

# Closed-Form Solution to Non-rigid 3D Surface Registration\*

Mathieu Salzmann, Francesc Moreno-Noguer,  
Vincent Lepetit, and 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 inelastic surface from 3D-to-2D correspondences. This lets us detect and reconstruct such a surface by matching individual images against a reference configuration, which is in contrast to all existing approaches that require initial shape estimates and track deformations from image to image.

We represent the surface as a mesh, and write the constraints provided by the correspondences as a linear system whose solution we express as a weighted sum of eigenvectors. Obtaining the weights then amounts to solving a set of quadratic equations accounting for inextensibility constraints between neighboring mesh vertices. Since available closed-form solutions to quadratic systems fail when there are too many variables, we reduce the number of unknowns by expressing the deformations as a linear combination of modes. The overall closed-form solution then becomes tractable even for complex deformations that require many modes.

## 1 Introduction

3D shape recovery of deformable surfaces from individual images is known to be highly ambiguous. The standard approach to overcoming this is to introduce a deformation model and to recover the shape by optimizing an objective function [1,2,3,4,5,6,7,8] that measures the fit of the model to the data. However, in practice, this objective function is either non-convex or involves temporal consistency. Thus, to avoid being trapped in local minima, these methods require initial estimates that must be relatively close to the true shape. As a result, they have been shown to be effective for tracking, but not for registration without *a priori* shape knowledge.

By contrast, we propose here a solution to detecting and reconstructing inelastic 3D surfaces from correspondences between an individual image and a reference configuration, in closed-form, and without any initial shape estimate.

More specifically, we model flexible inelastic surfaces as triangulated meshes whose edge lengths cannot change. Given an image of the surface in a known

---

\* This work was supported in part by the Swiss National Science Foundation and in part by the European Commission under the IST-project 034307 DYVINE (Dynamic Visual Networks).



**Fig. 1.** 3D Reconstruction of non-rigid objects from an individual image and a reference configuration. Results were obtained in closed-form, without any initial estimate. Top: Recovered mesh overlaid on the original image. Bottom: Re-textured side view of the retrieved surface.

3D configuration, and correspondences between that *model image* and an *input image* in which the shape is unknown, retrieving the mesh’s vertex coordinates involves solving a rank-deficient linear system encoding the projection equations. Taking our inspiration from our recent paper on rigid object pose estimation [9], we express the solution of this linear system as a weighted sum of the corresponding matrix’s eigenvectors associated with the smallest eigenvalues. We compute these weights by using Extended Linearization [10] to solve a set of quadratic constraints that preserve edge lengths. In its simplest form, this method is only directly applicable to very small meshes because, for larger ones, the number of unknowns after Extended Linearization grows fast, thus yielding an intractable problem. We overcome this difficulty by expressing the surface deformations as a linear combination of deformation modes. This preserves the linear formulation of the correspondence problem, but dramatically reduces the size of the corresponding linear system, while improving its conditioning. Therefore, the quadratic constraints required to guarantee inextensibility are also expressed in terms of a smaller number of variables, making Extended Linearization practical. As a result, we can solve our problem in closed-form even when using enough modes to model complex deformations such as those of Fig. 1, which yields a 3D reconstruction that jointly minimizes edge length variations and reprojects correctly on the input image.

## 2 Related Work

3D reconstruction of non-rigid surfaces from images has attracted increasing attention in recent years. It is a severely under-constrained problem and many different kinds of prior models have been introduced to restrict the space of possible shapes to a manageable size.

Most of the models currently in use trace their roots to the early physics-based models that were introduced to delineate 2D shapes [11] and reconstruct relatively simple 3D ones [12].

As far as 2D problems are concerned, their more recent incarnations have proved effective for image registration [13,14] and non-rigid surface detection [15,16]. Many variations of these models have also been proposed to address 3D problems, including superquadrics [1], triangulated surfaces [2], or thin-plate splines [17]. Additionally, dimensionality reduction was introduced through modal analysis [3,18], where shapes are represented as linear combinations of deformation modes. Finally, a very recent work [19] proposes to set bounds on distances between feature points, and use them in conjunction with a thin-plate splines model to reconstruct inextensible surfaces.

One limitation of the physics-based models is that they rarely describe accurately the non-linear physics of large deformations. In theory, this could be remedied by introducing more sophisticated finite-element modeling. However, in practice, this often leads to vastly increased complexity without a commensurate gain in performance. As a result, in recent years, there has been increasing interest in statistical learning techniques that build surface deformation models from training data. Active Appearance Models [20] pioneered this approach by learning low-dimensional linear models for 2D face tracking. They were quickly followed by Active Shape Models [5] and Morphable Models [4] that extended it to 3D. More recently, linear models have also been learned for structure-from-motion applications [6,21] and tracking of smoothly deforming 3D surfaces [7].

