Local Deformation Models for Monocular 3D Shape Recovery*

Mathieu Salzmann
EPFL - CVLab
1015 Lausanne, Switzerland
mathieu.salzmann@epfl.ch

Raquel Urtasun
UC Berkeley - EECS & ICSI
MIT - EECS & CSAIL
rurtasun@csail.mit.edu

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

Abstract

Without a deformation model, monocular 3D shape recovery of deformable surfaces is severly under-constrained. Even when the image information is rich enough, prior knowledge of the feasible deformations is required to overcome the ambiguities. This is further accentuated when such information is poor, which is a key issue that has not yet been addressed.

In this paper, we propose an approach to learning shape priors to solve this problem. By contrast with typical statistical learning methods that build models for specific object shapes, we learn local deformation models, and combine them to reconstruct surfaces of arbitrary global shapes. Not only does this improve the generality of our deformation models, but it also facilitates learning since the space of local deformations is much smaller than that of global ones.

While using a texture-based approach, we show that our models are effective to reconstruct from single videos poorly-textured surfaces of arbitrary shape, made of materials as different as cardboard, that deforms smoothly, and much lighter tissue paper whose deformations may be far more complex.

1. Introduction

Without a deformation model, recovering the 3D shape of a non-rigid surface from a single view is an ill-posed problem. Even given a calibrated perspective camera and a well-textured surface, the depth ambiguities cannot be resolved in individual images [15].

Standard approaches to solve this problem involve introducing either physics-based models [2, 12, 11, 14, 4] or models that can be learned from data [9, 17, 3, 10, 1, 18]. To be accurate, the former rely on knowledge of material properties, which may not be available, thus involving arbitrary parameter choices. Similarly, the latter require vast amounts of training data, which may not be available either, and produce models for specific object shapes. As a consequence, one has to learn as many deformation models as objects, even when these objects are made of the same material. Finally, most of these models are linear or quadratic and not designed to deal with complex deformations.

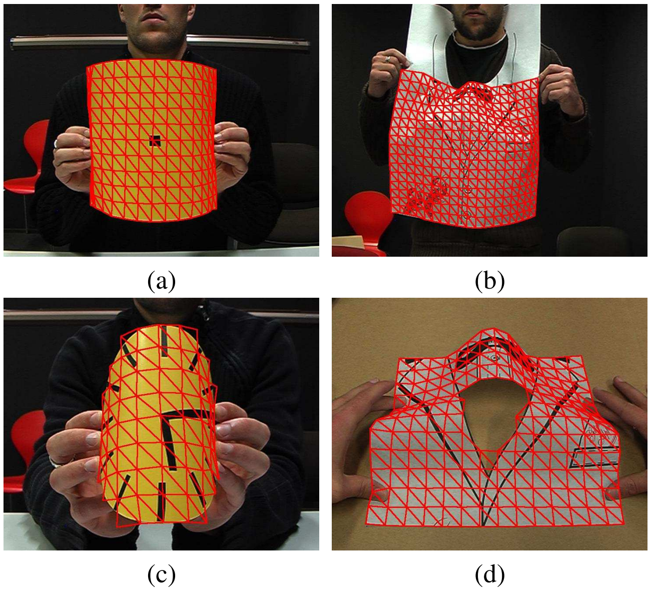


Figure 1. 3D reconstruction of poorly-textured deformable surfaces from single video sequences. (a) Cardboard with a single black square in its middle. (b) Paper napkin undergoing a much more complex deformation. (c) Even though the models have been learned by observing the deformations of rectangular sheets, they can be used to recover the shape of a circular object. (d) Similarly, they can handle one with a triangular hole in the center.

To overcome these limitations, we note that:

- locally all parts of a physically homogeneous surface obey the same deformation rules;
- the local deformations are more constrained than those of the global surface and can be learned from fewer examples.

We therefore use a non-linear statistical learning technique to represent the manifold of local surface deformations, and then combine the local models into a global one representing the particular shape of the object of interest. As shown

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

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

in Fig. 1(a,b), this approach is general and applies to materials with very different physical properties. In particular, we learned deformation models for a relatively rigid cardboard sheet that deforms smoothly and for a napkin made of much lighter tissue that undergoes more complex deformations.

Our contribution is twofold. First by using local deformation models which are more constrained than global ones, we reduce the complexity of learning and the required amount of training data. Second, because local models can be assembled into arbitrary global shapes, our deformation priors are independent of the overall shape of the object, and only one deformation model needs be learned per material. As shown in Fig. 1(c,d), having learned the model by observing a surface of a specific shape, we can use it to handle the deformations of a differently shaped surface made of the same material without any retraining.

Finally, while relying on template-matching, we deliberatly demonstrate the effectiveness of our approach on poorly-textured images that can be expected to defeat other texture-based methods. Note that our models do not depend on the specific source of image information we use. They could also improve the robustness of any shape-from-X algorithm.

