Surface Deformation Models for Nonrigid 3D Shape Recovery

Mathieu Salzmann, Julien Pilet, Slobodan Ilic, and
Pascal Fua, *Member*, *IEEE*

Abstract—Three-dimensional detection and shape recovery of a nonrigid surface from video sequences require deformation models to effectively take advantage of potentially noisy image data. Here, we introduce an approach to creating such models for deformable 3D surfaces. We exploit the fact that the shape of an inextensible triangulated mesh can be parameterized in terms of a small subset of the angles between its facets. We use this set of angles to create a representative set of potential shapes, which we feed to a simple dimensionality reduction technique to produce low-dimensional 3D deformation models. We show that these models can be used to accurately model a wide range of deforming 3D surfaces from video sequences acquired under realistic conditions.

Index Terms—3D shape recovery, deformation model, nonrigid surfaces.

1 Introduction

Without a strong model, 3D detection and shape recovery of a nonrigid surface from video sequences is a severely underconstrained problem. Such models have been built for specific object classes such as faces [4], but not for generic surfaces. These are typically represented as triangulated meshes with potentially many vertices to achieve the desired level of accuracy, which implies many degrees of freedom and a potentially hard to solve optimization problem when trying to fit the model to noisy image data.

Physics-based models have been extensively investigated as a potential answer to this problem. They have been shown to be excellent at fitting noisy image data and handling highly deformable 3D objects [16], [18], [5], [7], [15]. They incorporate regularization terms that implicitly or explicitly reduce the number of degrees of freedom. However, to the best of our knowledge, the effectiveness of such models has not yet been demonstrated on monocular sequences of deformable 3D surfaces such as those of Figs. 1, 2, and 3.

Here, we describe an approach to creating sufficiently low-dimensional models of deformable 3D surfaces that can be represented as 3D meshes without holes. Given the possibly nonplanar rest shape of the mesh, constraining its edges to retain their original lengths implies that all possible deformations are entirely specified by a small subset of the angles between its facets. This implies that the manifold of all possible deformations can be effectively sampled by randomly setting a limited number of angles. This, in turn, lets us generate a database of deformed shapes with identical topologies and use a standard dimensionality reduction technique to produce the low-dimensional 3D deformation models that we need for tracking and detection purposes.

The inextensible triangulations we use can be thought of as polyhedra made of metal plates and whose edges have been replaced by hinges. Such polyhedra have been extensively used in the classroom to teach elementary geometry but not in our field. Nevertheless, they can assume a surprisingly large range of shapes and, therefore, produce representative shape databases. Thus, as shown in Figs. 1, 2, and 3, our approach can be used to recover the deforming 3D shape of such diverse objects as a T-shirt, a sheet of paper, a sail, or an elastic surface. Even though these have very different physical properties, our model has the right degrees of freedom to capture their deformations, even when they are not isometric [10] and to take full advantage of available image information. In fact, if a textured 3D model of the object in a reference position is available, our system becomes completely automatic: Neither deformation model generation nor detection and tracking require any manual intervention.

We therefore view the contribution of this paper as twofold: On the theoretical side, we propose an approach to creating low-dimensional surface deformation models. On the practical side, we show that these models can be effectively used to pool noisy image information, thus letting us accurately model a wide range of 3D surfaces.

2 Related Work

Detecting and tracking 3D surface deformations in monocular video sequences requires deformable models to constrain the search space and make the problem tractable.

Such models have been created for feature point-based structure from motion [28], [27], [13], [29], [1] by tracking feature points and using them to learn both shape and motion. While effective, these algorithms are not designed to exploit other sources of image information than feature points or to use known surface properties to recover the shape far away from those feature points. This typically requires explicit surface modeling using as few degrees of freedom as possible.

One way to achieve this is to only consider the motion of a few control points. Free-form deformations [22], [8], [17] are a good example of this kind of approach, but there is currently no automated way to create appropriate sets of deformation modes or control points. Physics-based models are potentially more generic. The original ones [11] were 2D and have been shown to be effective for 2D deformable surface registration [2]. They were soon adapted for 3D surface modeling purposes by using deformable superquadrics [26], [16], triangulated surfaces [5], or thin-plate splines [15]. In this framework, modeling generic 3D surfaces often requires many degrees of freedom that are coupled by regularization terms. In practice, this coupling implicitly reduces the number of degrees of freedom, which makes these models robust to noise and is one of the reasons for their immense popularity. This reduction can also be explicitly achieved via modal analysis [18], [5], [7]. In our own cartographic work [9], we represented 3D surfaces as hexagonal meshes that deformed to minimize an energy that was the sum of an image-data term and a quadratic regularization term. This proved very effective for cartographic modeling, which is essentially 2.5D as opposed to fully 3D. But, it has turned out to be insufficient for robust monocular video-based tracking of deformable surfaces.

Since accurately capturing the physics of deformable surfaces in a dynamical model is difficult, example-based approaches are an attractive alternative. They involve creating a database of representative shapes and using them in conjunction with a dimensionality reduction technique to learn a low-dimensional model. Active appearance models [6] pioneered this approach in the 2D case and have since been extended to 3D [14]. Morphable models [4] rely on the same philosophy to build 3D face models: The database is made of 3D meshes that were fitted to laser scans and then registered to each other. Similar approaches were successfully used to learn models of articulated motion [3], [24]. However, in all these cases, gathering and registering enough examples to build a meaningful database represented a very significant amount of work. The difficulties involved in creating the databases have limited the spread of these example-based approaches.

3 Surface Deformation Models

One of the simplest ways to model a deformable surface is to represent it as a triangulated surface parameterized in terms of its

- *M. Salzmann, J. Pilet, and P. Fua are with the Computer Vision Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland.*
E-mail: {Mathieu.Salzmann, Julien.Pilet, Pascal.Fua}@epfl.ch.
- *S. Ilic is with the Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, 10587 Berlin, Germany. E-mail: Slobodan.Ilic@tu-berlin.de.*

Manuscript received 3 Mar. 2006; revised 14 June 2006; accepted 8 Nov. 2006; published online 18 Jan. 2007.
Recommended for acceptance by C. Kambhamettu.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0203-0306. Digital Object Identifier no. 10.1109/TPAMI.2007.1080.