There has also been a number of attempts at performing 3D surface reconstruction without resorting to a deformation model. One approach has been to use lighting information in addition to texture clues to constrain the reconstruction process [8], which has only been demonstrated under very restrictive assumptions on lighting conditions and is therefore not generally applicable. Other approaches have proposed to use motion models over video sequences. The reconstruction problem was then formulated either as solving a large linear system [22] or as a Second Order Cone Programming problem [23]. These formulations, however, rely on tightly bounding the vertex displacements from one frame to the next, which makes them applicable only in a tracking context where the shape in the first frame of the sequence is known.

In all the above methods, shape recovery entails minimizing an objective function. In most cases, the function is non convex, and therefore, one can never be sure to find its global minimum, especially if the initial estimate is far from the correct answer. In the rare examples formulated as convex problems [23], the solution involves temporal consistency, which again requires a good initialization.

By contrast, many closed-form solutions have been proposed for pose estimation of rigid objects [24,25,26]. In fact, the inspiration for our method came from our earlier work [9] in that field. However, reconstructing a deformable surface involves many more variables than the 6 rigid motion degrees of freedom. In the remainder of this paper, we show that this therefore requires a substantially different approach.

### 3 Closed-Form 3D Reconstruction

In this section, we show that recovering the 3D shape of a flexible surface from 3D-to-2D correspondences can be achieved by solving a set of quadratic equations accounting for inextensibility, which can be done in closed-form.

#### 3.1 Notations and Assumptions

We represent our surface as a triangulated mesh made of  $n_v$  vertices  $\mathbf{v}_i = [x_i, y_i, z_i]^T$ ,  $1 \leq i \leq n_v$  connected by  $n_e$  edges. Let  $\mathbf{X} = [\mathbf{v}_1^T, \dots, \mathbf{v}_{n_v}^T]^T$  be the vector of coordinates obtained by concatenating the  $\mathbf{v}_i$ .

We assume that we are given a set of  $n_c$  3D-to-2D correspondences between the surface and an image. Each correspondence relates a 3D point on the mesh, expressed in terms of its barycentric coordinates in the facet to which it belongs, and a 2D feature in the image.

Additionally, we assume the camera to be calibrated and, therefore, that its matrix of intrinsic parameters  $\mathbf{A}$  is known. To simplify our notations without loss of generality, we express the vertex coordinates in the camera referential.

#### 3.2 Linear Formulation of the Correspondence Problem

We first show that, given a set of 3D-to-2D correspondences, the vector of vertex coordinates  $\mathbf{X}$  can be found as the solution of a linear system.

Let  $\mathbf{x}$  be a 3D point belonging to facet  $f$  with barycentric coordinates  $[a_1, a_2, a_3]$ . Hence, we can write it as  $\mathbf{x} = \sum_{i=1}^3 a_i \mathbf{v}_{f,i}$ , where  $\{\mathbf{v}_{f,i}\}_{i=1,2,3}$  are the three vertices of facet  $f$ . The fact that  $\mathbf{x}$  projects to the 2D image location  $(u, v)$  can now be expressed by the relation

$$\mathbf{A} (a_1 \mathbf{v}_{f,1} + a_2 \mathbf{v}_{f,2} + a_3 \mathbf{v}_{f,3}) = k \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{1}$$

where  $k$  is a scalar accounting for depth. Since, from the last row of Eq. 1,  $k$  can be expressed in terms of the vertex coordinates, we have

$$[a_1 \mathbf{B} \ a_2 \mathbf{B} \ a_3 \mathbf{B}] \begin{bmatrix} \mathbf{v}_{f,1} \\ \mathbf{v}_{f,2} \\ \mathbf{v}_{f,3} \end{bmatrix} = \mathbf{0}, \text{ with } \mathbf{B} = \mathbf{A}_{2 \times 3} - \begin{bmatrix} u \\ v \end{bmatrix} \mathbf{A}_3, \tag{2}$$

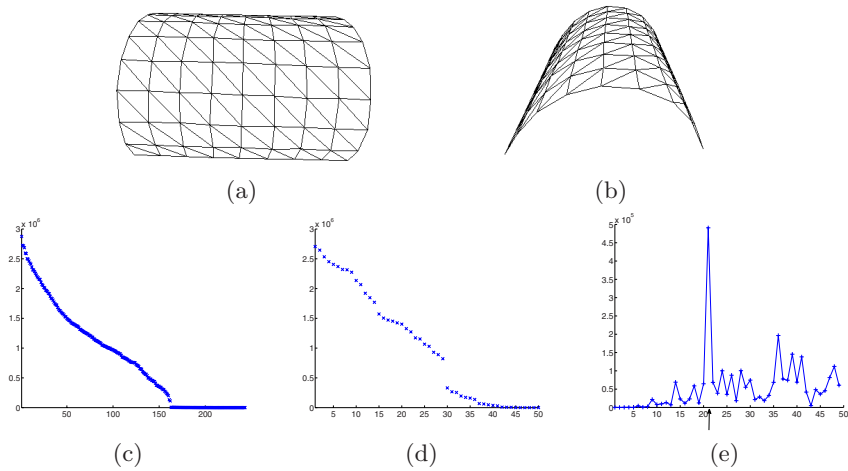
where  $\mathbf{A}_{2 \times 3}$  are the first two rows of  $\mathbf{A}$ , and  $\mathbf{A}_3$  is the third one.  $n_c$  such correspondences between 3D surface points and 2D image locations therefore provide  $2n_c$  linear constraints such as those of Eq. 2. They can be jointly expressed by the linear system