2. Related Work

Monocular 3D shape recovery of deformable surfaces is known to be under-constrained, even when they are sufficiently well-textured for structure-from-motion and template-matching approaches to be effective. A priori knowledge of the possible deformations is required to solve the ambiguities inherent to the problem.

Structure from motion methods have been proposed to retrieve the 3D locations of feature points on a non-rigid surface. They typically model the deformations of a surface as linear combinations of fixed basis vectors [9], which can be learned online [17]. Since the underlying linearity assumptions limit the applicability of these methods to smooth deformations, some researchers have advocated the use of weaker and more generally applicable constraints. It has been shown that the use of lighting [19] or weak motion models [15] can resolve the ambiguities but still requires assumptions about lighting conditions or frame-to-frame motion that may not apply. Moreover, since the reconstruction relies on detected feature points, the common weakness of all these approaches is that they require the presence of texture over the whole surface.

Physically-based approaches solve this problem by introducing a global model that can infer the shape of untextured surface portions from the rest of the surface. These approaches were first introduced for 2D delineation [6] and then quickly extended to 3D reconstruction [2, 12, 11]. Due to the high dimensionality of such representations, modal analysis [14, 4] was proposed to model the deformations as linear combinations of modes. While computationally efficient, this limits the method's applicability to smoothly deforming objects, as is the case for the structure-from-motion techniques discussed above [17, 9]. In any event, even assuming some knowledge of the surface material parameters, the complexity and non-linearity of the true physics make physically-based models rough approximations of reality that are only accurate for small deformations.

Statistical learning approaches have therefore become an attractive alternative that takes advantage of observed training data. Linear approaches have been applied to faces [3, 10, 1] as well as to general non-rigid surfaces [16]. However, they impose the same restrictive smoothness constraints as before. To the best of our knowledge, non-linear techniques have not been demonstrated for 3D surface reconstruction, but have proved effective for 3D human motion tracking [18, 13]. However, for highly deformable surfaces represented by meshes with many vertices, and therefore many degrees of freedom, the number of training examples required to learn the model would quickly become intractable.

Furthermore, whether using a linear or non-linear technique, learning a global model from a database of deformed versions of a particular mesh would yield a shape prior usable only for meshes of the same topology, thereby limiting its re-usability. The non-linear approach to statistical learning we propose in this paper overcomes these weaknesses by learning local deformation models, which can be done using manageable amounts of training data. These models can then be combined into global ones for meshes of arbitrary topologies whose deformations may or may not be smooth.

3. Acquiring the Training Data

Statistical learning methods require training data that, in the case of building deformation models, can typically be hard to obtain. When dealing with large surfaces, the amount of necessary data to cover the space of possible deformations can be very large. However, since local patches have fewer degrees of freedom and can only undergo relatively small deformations, learning local deformation models becomes easier.

To collect training examples, we use a Vicon$^{\text{TM}}$ optical motion capture system. We stick 3mm wide hemispherical reflective markers on a rectangular surface and deform it arbitrarily in front of six infrared Vicon$^{\text{TM}}$ cameras that reconstruct the 3D positions of individual markers.

Since the markers are positioned to form a $P \times Q$ rectangular grid, let $\tilde{\mathbf{y}} = [x_1, y_1, z_1, ..., x_{P\times Q}, y_{P\times Q}, z_{P\times Q}]^T$ be the vector of their concatenated coordinates acquired at a specific time. Our goal being to learn a local model, as opposed to a global one, we decompose $\tilde{\mathbf{y}}$ into overlapping

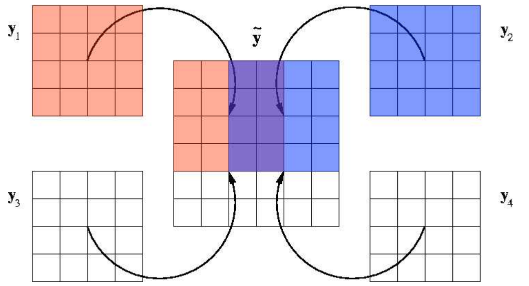


Figure 2. Decomposing the surface into patches. In this case, we cover the global surface $\tilde{\mathbf{y}}$ with four overlapping patches $\mathbf{y}_{1,..,4}$.

$p \times q$ rectangular patches centered on individual grid vertices, as shown in Fig. 2.

We collect these patches from individual temporal frames in several motion sequences, subtract their mean, and symmetrize them with respect to their x-, y- and z-axes to obtain additional examples. This results in a large set of N_a $p \times q$ patches $\mathbf{y}_{i\ ,\ i=1,..,N_a}$. Since the sequences are acquired at 30 Hz and might comprise similar deformations, we retain only a subset $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$ of $N < N_a$ patches that are different enough from each other based on a vertex-to-vertex distance criterion.

