Reconstructing Sharply Folding Surfaces: A Convex Formulation*

Mathieu Salzmann
EECS & ICSI
UC Berkeley
salzmann@icsi.berkeley.edu

Pascal Fua
EPFL - CVLab
1015 Lausanne, Switzerland
pascal.fua@epfl.ch

Abstract

In recent years, 3D deformable surface reconstruction from single images has attracted renewed interest. It has been shown that preventing the surface from either shrinking or stretching is an effective way to resolve the ambiguities inherent to this problem. However, while the geodesic distances on the surface may not change, the Euclidean ones decrease when folds appear. Therefore, when applied to discrete surface representations, such constant-distance constraints are only effective for smoothly deforming surfaces, and become inaccurate for more flexible ones that can exhibit sharp folds. In such cases, surface points must be allowed to come closer to each other.

In this paper, we show that replacing the equality constraints of earlier approaches by inequality constraints that let the mesh representation of the surface shrink but not expand yields not only a more faithful representation, but also a convex formulation of the reconstruction problem. As a result, we can accurately reconstruct surfaces undergoing complex deformations that include sharp folds from individual images.

1. Introduction

Being able to recover the 3D shape of deformable surfaces using a single camera would make it possible to field reconstruction systems that run on widely available hardware. However, because many different 3D shapes can have virtually the same projection, such monocular shape recovery is inherently ambiguous.

The solutions that have been proposed over the years mainly fall into two classes: Those that involve physics-inspired models [23, 6, 15, 14, 17, 16, 25, 2] and those that rely on a non-rigid structure-from-motion approach [5, 27, 1, 12, 24, 26]. The former solutions often entail designing complex objective functions and require hard-to-obtain knowledge about the precise material properties of the target surfaces. The latter depend on points being reliably tracked in image sequences and are only effective for relatively small deformations.

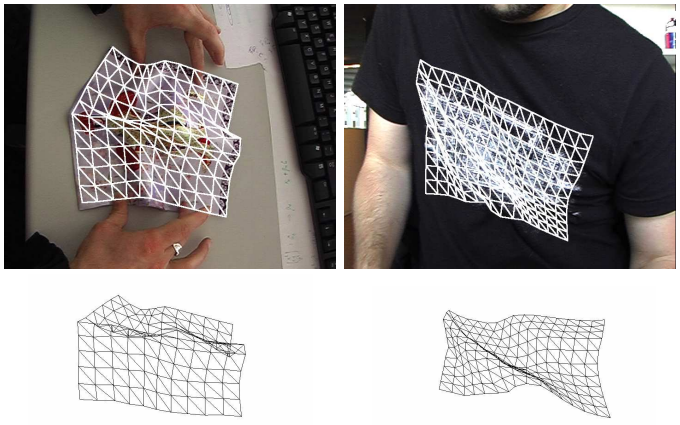


Figure 1. Reconstruction of highly flexible surfaces undergoing complex deformations. Top Row: Reconstructed 3D mesh overlaid on the input image. Bottom Row: Side view of the same mesh. As can be observed from the side view, our method correctly recovers the folds of the cloth and of the t-shirt.

Recently, it has been shown that simply constraining the distances between selected surface points to remain constant is enough to recover 3D shape from a single input image, provided that point correspondences can be established with a reference image in which the shape is known [20, 8, 19]. This makes it an attractive alternative to the techniques mentioned above when dealing with materials such as paper or cardboard that do not fold sharply. However, when dealing with more flexible materials such as the cloth and the t-shirt of Fig. 1, preventing surface points from moving closer to each other is an overly strong constraint. As shown in Fig. 2, even though the geodesic distances between surface points remain constant, the Euclidean ones decrease when folds appear.

In this paper, we propose a convex formulation that lets us correctly model folds and recover complex 3D shapes without requiring an initial guess. To this end, we replace

*This work has been funded in part by the Swiss National Science Foundation

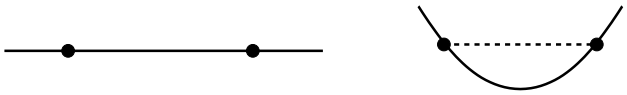


Figure 2. Schematic representation of why inextensibility constraints are ill-suited. Left: Two points of the discrete representation of a continuous surface in its rest configuration. Right: When deformed, while the geodesic distance between the two points is preserved, the Euclidean one decreases. This suggests that distance inequality constraints should be used rather than equalities.

the distance equality constraints of earlier techniques by inequality constraints that allow the vertices of the surface mesh representation to come closer to each other, but prevent them from moving further apart than their geodesic distance. Because of the scale ambiguity inherent to monocular shape reconstruction, these inequality constraints do not fully disambiguate the problem on their own; the surface can simply shrink until all distances are below the threshold. We overcome this problem by maximizing the distance to the camera of selected surface points under the inequality constraints. This can be formulated as maximizing a linear criterion under linear and quadratic constraints, which yields a convex problem that can be solved using standard mathematical routines [4].

Furthermore, when there are too few correspondences for shape recovery to be effective without additional knowledge, we introduce a linear local surface deformation model and a motion model that preserve the convexity of our formulation. These models adequately fill in the missing information while being flexible enough to allow reconstruction of complex deformations such as those of Fig. 1.