$$\mathbf{M} \mathbf{X} = \mathbf{0}, \tag{3}$$

where  $\mathbf{M}$  is a  $2n_c \times 3n_v$  matrix obtained by concatenating the  $[a_1 \mathbf{B} \ a_2 \mathbf{B} \ a_3 \mathbf{B}]$  matrices of Eq. 2.

Although solving this system yields a surface that reprojects correctly on the image, there is no guarantee that its 3D shape corresponds to reality. This stems from the fact that, for all practical purposes,  $\mathbf{M}$  is rank deficient. More specifically, even where there are many correspondences, one third, i.e.  $n_v$ , of the eigenvalues of  $\mathbf{M}^T\mathbf{M}$  are very close to zero [22], as illustrated by Fig. 2(c). As a result, even small amounts of noise produce large instability in the recovered shape.

This suggests that additional constraints have to be added to guarantee a unique and stable solution. In most state-of-the-art approaches, these constraints are provided by deformation models and are enforced via an iterative method. By contrast, we will argue that imposing inextensibility of the surface yields a closed-form solution to the problem.



**Fig. 2.** (a,b) Original and side views of a surface used to generate a synthetic sequence. The 3D shape was reconstructed by an optical motion capture system. (c,d) Eigenvalues of the linear system written from correspondences randomly established for the synthetic shape of (a). (c) The system was written in terms of 243 vertex coordinates. One third of the eigenvalues are close to zero. (d) The system was written in terms of 50 PCA modes. There are still a number of near zero eigenvalues. (e) First derivative of the curve (d) (in reversed  $x$ -direction). We take the maximum value of  $n_l$  to be the one with maximum derivative, which corresponds to the jump in (d).

### 3.3 Inextensible Meshes

Following the idea introduced in [9], we write the solution of the linear system of Eq. 3 as a weighted sum of the eigenvectors  $\mathbf{l}_i$ ,  $1 \leq i \leq n_v$  of  $\mathbf{M}^T\mathbf{M}$ , which are those associated with the eigenvalues that are almost zero. Therefore we write

$$\mathbf{X} = \sum_{i=1}^{n_v} \beta_i \mathbf{l}_i, \tag{4}$$

since any such linear combination of  $\mathbf{l}_i$  is in the kernel of  $\mathbf{M}^T\mathbf{M}$  and produces a mesh that projects correctly on the image. Our problem now becomes finding appropriate values for the  $\beta_i$ , which are the new unknowns.

We are now in a position to exploit the inextensibility of the surface by choosing the  $\beta_i$  so that edge lengths are preserved. Such  $\beta_i$  can be expressed as the solution of a set of quadratic equations of the form

$$\left\| \sum_{i=1}^{n_v} \beta_i \mathbf{l}_i^j - \sum_{i=1}^{n_v} \beta_i \mathbf{l}_i^k \right\|^2 = \left\| \mathbf{v}_j^{ref} - \mathbf{v}_k^{ref} \right\|^2, \tag{5}$$

where  $\mathbf{l}_i^j$  is the  $3 \times 1$  sub-vector of  $\mathbf{l}_i$  corresponding to the coordinates of vertex  $\mathbf{v}_j$ , and  $\mathbf{v}_j^{ref}$  and  $\mathbf{v}_k^{ref}$  are two neighboring vertices in the reference configuration.

### 3.4 Extended Linearization

Typical closed-form approaches to solving systems of quadratic equations involve linearizing the system and introducing new unknowns for the quadratic terms. This results in a system of the form

$$\mathbf{D}\mathbf{b} = \mathbf{d}, \tag{6}$$

where  $\mathbf{b} = [\beta_1\beta_1, \dots, \beta_1\beta_{n_v}, \beta_2\beta_2, \dots, \beta_2\beta_{n_v}, \dots, \beta_{n_v}\beta_{n_v}]^T$  is the vector of quadratic terms, of size  $n_v(n_v + 1)/2$ .  $\mathbf{D}$  is a  $n_e \times n_v(n_v + 1)/2$  matrix built from the known  $\mathbf{l}_i$ , and  $\mathbf{d}$  is the  $n_e \times 1$  vector of edge lengths in the reference configuration. Unfortunately, since, in hexagonal meshes, the number of edges grows as  $3n_v$ , the number of quadratic unknown terms in the linearized system quickly becomes larger than the number of equations.