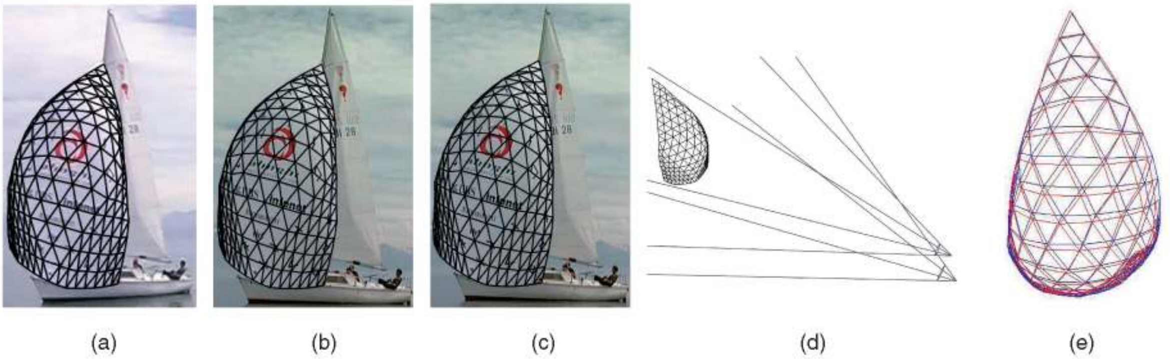

Fig. 1. Tracking a spinnaker with either one or two cameras. (a) and (b) Two synchronized images from independently moving cameras with recovered spinnaker reprojection. (c) Tracking using only one camera. Note that once reprojected on the images, the results are almost indistinguishable. (d) Three-dimensional results with two cameras. Both camera positions are also retrieved. (e) Superposed 3D shapes retrieved using either one (red) or two (blue) cameras. Note that both shapes are very similar, which indicates that the deformation model provides a good approximation when data are missing. See video submitted as supplementary material, which can be found at http://computer.org/tpami/archives.htm.

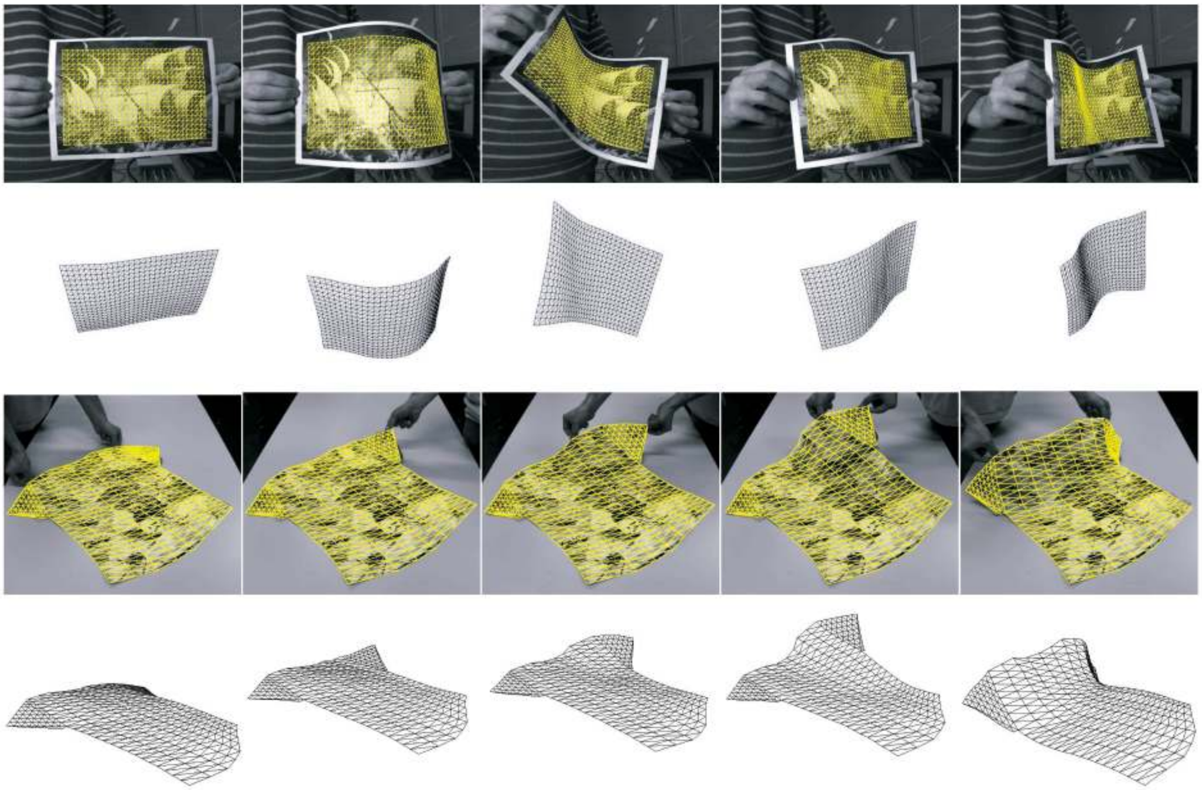

Fig. 2. Tracking a deforming sheet of paper and T-shirt. In both cases, we show the deformed 3D mesh overlaid on the original images in the top row and then seen from a different viewpoint in the bottom row.

