always less affected by changes in the direction of the facets normal. In any case, the lighting parameters we estimate give a clear idea of what is the mean direction of the light sources and their total intensity. Fig. 5(center) plots the error in the estimation of the light intensity as a function of the percentage of occluded light sources not seen by the facets. Of course, the error increases with the amount of occlusion. Fig. 5(right) plots the error produced when estimating the light direction, which is in all the situations smaller than 25 degrees. This value is relatively small, especially if we consider that even for the most deformed shape, a change of 25 degrees in the elevation angle of the environment map only produces a 2% change in the mean image intensity.

4.2. Results on real data

We also show results on two real sequences, one involving bending an inelastic sheet of paper and the other stretching a hair ribbon. The images were acquired with a Basler A601f firewire camera, that was geometrically calibrated, and whose radiometric response was linearized. To establish the 3D-to-2D correspondences we followed a similar strategy as in [16]: starting from the SIFT [9] matches between a reference frame and the input image, the surface was detected in 2D. This 2D detection was then used to compute dense correspondences based on normalized cross-correlation in very small regions. To facilitate registration, we use a reference image in which the surface was planar and seen under diffuse lighting, so that image intensity could be directly used to estimate albedo. To obtain reliable intensity estimates in both reference and input images, we took it to be the mean over small image patches.

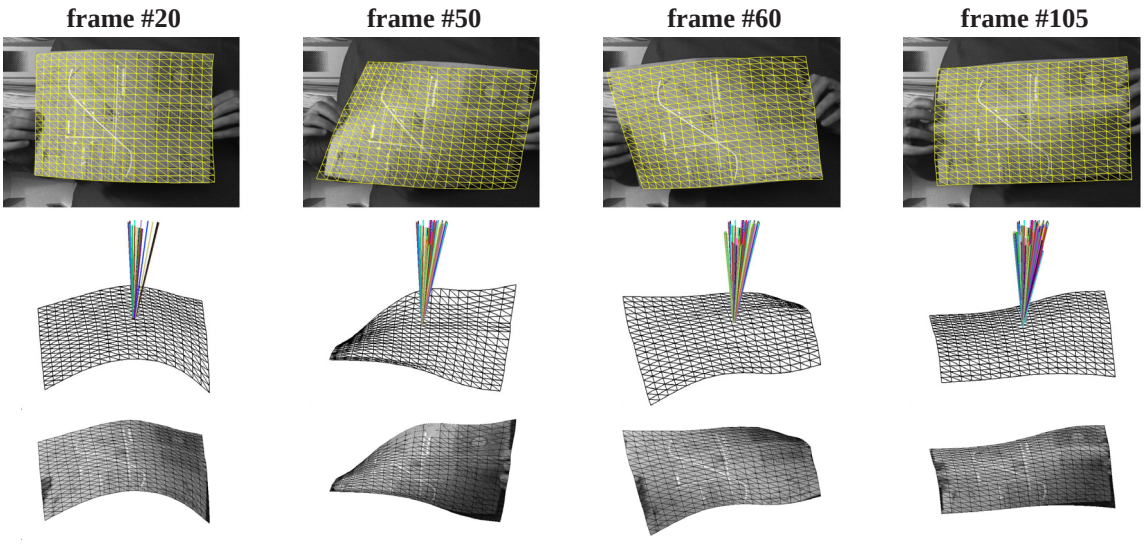


Figure 6. 3D registration of a non-rigid inelastic piece of paper. **Top row:** Retrieved mesh projected onto the original image. **Middle row:** 3D mesh seen from a different viewpoint. The colored lines in each image represent the light directions retrieved for all the previous frames in the sequence. **Bottom row:** Synthesized textured view of the retrieved shape.