In this paper, we solve this problem by using Extended Linearization [10], a simple and powerful approach to creating new equations in a linearized system, which performs better than Groebner bases and relinearization. The idea is to multiply the original set of equations by the monomials, and linearize the resulting system. In our particular case, we can, for example, multiply the existing quadratic equations by each of the linear terms, thus creating new equations of the form

$$\begin{aligned} \beta_1 \left( \left\| \sum_{i=1}^{n_v} \beta_i \mathbf{l}_i^j - \sum_{i=1}^{n_v} \beta_i \mathbf{l}_i^k \right\|^2 \right) &= \beta_1 \left( \left\| \mathbf{v}_j^{ref} - \mathbf{v}_k^{ref} \right\|^2 \right), \\ &\vdots \\ \beta_{n_v} \left( \left\| \sum_{i=1}^{n_v} \beta_i \mathbf{l}_i^j - \sum_{i=1}^{n_v} \beta_i \mathbf{l}_i^k \right\|^2 \right) &= \beta_{n_v} \left( \left\| \mathbf{v}_j^{ref} - \mathbf{v}_k^{ref} \right\|^2 \right). \end{aligned}$$

Let  $\mathbf{b}^c = [\beta_1\beta_1\beta_1, \dots, \beta_1\beta_1\beta_{n_v}, \beta_1\beta_2\beta_2, \dots, \beta_1\beta_2\beta_{n_v}, \beta_2\beta_2\beta_2, \dots, \beta_{n_v}\beta_{n_v}\beta_{n_v}]^T$ , and  $\mathbf{b}^l = [\beta_1, \dots, \beta_{n_v}]^T$ . The resulting system can be written as

$$\begin{bmatrix} 0 & \dots & 0 & \mathbf{D}_1^{1,1} & \dots & \mathbf{D}_1^{1,n_v} & \mathbf{D}_1^{2,2} & \dots & \mathbf{D}_1^{n_v,n_v} & 0 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ -\mathbf{d}_1 & 0 & \dots & 0 & \dots & \dots & \dots & \dots & 0 & \mathbf{D}_1^{1,1} & \dots & \mathbf{D}_1^{n_v,n_v} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \mathbf{b}^l \\ \mathbf{b} \\ \mathbf{b}^c \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{0} \\ \vdots \end{bmatrix}, \tag{7}$$

where we only show the first line of the original system of Eq. 6 and its product with  $\beta_1$ , and where  $\mathbf{D}_1^{i,j}$  stands for the coefficient on the first line of  $\mathbf{D}$  corresponding to the product  $\beta_i\beta_j$ .

It can be shown that multiplying the inextensibility equations by all the  $\beta_i$  only yields a sufficient number of equations for very small meshes, i.e. less than 12 vertices for a hexagonal mesh. In theory, one could solve this problem by applying Extended Linearization iteratively by re-multiplying the new equations by the linear terms. However, in practice, the resulting system quickly becomes so large that it is intractable, i.e. for a  $10 \times 10$  mesh, the number of equations only becomes larger than the number of unknowns when the size of the system is of the order  $10^{10}$ . In other words, Extended Linearization cannot deal with a problem as large as ours and we are not aware of any other closed-form approach to solving systems of quadratic equations that could. We address this issue in the next section.

### 3.5 Linear Deformation Model

As discussed above, to solve the set of quadratic equations that express edge length preservation, we need to reduce its size to the point where Extended Linearization becomes a viable option. Furthermore, we need to do this in such a way that the solution of the correspondence problem can still be expressed as the solution of a system of linear equations, as discussed in Section 3.2. To this end, we model the plausible deformations of the mesh as a linear combination of  $n_m$  deformation modes [6,7], much in the same spirit as those the morphable models used to represent face deformations [4]. We write

$$\mathbf{X} = \mathbf{X}_0 + \sum_{i=1}^{n_m} \alpha_i \mathbf{p}_i = \mathbf{X}_0 + \mathbf{P}\alpha , \tag{8}$$

where the  $\mathbf{p}_i$  are the deformation modes and the  $\alpha_i$  their associated weights. In our implementation, modes were obtained by applying Principal Component Analysis to a matrix of registered training meshes in deformed configurations, from which the mean shape  $\mathbf{X}_0$  was subtracted [7]. The  $\mathbf{p}_i$  therefore are the eigenvectors of the data covariance matrix. Nonetheless, they could also have been derived by modal analysis, which amounts to computing the eigenvectors of a stiffness matrix, and is a standard approach in physics-based modeling [3].

In this formulation, recovering the shape amounts to computing the weights  $\alpha$ . Since the shape must satisfy Eq. 3,  $\alpha$  must then satisfy

$$\mathbf{M}(\mathbf{X}_0 + \mathbf{P}\alpha) = \mathbf{0} . \tag{9}$$

When solving this system, to ensure that the recovered weights do not generate shapes exceedingly far from our training data, we introduce a regularization term by penalizing  $\alpha_i$  with the inverse of the corresponding eigenvalue  $\sigma_i$  of the data covariance matrix. We therefore solve

$$\begin{bmatrix} \mathbf{MP} & \mathbf{MX}_0 \\ w_r \mathbf{S} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \alpha \\ 1 \end{bmatrix} = \mathbf{0} , \tag{10}$$

where  $\mathbf{S}$  is an  $n_m \times n_m$  diagonal matrix whose elements are the  $\sigma_i^{-1}$  and  $w_r$  is a regularization weight that only depends on the maximum  $\sigma_i$ , and whose precise value has only little influence on the results.