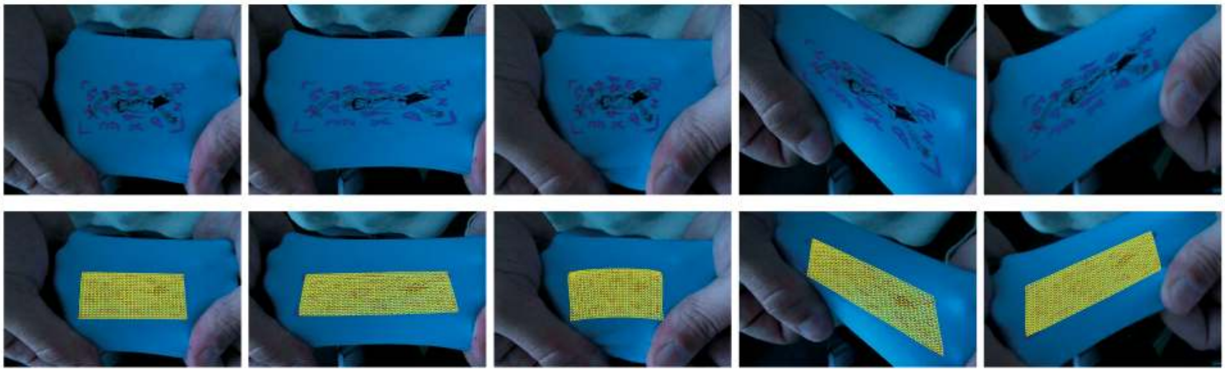

Fig. 3. Tracking an extensible surface undergoing anisotropic deformations. In the top row, we show the original images and, in the bottom row, we overlay the recovered 3D grid that stretches appropriately.

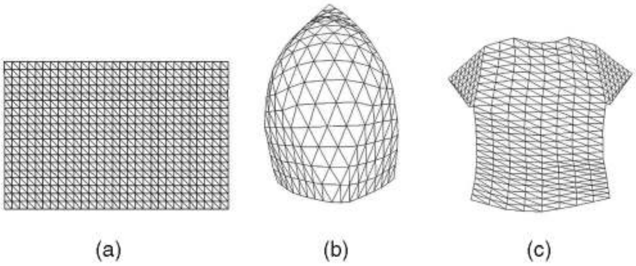

Fig. 4. Hexagonal triangulations. (a) Rectangular mesh used to model the piece of paper. (b) Triangular mesh used to model the spinnaker. (c) Stitching a rectangular patch for the body part and two triangular ones for the sleeves lets us model the T-shirt.

vertex coordinates. This parameterization, however, does not account for the fact that, in a real surface, the vertices cannot move independently from one another. By contrast, if we constrain the triangulation edges to retain their original length, the number of degrees of freedom (dofs) decreases very significantly, which lets us

1. represent the shape using few parameters,
2. create a representative sample of possible shapes, and
3. perform dimensionality reduction.

This results in a low-dimensional model whose dimension is independent of that of the meshes used to create it, but still captures the main deformation modes. Using PCA as our dimensionality reduction technique naturally yields not only bending modes but also rigid motion and extension modes, which we can then choose to penalize or not. In some sense, this is similar to modal analysis where the object's behavior is described by superposing its natural strain and vibration modes [18], [5], [7]. However, unlike modal analysis, we do not require the kind of physical knowledge that building the appropriate stiffness matrix requires and we are not limited to small deformations around the position for which it has been computed.

3.1 Dofs of Inextensible Triangulations

We seek to characterize the number of dofs of a triangulation T —containing V 3D vertices, F facets, and E edges—that has a planar topology, which means it can be unfolded to a plane and has an actual boundary that can form an arbitrary polygon. In general, T has three dofs per vertex. However, forcing the edges to retain their length when the triangulation deforms, imposes one quadratic constraint per edge and the total number of degrees of freedom drops to $Dof = 3\,V - E$. Let E_b be its number of boundary edges and $E_i = E - E_b$ the number of interior ones. Since the E_b boundary edges each belong to only one facet whereas the E_i internal ones belong to two, we have $3\,F = 2\,E_i + E_b$. Furthermore, according to Euler's well-known formula, if T has no holes, $V + F - E = 1$. Substituting these expressions into $Dof = 3\,V - E$ yields

$$Dof = 3 + E_b. \qquad (1)$$

In other words, the number of degrees of freedom of an inextensible triangulation grows as the number of its boundary edges. In this work, we exploit this behavior in the case of regular hexagonal triangulations such as those of Figs. 4a and 4b, which can easily be stitched together to model more complex surfaces such as the T-shirt of Fig. 4c.

More specifically, the regular grid of Fig. 4a has $M \times N$ vertices, $E_b = 2(N-1) + 2(M-1)$ boundary edges and, therefore, $2\,M + 2\,N - 1$ degrees of freedom, which is much smaller than the $3\,MN$ it would have without the inextensibility constraints. Furthermore, this number of dofs include the six that correspond to a rigid motion and can be ignored for our purposes. The triangulation of Fig. 4b has N vertices per side and was built by recursively subdividing a single triangle. It has $N(N+1)/2$ vertices and $E_b = 3(N-1)$ boundary edges, which results in $3\,N$ dofs instead of $3N(N+1)/2$.

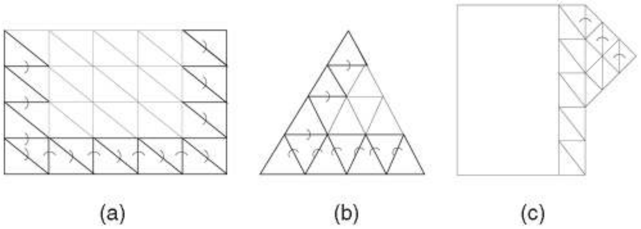

Fig. 5. Specifying the 3D shape of the rectangular mesh and subdvided triangle. (a) We fix the shape of the bottom row from left to right by rotating each facet with respect to its left neighbor. For each following row, we only need to set the angle between the left-most facet and the one below and the angle between the right-most facet and its left neighbor. (b) The angles between the facets of the bottom row are first set from left to right. For each upper row, only the angle of the first facet need be set. (c) Attaching two hexagonal patches together. Because the base of each triangular patch is attached to the body, only one single angle is required to fully specify their first row.