2. Related Work

3D reconstruction of non-rigid surfaces from single images is a severely under-constrained problem since many different shapes can produce very similar projections. Many methods have therefore been proposed over the years to give preference to the most likely shapes and disambiguate the problem.

The earliest approaches were inspired by physics and involved minimizing the difference between an internal energy representing the physical behavior of the surface and an external one derived from image data [23]. Many variations, such as balloons [6], deformable superquadrics [15] and thin-plates under tension [14], have since been proposed. Modal analysis was applied to reduce the number of degrees of freedom of the problem by modeling the deformations as linear combinations of vibration modes [17, 16]. Since these formulations oversimplify reality, especially in the presence of large deformations, more accurate but also more complex non-linear models were proposed [25, 2]. In short, even though incorporating physical laws into the algorithms seems natural, the resulting methods suffer from two major drawbacks. First, one must specify material parameters that are typically unknown. Second, making them accurate in the presence of large deformations requires designing very complex objective functions that are often difficult to optimize.

Methods that learn models from training data were introduced to overcome these limitations. Active Appearance Models pioneered this approach for faces [7] in 2D and were quickly followed by 3D Morphable Models [3]. As in modal analysis, surface deformations are expressed as linear combinations of deformation modes. These modes, however, are obtained from training examples rather than from stiffness matrices and can therefore capture more of the true variability, but only when sufficient training data is available.

Non-rigid structure-from-motion methods expanded on this idea by simultaneously recovering the shape and the modes from image sequences [5, 27, 1, 12, 24, 26]. While this is a very attractive idea, few implementations are truly practical because they require points to be tracked throughout the whole sequence. Furthermore, they are only effective for relatively small deformations since using a large number of deformation modes makes the solution more ambiguous.

Several methods have recently been proposed to recover the shape of inextensible surfaces. Some are specifically designed for applicable surfaces, such as sheets of paper [9, 11, 18]. Others explicitly incorporate the fact that the distances between surface points must remain constant as constraints in the reconstruction process [20, 8, 19]. This approach is a very attractive alternative to the earlier techniques, since many materials do not perceptibly shrink or stretch as they deform. However, as mentioned above, because the distances between points of a discrete surface representation can decrease in the presence of folds, these constraints are too restrictive for very flexible materials. By contrast, the approach we propose relies on distance inequalities, which are better suited for sharply folding surfaces. Furthermore, such constraints yield a convex formulation that can be made robust to noise and mismatches.

3. Problem Formulation

We now introduce our convex formulation of the 3D reconstruction problem. We represent the surface as a triangulated 3D mesh and assume we are given a set of correspondences between 3D surface points and 2D locations in an input image. In practice, we obtain them by matching SIFT features [13] between the input image and a reference image in which we know the surface shape. The 2D points in the reference image correspond to 3D points on the mesh that we express in terms of the barycentric coordinates with respect to the facet they belong to.

To simplify our notations, we express all 3D coordinates

in the camera referential. This entails no loss of generality since the surface can move rigidly with respect to the camera.

3.1. Noise-Free Shape Recovery

Let $\mathbf{A}$ be the matrix of known internal camera parameters and $\mathbf{q}_i = [u_i \ v_i]^T$ a feature point in the input image. The line-of-sight $\mathbf{s}_i$ defined by $\mathbf{q}_i$ can be written as

$$\mathbf{s}_i = \frac{\mathbf{A}^{-1}\left[\mathbf{q}_i^T \ \ 1\right]^T}{\|\mathbf{A}^{-1}\left[\mathbf{q}_i^T \ \ 1\right]^T\|} \ . \tag{1}$$

Let $\mathbf{p}_i$ be the 3D point projecting at $\mathbf{q}_i$. In the absence of noise, the position of $\mathbf{p}_i$ is entirely defined by its distance d_i from the camera along $\mathbf{s}_i$. Furthermore, if $\mathbf{p}_i$ belongs to the facet whose vertices are $\mathbf{v}_j$, $\mathbf{v}_k$, and $\mathbf{v}_l$, it can also be expressed as

$$\mathbf{p}_i = \begin{bmatrix} b_i^j & 0 & 0 & b_i^k & 0 & 0 & b_i^l & 0 & 0 \\ 0 & b_i^j & 0 & 0 & b_i^k & 0 & 0 & b_i^l & 0 \\ 0 & 0 & b_i^j & 0 & 0 & b_i^k & 0 & 0 & b_i^l \end{bmatrix} \begin{bmatrix} \mathbf{v}_j \\ \mathbf{v}_k \\ \mathbf{v}_l \end{bmatrix} \ , \tag{2}$$

where b_i^j, b_i^k, and b_i^l are its barycentric coordinates.

Given N_c such correspondences spread over all the facets of a mesh, recovering its 3D shape amounts to solving the feasibility problem

$$\begin{aligned} &\text{find} \ \ \mathbf{X}\,,\mathbf{d} \\ &\text{subject to} \ \ \mathbf{B}_i\mathbf{X} = d_i\mathbf{s}_i \ , \ 1 \leq i \leq N_c \ , \end{aligned}$$

where $\mathbf{X}$ is the vector of concatenated x-, y-, z-coordinates of the N_v mesh vertices, $\mathbf{d}$ is the vector of all depths d_i, and $\mathbf{B}_i$ contains the barycentric coordinates of each 3D points, as in Eq. 2 but rearranged to account for vertex order in the complete mesh.