As shown in Fig. 2(d), we have considerably reduced the number of near-zero eigenvalues. The system of Eq. 10 is therefore better conditioned than the one of Eq. 3, but still does not yield a well-posed problem that would have a unique solution. This is attributable to the fact that, because the solution is expressed as a sum of deformation modes, inextensibility constraints, which are non linear, are not enforced.

Nonetheless, we can follow the same procedure as in Sections 3.3 and 3.4. We write the solution of the linear system of Eq. 10 as a weighted sum of the eigenvectors  $\tilde{\mathbf{l}}_i$ ,  $1 \leq i \leq n_l \ll n_m$  associated with the smallest eigenvalues of its matrix, and find the weights  $\beta_i$  as the solution of the linearized system of quadratic equations

$$\tilde{\mathbf{D}}\tilde{\mathbf{b}} = \tilde{\mathbf{d}}, \quad (11)$$

where  $\tilde{\mathbf{b}} = [\tilde{\beta}_1, \dots, \tilde{\beta}_{n_l}, \tilde{\beta}_1\tilde{\beta}_1, \dots, \tilde{\beta}_1\tilde{\beta}_{n_l}, \tilde{\beta}_2\tilde{\beta}_2, \dots, \tilde{\beta}_2\tilde{\beta}_{n_l}, \dots, \tilde{\beta}_{n_l}\tilde{\beta}_{n_l}]^T$  now also contains the linear terms arising in the quadratic equations from the mean shape  $\mathbf{X}_0$ . Furthermore, the system also encodes the additional linear equation that constrains the  $\tilde{\beta}_i\tilde{\mathbf{l}}_{i,n_m+1}$  to sum up to 1, where  $\tilde{\mathbf{l}}_{i,n_m+1}$  is the last element of  $\tilde{\mathbf{l}}_i$ .

Since in practice  $n_l \ll n_m \ll n_v$ , the system is now much smaller. Therefore a single iteration of Extended Linearization is sufficient to constrain its solution while keeping it tractable, even for relatively large numbers of modes—in practice up to 60—thus allowing complex deformations.

In this formulation, the number  $n_l$  of eigenvectors strongly depends on the number  $n_m$  of modes used for the recovery. However, as shown in Fig. 2(e), we can easily set the maximum number  $\hat{n}_l$  of eigenvectors to use by picking the number corresponding to the maximum first derivative of the ordered eigenvalues curve. We then simply test for all  $n_l \leq \hat{n}_l$  and pick the optimal value as the one that, for a small enough reprojection error, gives the smallest mean edge length variation. In practice,  $\hat{n}_l$  was typically about 25 when using 60 deformation modes.

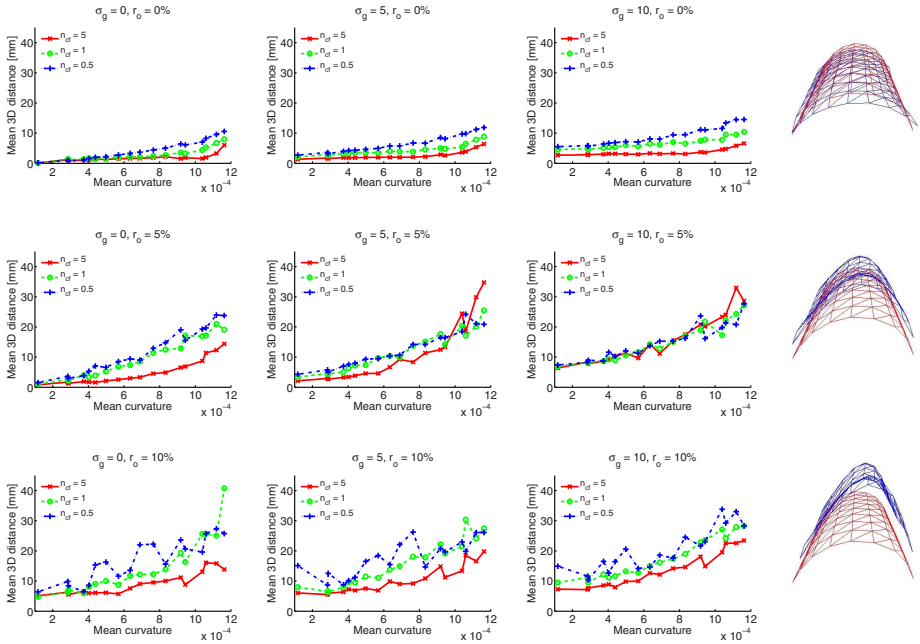
## 4 Experimental Results

In this section we show that our method can be successfully applied to reconstructing non-rigid shapes from individual images and a reference configuration. We present results on both synthetic data and real images.