The T-shirt of Fig. 4c is modeled by combining a rectangular patch for the body part and two triangular ones for the sleeves. In this case, the number of dofs of the triangular patches is reduced because they have common edges with the rectangular patch. As a result, the total number of dofs resulting from assembling the triangular and rectangular patches is less than the sum of dofs of each patch taken separately.

3.2 Angle-Based Parameterization

Here, we show that the shape of a wide class of inextensible meshes can be parameterized in terms of a small number N_a of *determining angles* between its facets. We present procedures for choosing the N_a angles so that the number of degrees of freedom of (1) can be written as

$$Dof = N_a + 6, \qquad (2)$$

where the six degrees of freedom added to N_a represent the rigid motion.

3.2.1 Simple Triangulations

Let us first consider the $M \times N$ mesh of Fig. 4a. As shown in Fig. 5a, if we constrain the horizontal, vertical and diagonal edges to retain their original lengths, only the facets of the bottom row and the first and last facets of each upper row need be set to completely determine the shape of the grid. Each one of the remaining vertices can then be computed as the intersection of three spheres centered on previously computed vertices. It can be easily checked that this requires specifying $N_a = 2\,M + 2\,N - 7$ determining angles and the six degrees of freedom that fix the position and orientation of the first facet. This corresponds to the predicted total of $Dof = 2\,M + 2\,N - 1$ dofs derived in Section 3.1. In other words, the chosen subset of angles gives us a model with the right number of degrees of freedom.

In the case of the subdivided triangle with N vertices per side of Fig. 4b, we use the very similar construction depicted by Fig. 5b. The total number of determining angles is $N_a = 2(N-2) + (N-2) = 3\,N - 6$. To this number, we must add the six dofs required to fix the position and orientation of the first facet in space to get the expected total of $Dof = 3\,N$ dofs discussed in Section 3.1.

3.2.2 Complex Triangulations

As discussed in Section 3.1, we modeled the T-shirt of Fig. 4c by combining a rectangular patch for the body part and two triangular ones for the sleeves. We parametrize the rectangular patch as before. As shown in Fig. 5c, because the base of each triangular patch is attached to the body, only one single angle is required to fully specify their first row. The remaining rows of the triangles can then be specified as before, which results in the expected number of determining angles.

Note that this approach is very general and could be extended to any surface without holes that can be unfolded to a planar polygon of arbitrary shape: Any polygon can be triangulated without adding

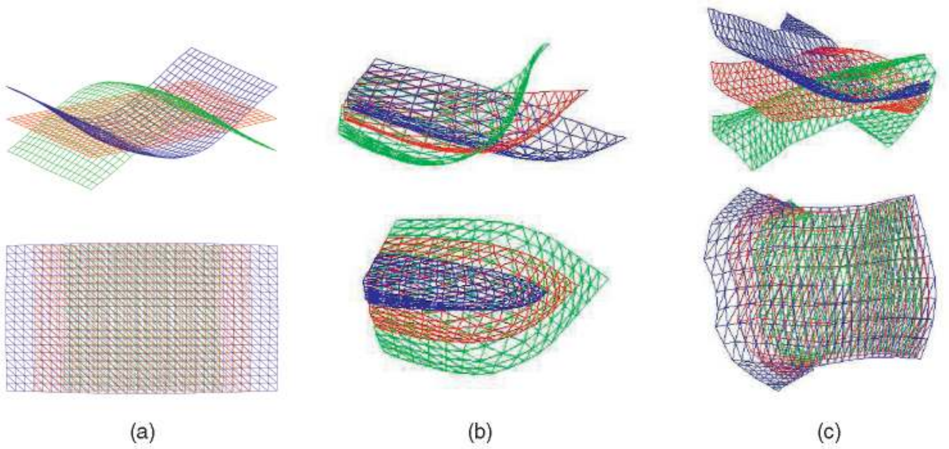

Fig. 6. Deformation modes of the meshes of Fig. 4. In all figures, $\bar{S}$, the average mesh, is shown in red. The other two are obtained by taking a single w_k to be nonzero. A positive value of that w_k yields the green mesh and a negative one the mesh shown in blue. (a) Bending and extension modes of (a) the flat rectangular mesh, (b) the triangular spinnaker, and (c) the T-shirt.

any interior vertex [20]. The dual graph of such a triangulation, that is, the graph connecting the centers of neighboring facets, cannot contain any cycle because such a cycle would have to enclose at least a vertex, which would then be an interior vertex. This implies that we can build the triangulation by sequentially inserting triangles in such a way that each new one, except the first, has a single common edge with one already present. Given this order, we can represent the individual triangles as hexagonal triangulations attached to each other and parameterize them as discussed above.

3.3 Dimensionality Reduction

The angle-based parameterization we introduced above reduces the number of parameters required to specify the shape of an inextensible mesh. However, it is not particularly well adapted to fitting surfaces to image data for several reasons. First, it imposes an arbitrary graph structure among the vertices and specifies the coordinates of *child* vertices as a function of those of *parent* vertices, which tends to degrade the performance of optimization algorithms. Second, computing the actual shape involves solving quadratic equations representing the intersection of three spheres, which is computationally expensive. Finally, its number of dofs still depends on the mesh resolution.

We therefore only use the angle-based parameterization as an intermediate representation that lets us sample the set of possible shapes by randomly drawing the angles from a uniform distribution between two bounds. For the rectangular mesh, the angles were drawn in the range $[-\pi/6, \pi/6]$ and, for the other cases, in the range $[-\pi/9, \pi/9]$. Since all the resulting deformed meshes have the same topology, we form a $3V$ vector for each one by concatenating the coordinates of its V vertices. By running PCA on these vectors and retaining only the first $N_c \leq Dof << 3V$ principal components, we can approximate the vector of coordinates of any mesh as

$$S = \bar{S} + \sum_{k=1}^{N_c} w_k S_k, \tag{3}$$

where $\bar{S}$ is the vector corresponding to an undeformed mesh, the S_k are the principal components or modes, and the w_k are weights that specify the surface shape. In other words, the shape of a mesh can now be expressed as a function of the vector $\Theta_S = \{w_1, \ldots, w_{N_c}\}$.