In the absence of additionnal constraints, the surface can be scaled and still reproject at the same place. This can be avoided by imposing inextensibility constraints to recover the surface whose edge lengths are the same as those of the reference shape. However, as illustrated by Fig. 2, such constraints are violated when folds appear between mesh vertices. It is therefore truer to reality to replace the inextensibility constraints by constraints that allow the vertices to come closer to each other, but not to move further apart than their geodesic distance. For all pairs of neighboring vertices $\mathbf{v}_j$ and $\mathbf{v}_k$, we therefore write

$$\|\mathbf{v}_k - \mathbf{v}_j\| \leq l_{j,k} \ , \tag{3}$$

where $l_{j,k}$ is the geodesic distance between the vertices.

This prevents the surface from expanding but not from shrinking to a single point. However, this can easily be remedied by exploiting the fact that, in the perspective camera model, the lines-of-sight are not parallel. Thus the largest distance between two points is reached when the surface is furthest away from the camera. Therefore, a nontrivial reconstruction can be obtained by solving the problem

$$\begin{aligned} &\underset{\mathbf{X},\mathbf{d}}{\text{maximize}} \ \sum_{i=1}^{N_c} d_i \\ &\text{subject to} \ \ \mathbf{B}_i\mathbf{X} = d_i\mathbf{s}_i \ , \ 1 \leq i \leq N_c \ , \\ &\qquad\qquad \|\mathbf{v}_k - \mathbf{v}_j\| \leq l_{j,k} \ , \ \forall (j,k) \in \mathcal{E} \ , \end{aligned} \tag{4}$$

where $\mathcal{E}$ is the set of all mesh edges. This maximization of a linear criterion under linear and quadratic constraints is a convex problem that can be easily solved using standard mathematical routines [4].

3.2. Dealing with Image Noise

Whereas, given perfect correspondences, 3D surface points are completely defined by their depth, they should be allowed to move away from the lines-of-sight if the locations $\mathbf{q}_i$ are inaccurate. To this end, rather that forcing $\mathbf{p}_i$ to lie on its line-of-sight and maximizing d_i, we consider its projection on the line-of-sight $\mathbf{s}_i$, which can be computed as

$$\begin{aligned} \tilde{d}_i &= \mathbf{p}_i^T\mathbf{s}_i \ , \\ &= \mathbf{X}^T\mathbf{B}_i^T\mathbf{s}_i \ . \end{aligned} \tag{5}$$

Replacing d_i by $\tilde{d}_i$ in the problem of Eq. 4 yields the new optimization problem

$$\begin{aligned} &\underset{\mathbf{X}}{\text{maximize}} \ \sum_{i=1}^{N_c} \mathbf{X}^T\mathbf{B}_i^T\mathbf{s}_i \\ &\text{subject to} \ \ \|\mathbf{v}_k - \mathbf{v}_j\| \leq l_{j,k} \ , \ \forall (j,k) \in \mathcal{E} \ . \end{aligned} \tag{6}$$

We cannot, however, obtain a meaningful solution by simply solving this problem because nothing forces the 3D point projections to remain close to their corresponding image locations. Therefore, we need to introduce a term that explicitly penalizes bad reprojections, and use the formulation introduced in [20]: Enforcing correct reprojection can be achieved by minimizing $\|\mathbf{M}\mathbf{X}\|$ where $\mathbf{M}$ is a matrix that depends on the image locations and barycentric coordinates of the correspondences. More specifically, $\mathbf{M}$ is formed by concatenating the individual projection equations written as

$$\begin{bmatrix} b_i^j\mathbf{H} & b_i^k\mathbf{H} & b_i^l\mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{v}_j \\ \mathbf{v}_k \\ \mathbf{v}_l \end{bmatrix} = \mathbf{0} \ , \tag{7}$$

with

$$\mathbf{H} = \mathbf{A}_{\mathbf{2\times 3}} - \begin{bmatrix} u_i \\ v_i \end{bmatrix} \mathbf{A}_{\mathbf{3}} \ , \tag{8}$$

where $\mathbf{A}_{\mathbf{2\times 3}}$ are the first two rows of $\mathbf{A}$, and $\mathbf{A}_{\mathbf{3}}$ is the third one.

In the end, we therefore recover the shape by solving the problem

$$\begin{aligned} &\underset{\mathbf{X}}{\text{maximize}} \;\; w_d \sum_{i=1}^{N_c} \mathbf{X}^T \mathbf{B}_i^T \mathbf{s}_i - \|\mathbf{M}\mathbf{X}\| \\ &\text{subject to} \;\; \|\mathbf{v}_k - \mathbf{v}_j\| \leq l_{j,k} \;,\; \forall (j,k) \in \mathcal{E} \;, \end{aligned} \tag{9}$$

where w_d is a weight that controls the relative influence of depth maximization and image error minimization. In practice, we set w_d to 2/3 because computing depths involves $3N_c$ values against $2N_c$ projection equations. This optimization problem remains convex and can be solved by introducing a slack variable [4].