### 4.1 Synthetic Data

We first applied our method to images, such as those of Fig. 2(a), synthesized by projecting known deformed shapes using a virtual camera. The deformed shapes were obtained by recovering the 3D locations of reflective markers stuck on a 200×200mm piece of cardboard with an optical motion capture system. This allowed us to randomly create  $n_{cf}$  perfect correspondences per facet to

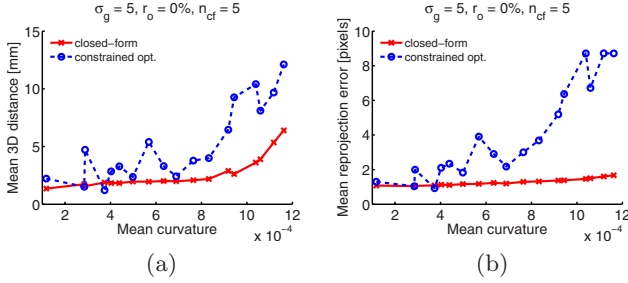




**Fig. 3.** Shape recovery of a  $200 \times 200$ mm synthetic mesh imaged by a virtual camera placed 20cm away from it. Each plot shows the mean vertex-to-vertex 3D distance between the recovered surface and the ground-truth as a function of its mean curvature. The three different curves in each graph correspond to a varying number of correspondences per facet. Left to right, the gaussian noise added to the correspondences increases. Top to bottom, the number of outliers grows. For each experiments, we plot the average over 40 trials. The rightmost column shows in blue recovered shapes for the ground-truth surface of Fig. 2(a,b), shown in red. The corresponding mean vertex-to-vertex distances are 9mm, 19mm and 38mm. This highlights the fact that even for distances around 40mm, the recovered shape remains meaningful.

which we added zero mean gaussian noise of variance  $\sigma_g$ . Finally, we simulated outliers by setting the image coordinates of  $r_o$  percents of the correspondences to uniformly and randomly distributed values.

In Fig. 3, we show results as a function of the surface's mean curvature, the maximum one being that of Fig. 2(a). Each plot includes three curves corresponding to  $n_{cf} = \{5, 1, 1/2\}$ , which depict the mean vertex-to-vertex 3D distance between the recovered mesh and ground-truth. The plots are ordered on a grid whose  $x$ -direction corresponds to  $\sigma_g = \{0, 5, 10\}$  and  $y$ -direction to  $r_o = \{0\%, 5\%, 10\%\}$ . Each experiment was repeated 40 times, and we show the average results. Note that the error grows with the mean curvature of the shape, which is natural since the shape becomes more ambiguous when seen from the viewpoint shown in Fig. 2(a). In the rightmost column, we display three shapes reconstructed from the image of Fig. 2(a) with their corresponding ground-truth. Note that even for average distances of 40mm between the true and recovered



**Fig. 4.** Comparison of our closed-form results against the results of constrained optimization. Optimization was performed on the vertex coordinates using Matlab’s `fmincon` function, and starting from the flat position. (a) Mean vertex-to-vertex distance. (b) Reprojection error. Constrained optimization is both much slower and far less accurate than our approach.

shape, the latter remains meaningful and could be used to initialize an iterative algorithm.

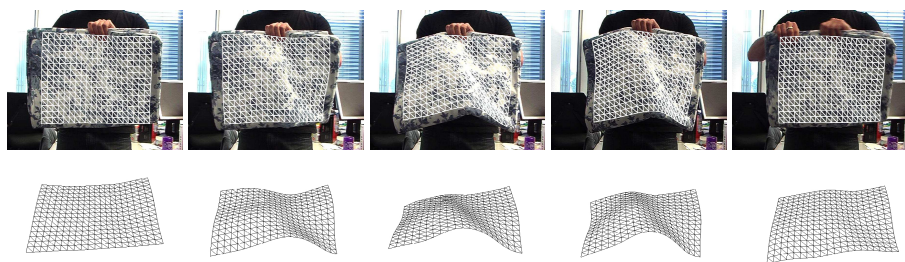
In Fig. 4, we compare our results against results obtained with Matlab’s constrained optimization `fmincon` function. We use it to minimize the residual of the linear system of Eq. 3 with respect to the vertex coordinates, under the constraints that edge lengths must remain constant. We first tried to use the similar representation in terms of modes. However, since the constraints could never be truly satisfied, the algorithm would never converge towards an acceptable solution. This forced us to directly use the vertex coordinates. To improve convergence and prevent the surface from crumpling, we added a smoothness term [11]. For all the frames, the initialization was set to the flat position. In Fig. 4(a), we show the mean 3D vertex-to-vertex distance for the case where  $\sigma_g = 5$ ,  $r_o = 0$ , and  $n_{cf} = 5$ . The red curve corresponds to our closed-form solution and the blue one to constrained optimization. Note that our approach gives much better results. Furthermore, it is also much faster, requiring only 1.5 minutes per frame as opposed to 1.5 hours for constrained optimization. Fig. 4(b) shows the reprojection errors for the same cases.