Fig. 6 depicts the influence of two of the most significant modes in the case of the meshes of Fig. 4. Giving weight to the first produces bending and, to the second, extension. The presence of extension modes may seem surprising since all the samples we used to learn the model are instances of the same inextensible mesh. However, given that the deformations are not linear when expressed in terms of 3D coordinates, there is no reason for the manifold of all resulting shapes to lie on a hyperplane. Intuitively, by using PCA, we consider the N_c-dimensional ellipsoid that includes this manifold without being limited to it. This produces not only extension modes but also rigid ones that we discard.

In practice, the presence of these extension modes makes the method more general: On one hand, if the surface whose deformations we seek to recover is truly inextensible, we can incorporate a term that prevents extension or shrinking into our optimization scheme. On the other hand, the presence of the extension terms lets us effectively model stretchable materials using a low-dimensional deformation model. In theory, it should be possible to remove those extension modes by replacing PCA by a nonlinear dimensionality reduction technique. However, we do not believe it will help much without using a database that is much closer to the true physics. This is because a nonlinear technique is very likely to force the model to stick much closer to the training data. In some sense, that would negate one of the strengths of our approach that does not require either accurate training data or precise knowledge of the physics, both of which are often hard to obtain.

4 Shape Recovery Framework

We outline here our approach to using our models to take full advantage of the available image information, acquired using one or more cameras, while ignoring erroneous data.

Recall from Section 3.3 that the shape of the mesh is controlled by the Θ_S vector of weights assigned to the PCA modes. To handle the potentially moving camera, or cameras, we introduce a vector of extrinsic parameters κ for each one and define the state vector

$$\Theta = (\kappa_1, \ldots, \kappa_C, \Theta_S)^T, \tag{4}$$

where C is the number of cameras being used. Note that this formulation can handle both one single camera and multiple cameras that may move with respect to each other.

We use the image data to write n_o observation equations of the form

$$O^t(\mathbf{x}_i, \Theta) = \epsilon_i, \ 1 \leq i \leq n_o, \tag{5}$$

where O^t is a differentiable objective function associated to image data of type t, $\mathbf{x}_i$ a data point, and ϵ_i an error term. Here, we consider the functions O^m, O^c, and O^e derived from model to image point correspondences, image to image point correspondences, and contour information, respectively.

4.1 Model to Image Correspondences

As shown in Figs. 7a and 7b, when a textured 3D model of the object in its rest position is available, we use a fast wide-baseline feature matching technique [12] to compute correspondences between surface 3D locations $\mathbf{c}$ and 2D image features. We define

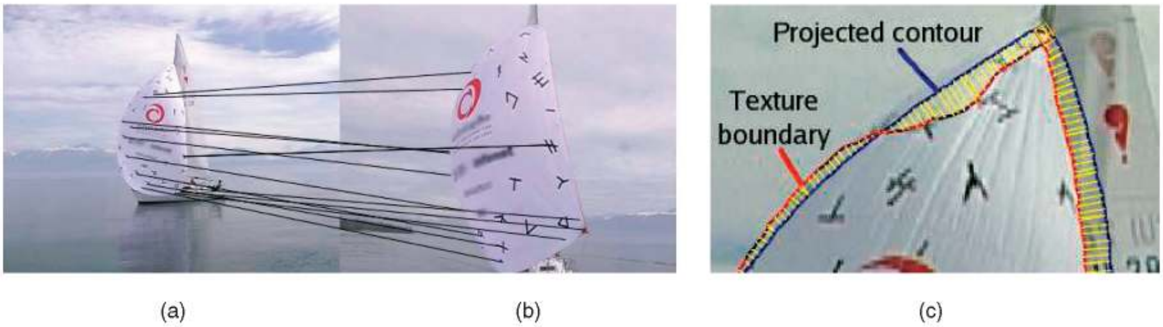

Fig. 7. Image data. (a) An image from an input sequence. (b) One of 15 images used to build a textured 3D model of the spinnaker. For our experiments, we added black scotch tape on the otherwise white parts of the sail to help our wide-baseline algorithm to find correspondences between model and input images such as those depicted by the black lines. (c) Contours detected as texture boundaries. Even though the boundary is not correct everywhere, thanks to the model and robust estimation we still recover the correct shape.

$O^m(\mathbf{c},\Theta)$ as the euclidean distance in the image plane between the projection of $\mathbf{c}$ and the corresponding image feature.

4.2 Image to Image Correspondences

Given a couple $\mathbf{u}_i = (p_i^1, p_i^2)$ of corresponding points in two different images of the surface found using the same technique as before [12], we define $O^c(\mathbf{u}_i,\Theta)$ as follows: We back-project p_i^1 to the 3D surface and reproject it to the second image. We then take $O^c(\mathbf{u}_i,\Theta)$ to be the euclidean distance in the image plane between this reprojection and p_i^2.

4.3 Boundary and Occluding Contours

As shown in Fig. 7c, given the last known shape of the target object, we predict the location of the projection boundaries and occluding contours. We then sample these 2D contours and look for the closest edge or texture boundary in the normal direction [23]. We take O^e to be the euclidean distance between the projection and the image edges.