An alternative to this formulation would have been to use the L_∞-norm as suggested in [10]. That approach involves finding a solution for which all reprojection errors are smaller than a threshold that is iteratively decreased. We experimented with it in our framework. However, because we simultaneously maximize depths, large thresholds allowed incorrect deformations that prevented the process from converging towards a meaningful solution.

In addition to noise, correspondences may include gross errors. To remove them, we implemented an iterative procedure that decreases a radius inside which correspondences are considered as inliers. In practice, we initialize this radius to 50 pixels and divide it by 2 at every iteration. Furthermore, at each iteration, each valid correspondence equation is assigned a weight w_i computed as

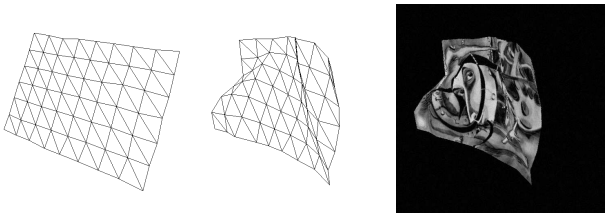
$$w_i = \exp\left(-\frac{e_i}{\text{median}(e_j \;,\; 1 \leq j \leq N_{in})}\right) \;, \tag{10}$$

where e_i is the reprojection error of correspondence i, and N_{in} is the number of inliers. This made our approach robust to noise and outliers.

4. Using Deformation Models

In the previous section, we presented an approach to reconstruct deformable surfaces from a single image given correspondences between that image and a reference image with a known shape. Our algorithm is robust to noise and outliers, but requires matches over the whole surface to correctly reconstruct all of it. In practice, such correspondences can only be obtained if the surface is consistently well textured. Since this rarely is the case, we now introduce models that supply the missing information while allowing us to retain our convex formulation.

In this section, we first present a linear local deformation model that constrains the poorly-textured parts of the surface to assume meaningful shapes. This allows us to reconstruct surfaces from sparse sets of correspondences, which is the best we can do given only one image. If we are instead given a short sequence, such as 3 consecutive video frames, we can compute the shape in each frame simultaneously and exploit the temporal consistency of motion to prevent jitter from one frame to the next. In other words, given a complete video sequence, we can reconstruct 3D shapes in each individual frame but the motion, while roughly correct, will look jerky. By contrast, if we compute it over batches of 3 frames, it will look much smoother, as will be seen in Section 5.



(a) (b) (c)

Figure 3. Synthetic data. (a) Undeformed mesh that was reconstructed from a motion capture system seen from the viewpoint used to generate correspondences. (b) Same mesh in the largest deformation of our data. (c) We textured the meshes to create images in which we established correspondences using SIFT.

4.1. Linear Local Models

Representing the shape of a non-rigid surface as a linear combination of basis vectors is a well-known technique. Such a deformation basis can be obtained by modal analysis [17, 16], from training data [7, 3], or directly from the images [27, 1, 12, 24, 26]. Here, we follow a similar idea, but, rather than introducing a single model for the whole surface, we represent the deformation of each local patch as a linear combination of modes. Not only does this yield a more flexible global model, but it also lets us explicitly account for the fact that parts of the surface are much less textured than others and should therefore rely more strongly on the deformation model. This would not be possible with a global representation, which would either penalize complex deformations excessively, or allow the poorly textured regions to assume unlikely shapes.

Let $\mathbf{X}_i$ be the x-, y-, z-coordinates of an $N_l \times N_l$ square patch of the mesh. We model the variations of $\mathbf{X}_i$ as a linear combination of N_m modes, which we write in matrix form as

$$\mathbf{X}_i = \mathbf{X}_i^0 + \Lambda \mathbf{c}_i \;, \tag{11}$$

where $\mathbf{X}_i^0$ represents the coordinates of the patch in the reference image, Λ is the matrix whose columns are the modes, and $\mathbf{c}_i$ is the corresponding vector of modes weights. In practice, the columns of Λ contain the eigenvectors of the training data covariance matrix, computed by performing Principal Component Analysis on a set of deformed 5×5 meshes that were obtained by simulating inextensible deformations. Furthermore, to deal with arbitrarily complex local deformations, we use *all* N_l^2 modes.

The standard approach when using a linear model is to replace the original unknowns by the modes weights. However, since we model the global surface with overlapping local patches, doing so would not guarantee that the shapes predicted by the weights associated to two such patches are consistent. Fortunately, since the deformation modes are orthonormal, the coefficients $\mathbf{c}_i$ of Eq. 11 can be directly computed from $\mathbf{X}_i$ as $\mathbf{c}_i = \Lambda^T \left(\mathbf{X}_i - \mathbf{X}_i^0\right)$. We therefore use the same global surface unknown $\mathbf{X}$ as before and encourage its patches to follow our linear local model. Since we use all the modes, this can be done by simply penalizing

$$\left\|\Sigma^{-1/2}\mathbf{c}_i\right\| = \left\|\Sigma^{-1/2}\Lambda^T\left(\mathbf{X}_i - \mathbf{X}_i^0\right)\right\| \qquad (12)$$

which measures how far the $\mathbf{c}_i$, and therefore $\mathbf{X}_i$, are from the training data, and where Σ is a diagonal matrix that contains the eigenvalues associated with the eigenvectors in Λ. We can then define the global regularization term

$$E_r(\mathbf{X}) = \sum_{i=1}^{N_p} w^i \left\|\Sigma^{-1/2}\Lambda^T\left(\mathbf{X}_i - \mathbf{X}_i^0\right)\right\| \ , \qquad (13)$$

by summing the measure of Eq. 12 over all N_p overlapping patches in the mesh. w^i is a weight designed to account for the fact poorly-textured areas should rely more strongly on the model than well-textured ones. In other words, it should be inversely proportional to the number of correspondences. We define it as

$$w^i = \exp\left(-\frac{N_c^i}{\text{median}(N_c^k > 0 \ , \ 1 \leq k \leq N_p)}\right) \ , \qquad (14)$$

where N_c^j is the number of matches in patch j.