## 4.2 Real Images

We tested our method on a folded bed-sheet, a piece of cloth and a t-shirt deforming in front of a 3-CCD DV-camera. In all these cases, we first established SIFT [27] correspondences between the reference image and the input one. We then detected the surface in 2D, which can be done in closed-form by simply solving the linear system built from SIFT matches, augmented with linear smoothing equations [11]. For each facet, we then warped the reference image to best match the input one based on the retrieved 2D shape, and finally established dense correspondences by sampling the barycentric coordinates of the facet, and matching small regions between the input image and the warped reference one using normalized cross-correlation. Note that, even when we show results on



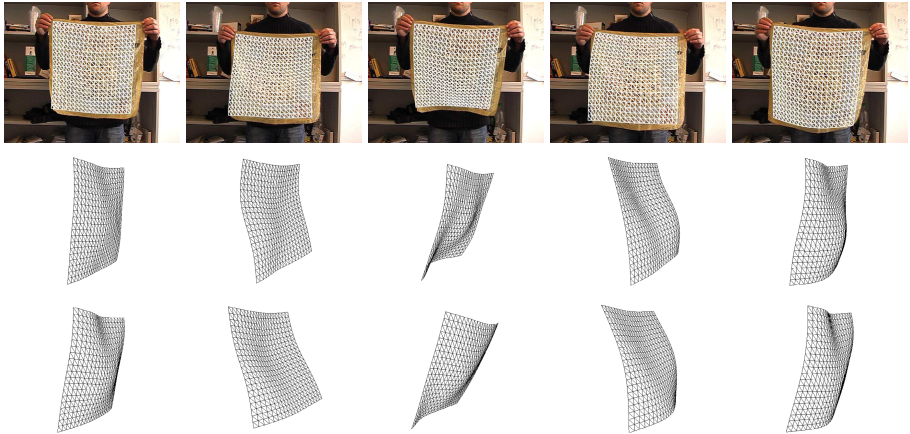
**Fig. 5.** 3D registration of a folded bed-sheet to an individual image given a reference configuration. Top Row: Recovered mesh overlaid on the original image. Middle Row: Synthesized textured view using the recovered shape. Bottom Row: Real side view of the sheet from similar viewpoints. Despite lighting changes, the synthetic images closely match the real ones.



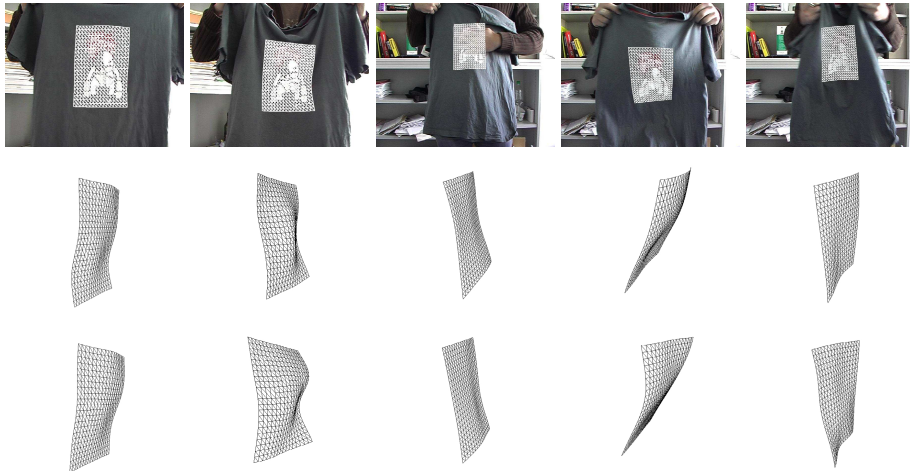
**Fig. 6.** Shape recovery of a bed-sheet. Top Row: Recovered mesh overlaid on the original image. Bottom Row: Mesh seen from a different viewpoint.

video sequences, nothing links one frame to the next, and no initialization is required. Corresponding videos are given as supplementary material.

In the case of the sheet, we deformed it into several unrelated shapes, took pictures from 2 different views for each deformation, and reconstructed the surface from a single image and a reference configuration. In Fig. 5, we show the results on four different cases. From our recovered shape, we generated synthetic textured images roughly corresponding to the viewpoint of the second image. As can be seen in the two bottom rows of Fig. 5, our synthetic images closely match the real side views. Additionally, we also reconstructed the same sheet



**Fig. 7.** Shape recovery of a piece of cloth. From Top to Bottom: Mesh computed in closed-form overlaid on the input image, side view of that mesh, refined mesh after 5 Gauss-Newton iterations.



**Fig. 8.** Shape recovery of the central part of a t-shirt. From Top to Bottom: Mesh computed in closed-form overlaid on the input image, side view of that mesh, refined mesh after 5 Gauss-Newton iterations.

from the images of a video sequence, and show the results in Fig. 6. Note that no initialization was required, and that nothing links one frame to the next.