As we saw in Section 3.3, a linear combination of principal components can result in a mesh that expands or shrinks. To model surfaces that do not stretch, we force edge lengths to remain constant by introducing a penalty term

$$E_D = \sum_{i=1}^{V} \sum_{v_j \in \mathcal{N}(v_i)} \left(\|v_i - v_j\| - L_{i,j} \right)^2, \tag{6}$$

where v_i is a vertex of the mesh, $\mathcal{N}(v_i)$ represents the set of all its neighbors, and $L_{i,j}$ is the initial edge length. Finally, we take the global objective function E we minimize to be

$$E = \frac{1}{2} \sum_{i=0}^{n_o} w_t \rho\left(\left\| O^{t_i}(\mathbf{x}_i,\Theta) \right\|^2, r \right) + w_D E_D, \tag{7}$$

where the w_t are weights associated to particular observation types and designed so that the derivatives of all observations are of commensurate magnitude, w_D is a user defined weight, and ρ a robust estimator whose radius of confidence r progressively decreases during the optimization. As discussed in [19], this schedule allows convergence from arbitrary starting positions. A small, or zero, w_D lets the mesh stretch or shrink. Note that besides the term that constrains the length, we introduce no other shape regularization term.

To demonstrate that our optimization process is well-posed and insensitive to initial conditions, we ran our system on synthetic data. We created randomly deformed versions of the rectangular mesh such as the ones shown in the first column of Fig. 8 and used them to produce large numbers of the synthetic model-to-image correspondences introduced above. For each test-run, we started from a different random initialization such as the ones of the second column of Fig. 8 and minimized the objective function of (7) using random subset of the correspondences. In the third column of the figure, we plot the median of the mean distances between the vertices of the recovered mesh to the correct one as a function of the number of correspondences that were used. More precisely, given a number n between 5 and 600, we picked four different subsets of n correspondences and ran the algorithm with 100 different initial shapes for each. As soon as enough correspondences are used, the algorithm consistently converges toward the correct solution.

5 Results

Here, we demonstrate the capabilities of our method using the four very different kinds of objects depicted by Figs. 1, 2, and 3. They cover a wide range of physical properties and the images have been acquired using ordinary camcorders. In all four cases, we first created a 3D textured model offline using one or more static images acquired independently of the videos.

We then optimized the criterion of (7) using model-to-image correspondences, correspondences with the previous image, and optionally silhouette information. This runs at between 0.3 and 2.5 frames/second when using 40 modes and $1,024 \times 768$ images, the faster rate being obtained when not using silhouettes. Recall from Section 4 that we use an optimization schedule that lets us start from arbitrary positions, which means that this does not require any manual intervention at runtime. As shown in the figures of this section, the resulting shapes are accurate enough for correct reprojection. However, enforcing temporal consistency only over image pairs might leave a residual 3D jittering motion across frames when creating videos. Therefore, to create those we supply as supplementary material, we eliminated this jitter by reoptimizing our criterion using the same observations as before, but over larger sets of eight overlapping frames, and enforcing temporal coherence by penalizing the second derivatives of all parameters.

We represent both the sheet of paper and the elastic surface, which was cut out of an inflatable balloon, of Figs. 2 and 3 as a 30×20 rectangular grid. We model the spinnaker using a 153-vertex triangle. The T-shirt mesh is made of two sleeves that are 45-vertex equilateral triangle attached to a 9×25 rectangular grid. We used 45 PCA modes to track the sheet of paper, 10 for the balloon, 40 for the spinnaker, and 50 for the T-Shirt.

The rest shape of the spinnaker is not planar and NorthSails gave us the CAD model that was used to design it, thus allowing us to fit a triangular mesh to it. This gave us the initial shape from which we computed the deformation modes. To create the textured model needed for automated runtime operation, we developed a software tool that allows us to manually supply a few image-to-model correspondences in images such as those of Fig. 9, which do *not* belong to the test video. By feeding these correspondences along with automatically detected silhouettes into the optimization framework of Section 4, we recover the spinnaker's shape into the images and, thus, a textured model. In the specific case depicted by Fig. 9, we only supplied 10 correspondences per image, which did not take long to do. In short, our deformation modes not only lead to robust

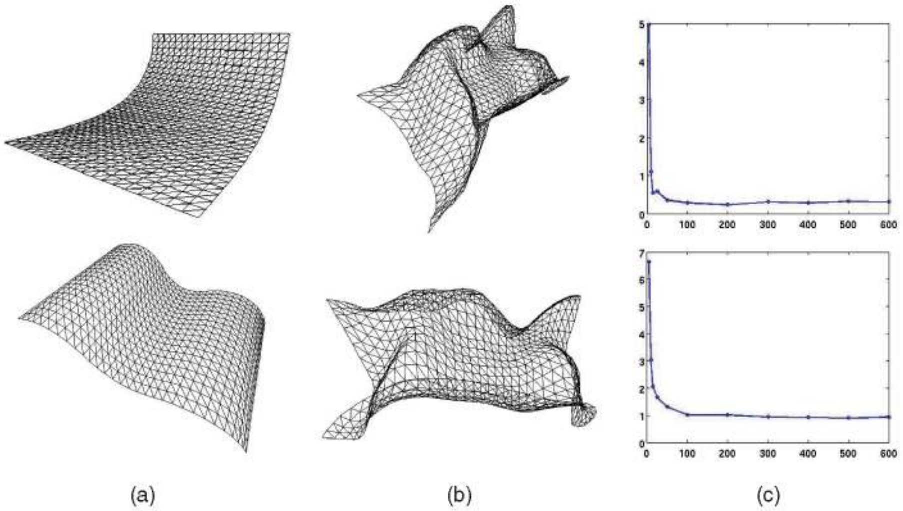

Fig. 8. Convergence using synthetic data. (a) Projections of the synthetic surfaces used as input for the optimization process. (b) Examples of initializations. (c) Median of the mean distances between the vertices of the recovered mesh and the synthetic surface as a function of the number of correspondences that were used. The measures are given as a percentage of the longest side of the initial rectangle. We did not draw error bars because, as soon as we used more than 50 matches, the first and third quartile of the mean distances are indistinguishable from the median.