Since this new regularization term has a quadratic formulation similar to the one used for projection equations, we can include it in the convex optimization problem of Eq. 9, which yields the new problem

$$\begin{aligned} &\underset{\mathbf{X}}{\text{maximize}} \quad E_f(\mathbf{X}) - w_r E_r(\mathbf{X}) \qquad (15)\\ &\text{subject to} \quad \|\mathbf{v}_k - \mathbf{v}_j\| \leq l_{j,k} \ , \ \forall (j,k) \in \mathcal{E} \ , \end{aligned}$$

where $E_f(\mathbf{X})$ contains the depths and correspondence terms of Eq. 9. w_r controls the amount of regularization we want to impose and its exact value has relatively little influence on the final result as long as it is large enough to have a noticeable effect.

Note that our linear local models are in the same spirit as those introduced in [21], but without having to explicitly introduce either additional latent variables or a sophisticated non-linear model.

4.2. Motion Model

In presence of a video sequence, or of several consecutive images, motion can also act as a reliable cue to reconstruct deformable surfaces. Indeed, we expect the deformations of the surface between consecutive frames to be coherent. We can therefore use this information to link the shapes

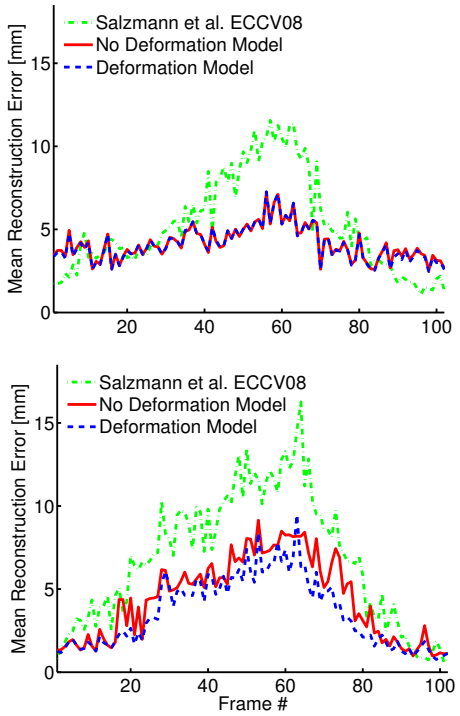


Figure 4. Comparison of our approach with the one proposed in [20] on synthetic data. Top: We sampled the facets of 3D meshes reconstructed from an optical motion capture system to create correspondences to which we added gaussian noise with variance 5. We plot the mean vertex-to-vertex reconstruction errors as a function of time for the method of [20], and our approach with and without using a deformation model, which, in this case, makes almost no difference. Bottom: We textured the 3D meshes and projected them to synthesize images from which we extracted SIFT correspondences. We plot the same errors as before. Note that our approach performs significantly better in the middle part of the sequence, which corresponds to the largest deformations.

in three images with a second order motion model. To this end, we minimize the error between the model's prediction and the true motion, which can be written as

$$E_m(\mathbf{X}^{t-1}, \mathbf{X}^t, \mathbf{X}^{t+1}) = \left\|\mathbf{X}^{t-1} - 2\mathbf{X}^t + \mathbf{X}^{t+1}\right\| \ , \qquad (16)$$

where $\mathbf{X}^t$ is the vector of mesh vertices at time t. Since this again involves a similar quadratic formulation as before, we can introduce it in our convex optimization problem, which becomes

$$\begin{aligned} &\underset{\mathbf{X}^{t-1},\mathbf{X}^t,\mathbf{X}^{t+1}}{\text{maximize}} \sum_{\delta=-1}^{1} E_t(\mathbf{X}^{t+\delta}) - w_m E_m(\mathbf{X}^{t-1}, \mathbf{X}^t, \mathbf{X}^{t+1}) \qquad (17)\\ &\text{subject to } \|\mathbf{v}_k^{t+\delta} - \mathbf{v}_j^{t+\delta}\| \leq l_{j,k} \ , \ \forall (j,k) \in \mathcal{E} \ , \\ &\qquad\qquad \delta \in \{-1, 0, 1\} \ , \end{aligned}$$

where $E_t(\mathbf{X}^t)$ is the global objective function for a single frame given in the optimization problem of Eq. 15, and w_m sets the influence of the motion model. In the experiments where we used the motion model, w_m was set to 100.