In particular, as shown in Fig. 1, we demonstrate the generality of our approach considering two different materials: Relatively rigid cardboard and more flexible tissue paper. For the cardboard, we placed the reflective markers on a 9×9 grid, and for the napkin, in a 9×7 one. This difference in resolution was only introduced to facilitate the motion capture and has no bearing on the rest of the approach. In both cases, the markers were placed 2cm apart in both directions. Out of 10 motion sequences for each material, we set one aside for validation purposes and used the other 9 for learning. In each frame, we selected five 5×5 patches for the cardboard and six for the napkin, and pruned the resulting set such that the minimum distance between corresponding vertices in separate patches was greater than 0.7cm for the cardboard and 1cm for the napkin. This produced 2032 patches for the cardboard and 2881 for the napkin. The larger number of the latter reflects the greater flexibility of the tissue paper.

4. Local Surface Model

In the previous section, we explained how we gathered data as patches of a surface. We will now show how to learn local deformation models from such data. Recall that the sample patches are in the form of $p \times q$ arrays of 3D vertices. Since $p = q = 5$ in our experiments, our task becomes learning a deformation model in a $D = p \times q \times 3 = 75$ dimensional space.

In theory, any technique that provides a probability density function over such a space is suitable. However, the number of training examples required to fully cover the space of possible deformations grows exponentially with the dimensionality and quickly becomes too high to model the density in shape space. We handle the curse of dimensionality using the Gaussian Process Latent Variable Model (GPLVM) [7, 8] whose probability density function is conditioned on a space of reduced dimensionality.

The GPLVM relates a high-dimensional data set, $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$, where $\mathbf{y}_i \in \Re^D$, and a low dimensional latent space, $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T$, where $\mathbf{x}_i \in \Re^d$, using a Gaussian process mapping from $\mathbf{X}$ to $\mathbf{Y}$. The likelihood $p(\mathbf{Y} \,|\, \mathbf{X}, \Theta)$ of the data given the latent positions is

$$\frac{1}{\sqrt{(2\pi)^{ND}|\mathbf{K}|^D}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\right)\right), \tag{1}$$

where the elements of the kernel matrix $\mathbf{K}$ are defined by the covariance function, $k(\mathbf{x}, \mathbf{x}')$, such that $(\mathbf{K})_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, with kernel hyper-parameters Θ. Here, we use a kernel that is the sum of an RBF, a bias or constant term, and a noise term. Learning the GPLVM [7] involves maximizing the posterior $p(\mathbf{X}, \Theta | \mathbf{Y}) \propto p(\mathbf{Y} \,|\, \mathbf{X}, \Theta)\, p(\mathbf{X})\, p(\Theta)$ with respect to $\mathbf{X}$ and Θ, where $p(\Theta)$ is a simple prior over the hyper-parameters of the kernel covariance function, and $p(\mathbf{X})$ encourages the latent positions to be close to the origin.

While effective at learning complex manifolds, the GPLVM suffers from the fact that its computional cost grows as $\mathcal{O}(N^3)$. Sophisticated sparsification techniques have recently been proposed [8] and have proven more accurate than simply using a subset of the data. By introducing a set of m inducing variables $\mathbf{X}_\mathbf{u}$ and assuming independence of the training and testing outputs given the inducing variables, the computational complexity can be reduced to $\mathcal{O}(Nm^2)$.

Learning the sparse GPLVM is done by maximizing with respect to $\mathbf{X}$, $\mathbf{X}_\mathbf{u}$ and Θ the posterior

$$\begin{aligned} p(\mathbf{Y} \,|\, \mathbf{X}, \mathbf{X}_\mathbf{u}, \Theta) &= \\ &\mathcal{N}\left(\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{X}_\mathbf{u}, diag[\mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{Q}_{\mathbf{f},\mathbf{f}}] + \beta^{-1}\mathbf{I}\right) , \end{aligned} \tag{2}$$

where $diag[\mathbf{B}]$ is a diagonal matrix whose elements match the diagonal of $\mathbf{B}$, and $\mathbf{Q}_{\mathbf{f},\mathbf{f}} = \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}$. $\mathbf{K}_{\mathbf{u},\mathbf{u}}$ is the kernel matrix for the elements of $\mathbf{X}_\mathbf{u}$, $\mathbf{K}_{\mathbf{f},\mathbf{u}}$ denotes the covariance between $\mathbf{X}$ and $\mathbf{X}_\mathbf{u}$, and β is the inverse noise variance.