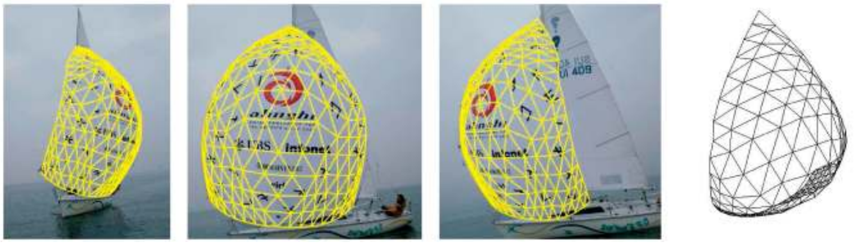

Fig. 9. Three-dimensional model of the spinnaker overlaid on the three images used to compute its reference shape and texture. In each image, we specified 10 correspondences with a CAD model of the spinnaker and used them, along with automatically detected silhouettes, to deform it. We assume that the spinnaker did not deform in these images because they were taken in quick succession by a chase boat.

and automated runtime operation but can also be used to limit the required amount of manual intervention during model building.

These experiments display several strengths of our method. First, as shown in the videos corresponding to the deforming sheet of paper of Fig. 2 and spinnaker of Fig. 1, our system is robust enough to process sequences of more than 1,500 frames acquired both indoors and outdoors without getting lost or drifting. When the image data is too weak, 3D shape recovery becomes temporarily less accurate but the system soon recovers.

The T-shirt example of Fig. 2 shows that even though the resulting PCA modes represent global deformations, we can still track very local ones, such as only one moving sleeve, by superposing these modes. Finally, in the case of the stretching surface of Fig. 3, we can see that not only global extension can be modeled, but also anisotropic stretching. This, again, is due to the fact that local deformations can be accurately described by appropriately superposing global modes.

All these results were generated using a single video sequence per object except in the spinnaker case where we used either one or two cameras. In this case, the two cameras were hand-held by two people on a chase-boat so that they move with respect to each other in an unpredictable fashion and do not form a stereo-rig in the usual sense of the term. Our framework is powerful enough to handle this case and to take full advantage of all the available information, even under such nonstandard conditions. As shown in Fig. 1, once reprojected on the images, the results are almost indistinguishable. Of course, because a single camera cannot see both sides of a curvy object, the quality of the 3D results is bound to be better when using two cameras looking from very different angles so as to see different parts of the object. However, the superposition of both 3D results in Fig. 1e shows that, in this case, the model approximates the hidden part well. This behavior is consistent over the whole video sequence.

Finally, to estimate the accuracy of our reconstructions, we acquired three videos of another sheet of paper using three calibrated and synchronized cameras. We used one to monocularly reconstruct the deforming shape using our method. We used the other two to triangulate the X,Y, and Z coordinates of 10 selected points by manually establishing correspondences every 10 frames of the 70 frame-long sequences. These points—the four corners of the sheet plus six additional ones spread over its surface—are depicted by Fig. 10d and were chosen to be representative of the whole surface. As shown in Fig. 10e, the largest average errors occur in the Z direction, which was to be expected since it is close to the viewing direction. In euclidean terms, this corresponds to the median errors of Fig. 10f, which are in the order of 1cm. This is quite small considering that this was achieved using a single camera that was approximately 1.5 meter away from the $29.7\,cm \times 21.0\,cm$ rectangular sheet of paper.

6 Conclusion

In this paper, we have developed an automated method for generating deformation models. We have shown that they are effective to recover the 3D shape of a variety of nonrigid surfaces undergoing large deformations. A key ingredient is an angle-based parameterization of inextensible meshes that has allowed us to create samples of possible shapes, on which we perform PCA to produce the low-dimensional models we use.

One limitation of our current approach is that those samples are not fair in the sense that all shapes are deemed equally probable and therefore similarly influence the model we derive from them. In future work, we will explore ways of making our models more accurate by weighting the samples according to how well they

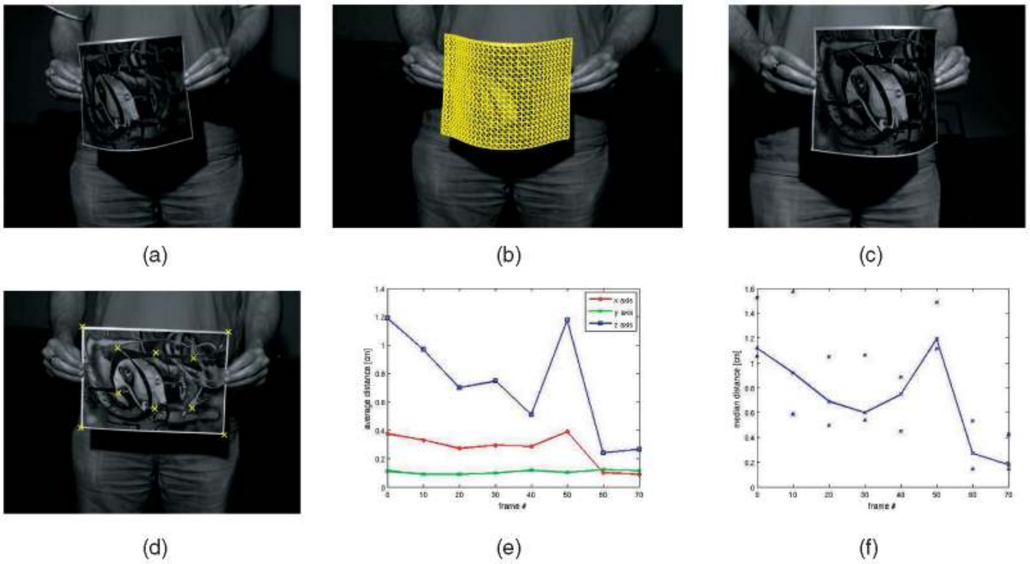

Fig. 10. Evaluating the accuracy of our approach. (a), (b), and (c) Images from videos acquired using three synchronized and calibrated cameras. Image (b) belongs to the video we used to monocularly reconstruct the 3D shape using our method and, then, reprojecting it into the image. (d) We triangulated the 3D coordinates of the 10 keypoints shown as crosses by manually establishing correspondences in images (a) and (c). (e) We repeated this operation every 10 frames and plot the average differences between the X, Y, and Z coordinates of those manually computed and those derived from our automated and monocular reconstruction. (f) We also computed the euclidean distances between the monocular reconstructions and the manually computed points, and plot their medians together with values at 25 percent and 75 percent.

conform to the true physics. This should yield improved shape databases. Given these, we will also investigate the use of nonlinear dimensionality reduction techniques such as Locally Linear Embedding (LLE) [21] or Isomap [25] to guarantee that the resulting deformations are truer to the physics than those PCA produces.