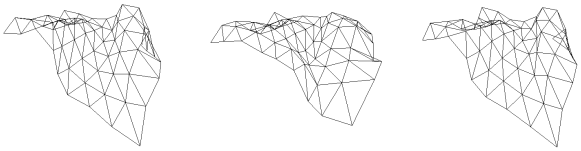


Figure 5. Visual comparison of the largest deformation of the synthetic sequence. From left to right: Ground-truth mesh, mesh recovered with the approach proposed in [20], mesh recovered with our approach using a deformation model. Note that our approximation is better than that of the earlier method.

5. Experimental Results

We now present results obtained on synthetic and real data by solving the optimization problem of Eqs. 9, 15, or 17 depending on whether we used a model or not. To this end, we used the matlab SeDuMi package [22], that effectively solves convex optimization problems.

5.1. Synthetic Data

We applied our approach to synthetic data to quantitatively evaluate its performance, and to compare it against a state-of-the-art technique. To make our experiments as realistic as possible, we obtained 3D meshes, such as those of Fig. 3(a,b), by deforming a flexible piece of cloth in front of an optical motion capture system. We then created correspondences by randomly sampling the barycentric coordinates of the mesh facets and projecting them with a known camera. We added zero-mean gaussian noise with variance 5 to the image locations. In Fig. 4(a), we compare the results of our technique with those obtained with the method proposed in [20]. We plot the mean vertex-to-vertex distance between the reconstructed mesh and the ground-truth one. In Fig. 5, we visually compare the results of both approaches for the largest deformation of the sequence. Note that our approach performs better both with and without using the deformation models. To even more accurately simulate real data, we textured the meshes and generated images, such as the one of Fig. 3(c), with uniform intensity noise in the range $[-10, 10]$. We then obtained correspondences by matching SIFT features between a reference image and the input images. Fig. 4(b) depicts the same errors as before computed from these correspondences. All results presented above were obtained from single images, since enough correspondences could be established, and, therefore, the motion model of Section 4.2 brought no improvement. Finally, we tested the robustness of our approach to outliers by assigning random image locations to a given percentage of the correspondences. In Fig. 6, we plot the mean reconstruction error over the sequence as a function of the outlier rate with and without using the deformation model. In this case, the motion model proved helpful to further improve the results, particularly in the case when no deformation model was used. As can be observed from the plots, our method is robust to up to 40% of outliers.

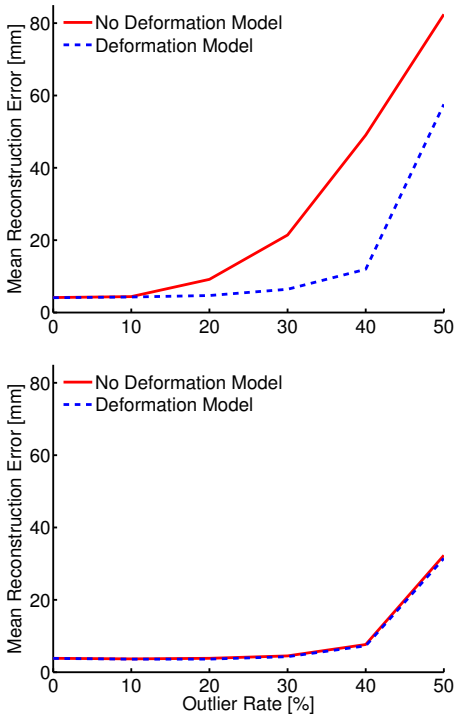


Figure 6. We added outliers to our synthetic correspondences and computed the shapes with (red) and without (blue) deformation models. We plot the average over the sequence of the vertex-to-vertex mean distances as a function of outlier rate. Top: Errors without using our motion model. Bottom: Smaller errors using it.

5.2. Real Images

We tested our approach on real images taken with a 3-CCD DV camera. We recovered the deformations of flexible objects such as the cloth of Fig. 7, the cushion of Fig. 8, and the t-shirt of Fig. 9. Due to the partial lack of texture and the possible mismatches generated by SIFT, these results were computed using the local deformation models. As a consequence of having poor correspondences, parts of the surface are sometimes not reconstructed absolutely correctly. However, thanks to our local deformation models, their shape remains meaningful. In each one of the figures, we show the mesh recovered using the motion model overlaid on the input image, the same mesh seen from a different viewpoint, and the reconstruction obtained without using the motion model. While, from static images, the meshes obtained with and without using the motion model look very similar, it can be seen from the videos that the motion model greatly stabilizes the results. In Fig. 7, we also show the reconstruction obtained by using the technique of [20]. As expected, it oversmoothes the sharp folds whereas our method yields more accurate reconstructions.