In Figs. 7 and 8, we show results for images of a piece of cloth and of a t-shirt waved in front of the camera. Note that in both cases, the closed-form solution closely follows what we observe in the videos. To further refine it, we implemented a simple Gauss-Newton optimization technique, and minimize the

residual  $\|\tilde{\mathbf{D}}\tilde{\mathbf{b}} - \tilde{\mathbf{d}}\|$  corresponding to Eq. 11 with respect to the  $\tilde{\beta}_i$ . In the third row of the figures, we show the refined mesh after 5 iterations this scheme. This proved sufficient to recover finer details at a negligible increase in overall computation time.

## 5 Conclusion

In this paper, we presented a closed-form solution to the problem of recovering the shape of a non-rigid inelastic surface from an individual image and a reference configuration. We showed that the reconstruction could be obtained by solving a system of quadratic equations representing distance constraints between neighboring mesh vertices.

In future work, we intend to investigate what additional quadratic constraints could be introduced to the current system of distance constraints. They could come from additional sources of image information, such as lighting. Having a larger number of quadratic equations would hopefully relieve the need for Extended Linearization, and result in smaller, and therefore faster to solve, linear systems.

## References

1. Metaxas, D., Terzopoulos, D.: Constrained deformable superquadrics and nonrigid motion tracking. *PAMI* 15, 580–591 (1993)
2. Cohen, L., Cohen, I.: Deformable models for 3-d medical images using finite elements and balloons. In: *CVPR*, pp. 592–598 (1992)
3. Pentland, A.: Automatic extraction of deformable part models. *IJCV* 4, 107–126 (1990)
4. Blanz, V., Vetter, T.: A Morphable Model for The Synthesis of 3-D Faces. *ACM SIGGRAPH*, 187–194 (1999)
5. Matthews, I., Baker, S.: Active Appearance Models Revisited. *IJCV* 60, 135–164 (2004)
6. Torresani, L., Hertzmann, A., Bregler, C.: Learning non-rigid 3d shape from 2d motion. In: *NIPS* (2003)
7. Salzmann, M., Pilet, J., Ilić, S., Fua, P.: Surface Deformation Models for Non-Rigid 3-D Shape Recovery. *PAMI* 29, 1481–1487 (2007)
8. White, R., Forsyth, D.: Combining cues: Shape from shading and texture. In: *CVPR* (2006)
9. Moreno-Noguer, F., Lepetit, V., Fua, P.: Accurate Non-Iterative  $O(n)$  Solution to the  $PnP$  Problem. In: *ICCV* (2007)
10. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) *EUROCRYPT 2000*. LNCS, vol. 1807. Springer, Heidelberg (2000)
11. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. *IJCV* 1, 321–331 (1988)
12. Terzopoulos, D., Witkin, A., Kass, M.: Symmetry-seeking Models and 3D Object Reconstruction. *IJCV* 1, 211–221 (1987)

13. Bartoli, A., Zisserman, A.: Direct Estimation of Non-Rigid Registration. In: BMVC (2004)
14. Gay-Bellile, V., Bartoli, A., Sayd, P.: Direct estimation of non-rigid registrations with image-base self-occlusion reasoning. In: ICCV (2007)
15. Pilet, J., Lepetit, V., Fua, P.: Real-Time Non-Rigid Surface Detection. In: CVPR (2005)
16. Zhu, J., Lyu, M.R.: Progressive finite newton approach to real-time nonrigid surface detection. In: ICCV (2007)
17. McInerney, T., Terzopoulos, D.: A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking. In: ICCV (1993)
18. Delingette, H., Hebert, M., Ikeuchi, K.: Deformable surfaces: A free-form shape representation. *Geometric Methods in Computer Vision* (1991)
19. Perriollat, M., Hartley, R., Bartoli, A.: Monocular Template-based Reconstruction of Inextensible Surfaces. In: BMVC (2008)
20. Cootes, T., Edwards, G., Taylor, C.: Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407. Springer, Heidelberg (1998)
21. Llado, X., Bue, A.D., Agapito, L.: Non-rigid 3D Factorization for Projective Reconstruction. In: BMVC (2005)
22. Salzmann, M., Lepetit, V., Fua, P.: Deformable Surface Tracking Ambiguities. In: CVPR (2007)
23. Salzmann, M., Hartley, R., Fua, P.: Convex Optimization for Deformable Surface 3-D Tracking. In: ICCV (2007)
24. Quan, L., Lan, Z.: Linear N-Point Camera Pose Determination. *PAMI* 21, 774–780 (1999)
25. Fiore, P.D.: Efficient linear solution of exterior orientation. *PAMI* 23, 140–148 (2001)
26. Ansar, A., Daniilidis, K.: Linear pose estimation from points or lines. *PAMI* 25, 578–589 (2003)
27. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. *IJCV* 20, 91–110 (2004)