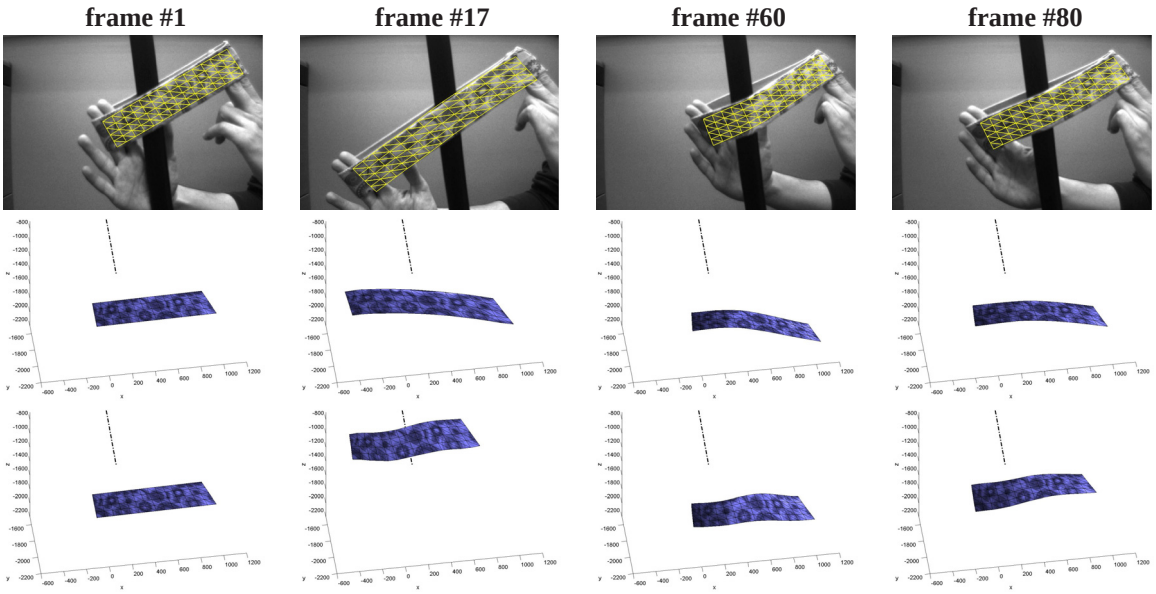


Figure 7. 3D registration of a non-rigid elastic hair ribbon. **Top and middle rows:** Registration and 3D reconstruction results obtained with the method presented in the paper. Despite the area extension, the shape is correctly retrieved. **Bottom row:** 3D reconstruction based on [16], which makes use of inextensibility constraints. Observe that the increase in area size is not detected, and the size change in the image plane is interpreted as a translation of the object towards the camera. The dot-and-dash line indicates the camera optical axis.

In the 120-frame video sequence depicted by Figure 6, we show that our method can be successfully applied to detect an inelastic deformable surface. Note the that the light source is a window located behind the camera, and therefore an extended one as opposed to a point light source. The top row depicts the recovered 19×19 meshes overlaid on the original frames. In the middle row, the computed meshes are seen from a different viewpoint. For each frame the mean direction of the light sources computed in all the previous frames is also shown as a random color line. Note that all the estimated directions form a cone with a relatively small apex angle, meaning that more or less the same light direction is retrieved in all the frames even though the computation is performed completely independently in all frames. This direction is roughly correct because it coincides with the window direction from where the light comes. The last row of Fig. 6 shows a synthetically generated textured view. In Fig. 8, we plot the estimated

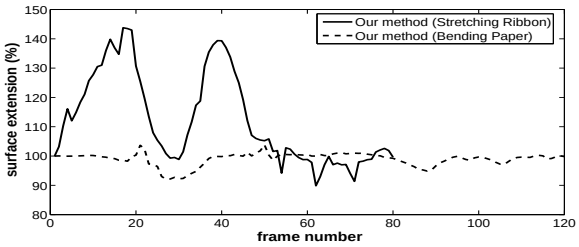


Figure 8. Surface extension estimated by our algorithm for the results with real data. Note that in the case of the inelastic paper our algorithm predicts very small extension values, while for the stretching ribbon, extensions of about 150% are estimated.

surface extension –ratio between the area of the surfaces at a given frame and the area of the initial mesh–, which remains close to one, as it should, even though we do not explicitly enforce this constraint.