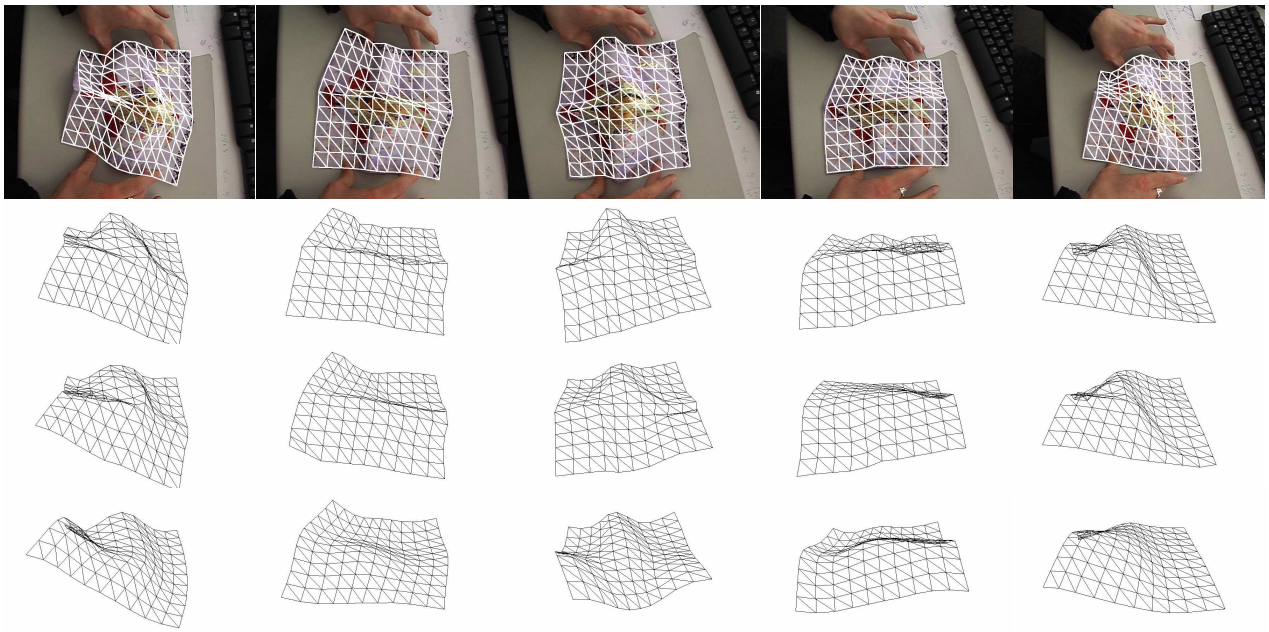


Figure 7. Reconstruction of a deforming cloth. From top to bottom: Mesh recovered using the motion model overlaid on the original image, same mesh seen from another viewpoint, mesh recovered without using the motion model, mesh recovered with the method of [20]. Note that their method oversmoothes the sharp folds whereas ours yields more accurate reconstructions.

6. Conclusion

In this paper, we have presented a convex formulation to the problem of recovering the 3D shape of sharply folding surfaces. Because the Euclidean distance between two surface points may decrease when folds appear, the usual distance equality constraints are only adapted to reconstruct smoothly deforming surfaces. We have therefore introduced inequality constraints that prevent points from moving further apart than their true geodesic distance, but allow them to come closer to each other. Maximizing the distance of surface points to the camera under these constraints, in conjunction with local deformation models if necessary, has proved effective to recover the complex deformations of flexible materials from relatively sparse, noisy correspondences.

In future work, we will seek to remove the requirement for a reference image in which we know the shape and, instead, exploit temporal motion consistency more thoroughly. More specifically, the frame-to-frame motion of individual mesh facets can be recovered from correspondences [28] but the estimates are bound to be noisy. However, considering all mesh facets simultaneously over short sequences and imposing local deformation models such as the ones of Section 4.1 will give rise to equations that are formally very similar to the ones presented in this paper and should therefore be solvable in a similar manner.

Acknowledgements

We would like to thank Prof. A. Shokrollahi for his useful suggestions which initiated this work.

References

[1] A. Bartoli and S. Olsen. A Batch Algorithm For Implicit Non-Rigid Shape and Motion Recovery. In *ICCV Workshop on Dynamical Vision*, 2005.

[2] K. S. Bhat, C. D. Twigg, J. K. Hodgins, P. K. Khosla, Z. Popovic, and S. M. Seitz. Estimating cloth simulation parameters from video. In *ACM SCA*, 2003.

[3] V. Blanz and T. Vetter. A Morphable Model for The Synthesis of 3–D Faces. In *ACM SIGGRAPH*, 1999.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] M. Brand. Morphable 3d models from video. *CVPR*, 2001.

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

[7] T. Cootes, G. Edwards, and C. Taylor. Active Appearance Models. In *ECCV*, 1998.

[8] A. Ecker, A. D. Jepson, and K. N. Kutulakos. Semidefinite programming heuristics for surface reconstruction ambiguities. In *ECCV*, 2008.

[9] N. Gumerov, A. Zandifar, R. Duraiswami, and L. Davis. Structure of Applicable Surfaces from Single Views. In *ECCV*, 2004.

[10] F. Kahl. Multiple view geometry and the L_∞-norm. In *ICCV*, 2005.

[11] J. Liang, D. DeMenthon, and D. Doermann. Flattening curved documents in images. In *CVPR*, 2005.

[12] X. Llado, A. Del Bue, and L. Agapito. Non-rigid 3D Factorization for Projective Reconstruction. In *BMVC*, 2005.

[13] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 20(2):91–110, 2004.