ACKNOWLEDGMENTS

This work was supported in part by the EU Vision Train project.

REFERENCES

[1] A. Bartoli and S.I. Olsen, "A Batch Algorithm for Implicit Non-Rigid Shape and Motion Recovery," *Proc. Int'l Conf. Computer Vision Workshop Dynamical Vision,* Oct. 2005.

[2] A. Bartoli and A. Zisserman, "Direct Estimation of Non-Rigid Registration," *Proc. British Machine Vision Conf.,* Sept. 2004.

[3] M.J. Black and A.D. Jepson, "Eigentracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," *Proc. European Conf. Computer Vision,* pp. 329-342, 1996.

[4] V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3-D Faces," *Proc. ACM SIGGRAPH,* pp. 187-194, Aug. 1999.

[5] L. Cohen and I. Cohen, "Deformable Models for 3-D Medical Images Using Finite Elements and Balloons," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 592-598, 1992.

[6] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Models," *Proc. European Conf. Computer Vision,* pp. 484-498, June 1998.

[7] H. Delingette, M. Hebert, and K. Ikeuchi, "Deformable Surfaces: A Free-Form Shape Representation," *Proc. SPIE Geometric Methods in Computer Vision,* vol. 1570, pp. 21-30, 1991.

[8] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "Dynamic Free-Form Deformations for Animation Synthesis," *IEEE Trans. Visualization and Computer Graphics,* 1997.

[9] P. Fua and Y.G. Leclerc, "Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading," *Int'l J. Computer Vision,* vol. 16, pp. 35-56, Sept. 1995.

[10] A. Kambhamettu, D. Goldgoff, D. Terzopoulos, and T.S. Huang, "Non Rigid Motion Analysis," *Handbook of Pattern Recognition and Image Processing: Computer Vision,* Academic Press, pp. 405-430, 1994.

[11] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J.Computer Vision,* vol. 1, no. 4, pp. 321-331, 1988.

[12] V. Lepetit and P. Fua, "Keypoint Recognition Using Randomized Trees," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 9, pp. 1465-1479, Sept. 2006.

[13] X. Llado, A. Del Bue, and L. Agapito, "Non-Rigid 3D Factorization for Projective Reconstruction," *Proc. British Machine Vision Conf.,* Sept. 2005.

[14] I. Matthews and S. Baker, "Active Appearance Models Revisited," *Int'l J. Computer Vision,* vol. 60, pp. 135-164, Nov. 2004.

[15] T. McInerney and D. Terzopoulos, "A Dynamic Finite Element Surface Model for Segmentation and Tracking in Multidimensional Medical Images with Application to Cardiac 4D Image Analysis," *Computerized Medical Imaging and Graphics,* vol. 19, no. 1, pp. 69-83, 1995.

[16] D. Metaxas and D. Terzopoulos, "Constrained Deformable Superquadrics and Nonrigid Motion Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 6, pp. 580-591, June 1993.

[17] L. Moccozet and N. Magnenat-Thalmann, "Dirichlet Free-Form Deformation and Their Application to Hand Simulation," *Computer Animation,* 1997.

[18] A. Pentland, "Automatic Extraction of Deformable Part Models," *Int'l J. Computer Vision,* vol. 4, no. 2, pp. 107-126, 1990.

[19] J. Pilet, V. Lepetit, and P. Fua, "Real-Time Non-Rigid Surface Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2005.

[20] R. Seidel, "A Simple and Fast Incremental Randomized Algorithm for Computing Trapezoidal Decompositions and for Triangulating Polygons," *Computational Geometry,* vol. 1, pp. 51-64, 1991.

[21] S. Roweis and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 5500, pp. 2323-2326, 2000.

[22] T.W. Sederberg and S.R. Parry, "Free-Form Deformation of Solid Geometric Models," *Proc. ACM SIGGRAPH,* vol. 20, no. 4, 1986.

[23] A. Shahrokni, T. Drummond, and P. Fua, "Fast Texture-Based Tracking and Delineation Using Texture Entropy," *Proc. Int'l Conf. Computer Vision,* Oct. 2005.

[24] H. Sidenbladh, M.J. Black, and D.J. Fleet, "Stochastic Tracking of 3D Human Figures Using 2D Image Motion," *Proc. European Conf. Computer Vision,* June 2000.

[25] J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, no. 5500, pp. 2319-2323, 2000.

[26] D. Terzopoulos and D. Metaxas, "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, pp. 703-714, 1991.

[27] L. Torresani, A. Hertzmann, and C. Bregler, "Learning Non-Rigid 3D Shape from 2D Motion," *Advances in Neural Information Processing Systems,* MIT Press, 2003.

[28] L. Torresani, D.B. Yang, E.J. Alexander, and C. Bregler, "Tracking and Modeling Non-Rigid Objects with Rank Constraints," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 493-500, 2001.

[29] J. Xiao, J.-X. Chai, and T. Kanade, "A Closed-Form Solution to Non-Rigid Shape and Motion Recovery," *Proc. European Conf. Computer Vision,* pp. 573-587, 2004.