Finally, we compare again our algorithm to [16] on an 80-frame sequence of a hair-ribbon being stretched and bent. The first two images in the top row of Fig. 7 show the configurations with minimal and maximal stretching, while the other two images on the right show the ribbon being bent backwards. Observe again in Fig. 8 that our algorithm correctly sees the mesh as being stretched by about 150%. This is in contrast to the results by applying [16], which interprets the motion between frame #1 and #17 as if the ribbon moved towards the camera.

5. Conclusion

In this paper we have presented a closed-form solution to 3D shape recovery of stretchable surfaces from point correspondences between an input image and a reference configuration. Since state-of-the-art methods make use of inextensibility contraints between neighboring points, they are limited to retrieving the shape of inelastic surfaces. We remove this limitation by replacing the constant-length constraints by shading ones, which still permit solving the problem in closed form.

In future work we plan to use more complex shading models and parameterizations of the lighting environment map, such as those based on spherical harmonics. To this end that we will have to consider additional unknowns and introduce visibility constraints accounting for the visible light directions for each facet. This will entail an iterative optimization scheme. But, since the closed form approach we have presented here can, in practice, handle somewhat extended light sources, we believe that it will give us the initial estimates we need to ensure convergence.

References

[1] K. Bhat, C. Twigg, J. Hodgins, P. Khosla, Z.Popovic, and S. Seitz. Estimating cloth simulation parameters from video. In *Proc. SCA*, pages 37–51, 2003.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3–d faces. In *SIGGRAPH*, pages 187–194, 1999.

[3] L. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *Trans. PAMI*, 15(11):1131–1147, 1993.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *Proc. ECCV*, pages 484–498, 1998.

[5] H. Delingette, M. Hebert, and K. Ikeuchi. Deformable surfaces: A free-form shape representation. In *SPIE Geom. Methods in Comp. Vision*, pages 21–30, 1991.

[6] A. Ecker, A. D. Jepson, and K. N. Kutulakos. Semidefinite programming heuristics for surface reconstruction ambiguities. In *Proc. ECCV*, pages 127–14, 2008.

[7] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *Int. Journal Comp. Vision*, 16:35–56, 1995.

[8] B. Horn and M. Brooks. *Shape from Shading*. MIT Press, 1989.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal Comp. Vision*, 60:91–110, 2004.

[10] I. Matthews and S. Baker. Active appearance models revisited. *Int. Journal Comp. Vision*, 60:135–164, 2004.

[11] T. McInerney and D. Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. In *Proc. ICCV*, pages 518–523, 1993.

[12] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. *Trans. PAMI*, 15(6):580–591, 1993.

[13] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *Trans. PAMI*, 13:715–729, 1991.

[14] M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. In *Proc. BMVC*, 2008.

[15] M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *Proc. CVPR*, 2007.

[16] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3d surface registration. In *Proc. ECCV*, pages 581–594, 2008.

[17] L. Torresani, A. Hertzmann, and C. Bregler. Learning non-rigid 3d shape from 2d motion. In *Advances in Neural Information Processing Systems*. MIT Press, 2003.

[18] L. V. Tsap, D. B. Goldgof, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. *Trans. PAMI*, 22(5):526–543, 2000.

[19] R. Vidal and R. Hartley. Perspective nonrigid shape and motion recovery. In *Proc. ECCV*, 2008.

[20] Y. Wang, Z. Liu, G., Hua, Z. Wen, Z. Zhang, and D. Samaras. Face re-lighting from a single image under harsh lighting conditions. In *Proc. CVPR*, pages 1–8, 2007.

[21] R. White and D. Forsyth. Combining cues: Shape from shading and texture. In *Proc. CVPR*, pages 1809–1816, 2006.

[22] A. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17:17–45, 1981.

[23] L. Zhang, S. Wang, and D. Samaras. Face synthesis and recognition from a single image under arbitrary unknown lighting using a spherical harmonic basis morphable model. In *Proc. CVPR*, pages 209–216, 2005.