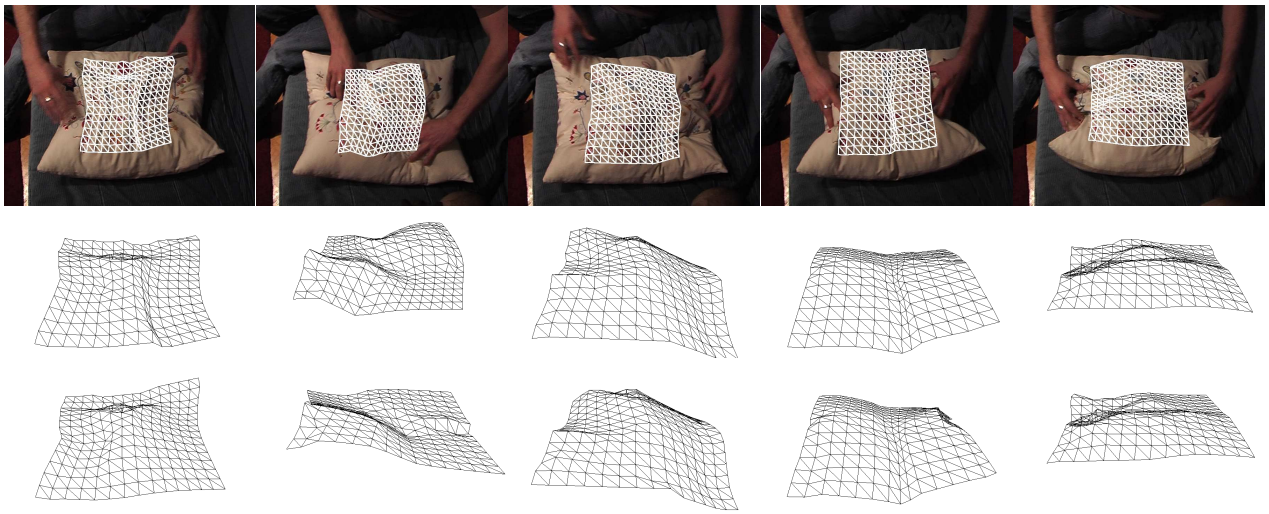


Figure 8. We recovered several complex deformations of a relatively poorly-textured cushion. Top: Mesh recovered using the motion model overlaid on the original image. Middle: Same mesh seen from a different viewpoint. Bottom: Mesh recovered without using the motion model.

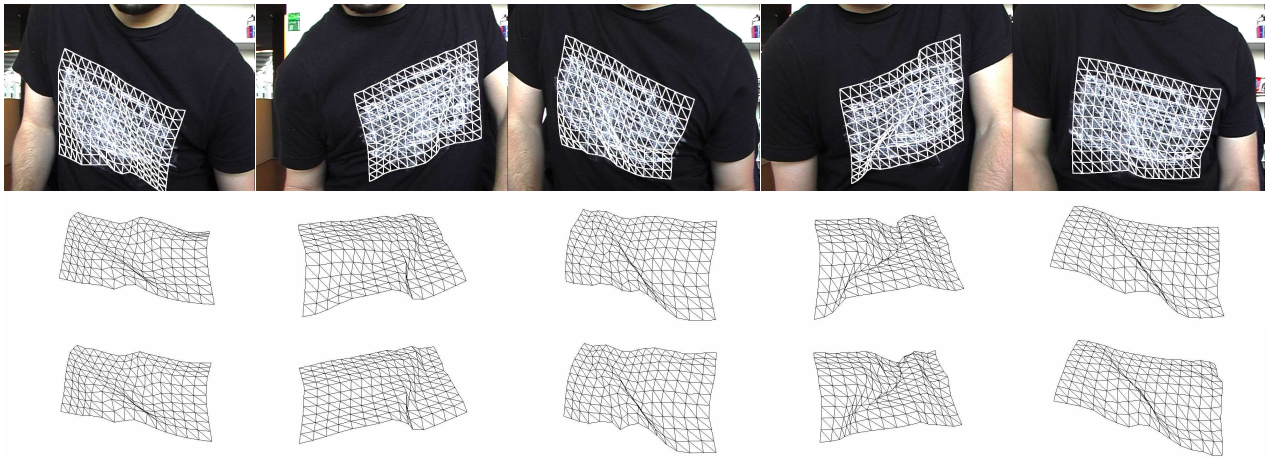


Figure 9. Reconstruction of a deforming t-shirt. Top: Mesh recovered using the motion model overlaid on the original image. Middle: Same mesh seen from a different viewpoint. Bottom: Mesh recovered without using the motion model.

[14] T. McInerney and D. Terzopoulos. A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking. In *ICCV*, 1993.

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

[16] C. Nastar and N. Ayache. Frequency-based nonrigid motion analysis. *PAMI*, 18(11):1067–1079, 1996.

[17] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *PAMI*, 13(7):715–729, 1991.

[18] M. Perriollat and A. Bartoli. A quasi-minimal model for paper-like surfaces. In *CVPR BenCos Workshop*, 2007.

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

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

[21] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3d shape recovery. In *CVPR*, 2008.

[22] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, 1999.

[23] D. Terzopoulos, J. Platt, A. Barr, and K. Fleicher. Elastically Deformable Models. *ACM SIGGRAPH*, 1987.

[24] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 30(5):878–892, 2008.

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

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

[27] J. Xiao and T. Kanade. Uncalibrated perspective reconstruction of deformable structures. In *ICCV*, 2005.

[28] Z. Zhang, and A. R. Hanson Scaled Euclidean 3D reconstruction based on externally uncalibrated cameras. In *Symposium on Computer Vision*, 1995.