Given a new test point $\mathbf{y}'$, inference in the sparse GPLVM is done by maximizing $p(\mathbf{y}', \mathbf{x}' | \mathbf{X}_\mathbf{u}, \mathbf{Y}, \Theta)$, with respect to $\mathbf{x}'$, the latent coordinates of the test point, or equivalently by minimizing its negative log likelihood given, up to an additive constant, as

$$\mathcal{L}(\mathbf{x}', \mathbf{y}') = \frac{\|\mathbf{y}' - \mu(\mathbf{x}')\|^2}{2\sigma^2(\mathbf{x}')} + \frac{D}{2}\ln\sigma^2(\mathbf{x}') + \frac{1}{2}\|\mathbf{x}'\|^2 , \tag{3}$$

with the mean and variance given by

$$\mu(\mathbf{x}') = \mathbf{Y}^T\mathbf{K}_{\mathbf{f},\mathbf{u}}^T\mathbf{A}^{-1}\mathbf{k}_\mathbf{u} , \tag{4}$$

$$\sigma^2(\mathbf{x}') = k(\mathbf{x}', \mathbf{x}') - \mathbf{k}_\mathbf{u}^T(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} - \beta^{-1}\mathbf{A}^{-1})\mathbf{k}_\mathbf{u} , \tag{5}$$

where $\mathbf{A} = \beta^{-1}\mathbf{K}_{\mathbf{u},\mathbf{u}} + \mathbf{K}_{\mathbf{u},\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{u}}$, and $\mathbf{k}_{\mathbf{u}}$ is the vector with elements $k(\mathbf{x}', \mathbf{x}_j)$ for latent positions $\mathbf{x}_j \in \mathbf{X}_{\mathbf{u}}$.

5. From Local to Global

To model the global behavior of a surface we combine local models using a Product of Experts (PoE) paradigm. We first argue that this gives a valid representation for a surface of infinite length. We then show how the basic scheme can be modified to account for boundaries.

5.1. PoE for Deformable Surfaces

Products of Experts (PoE) [5] provide a good solution to representing high-dimensional data subject to low-dimensional constraints by combining probabilistic models. Each constraint is treated by an individual expert, which gives a high probability to the examples that satisfy it. The probability of examples statisfying some constraints but violating others will naturally be decreased by the experts associated with the violated ones.

In the general case, training a PoE is difficult because one has to identify the experts that simultaneously maximize the probabilities of training examples and assign low probabilities to unobserved regions of the data space. However, in the case of homogeneous surfaces, this task becomes much easier; The PoE does not have to identify the different experts since all local patches obey the same deformation rules. As a consequence, one can simply train a single local deformation model corresponding to one expert and, for inference, replicate it to cover the entire surface as shown in Fig. 2. This simply assumes that maximizing the likelihood of a global shape is achieved through maximizing the likelihoods of all the patches. Note that the choice of the patch size influences both the local and global representations. Smaller sizes result in models that are more constrained since less deformations are possible, but impose a higher number of experts to cover the global surface.

More formally, let $\tilde{\mathbf{y}}$ be the vector of all 3D vertex coordinates, $\mathbf{y}' = \left[\mathbf{y}_1'^T, ..., \mathbf{y}_S'^T\right]^T$ the 3D coordinates of the S overlapping patches associated with the experts, where $\mathbf{y}_i'$ is a subset of $\tilde{\mathbf{y}}$, and $\mathbf{x}' = \left[\mathbf{x}_1'^T, ..., \mathbf{x}_S'^T\right]^T$ the experts' latent coordinates. The conditional probability of the global surface can be written as

$$p(\tilde{\mathbf{y}}|\mathbf{x}', \mathcal{M}) = \frac{\prod_i p_i(\mathbf{y}_i'|\mathbf{x}_i', \mathcal{M})}{\int \prod_i p_i(\mathbf{y}_i'|\mathbf{x}_i', \mathcal{M}) d\tilde{\mathbf{y}}} \ , \tag{6}$$

where $\mathcal{M}$ is the local GPLVM described in Section 4.

Assuming that the denominator of Eq. 6 is constant, we can define a prior over the deformation of the whole surface according to all the experts as

$$\mathcal{L}_{poe} = \sum_{i=1}^{S} \mathcal{L}(\mathbf{x}_i', \mathbf{y}_i') \ , \tag{7}$$

where $\mathcal{L}$ is defined in Eq. 3. We use overlapping experts to enforce smooth transitions between neighboring experts. However this does *not* impose global surface smoothness, since the local models may allow for sharp folds.

5.2. Surface Boundary Effects

As shown in Fig. 2 boundary vertices influence fewer experts than interior ones. As a consequence, their position has only little effect on the likelihood of Eq. 7, resulting in vertices that can move freely.

To avoid this undesirable effect, we re-weight the terms of Eq. 7 such that the influence of each vertex is inversely proportional to the number of patches it belongs to. The negative log likelihood $\mathcal{L}_{global}$ of the global surface is then

$$\sum_{i=1}^{S} \left(\frac{\sum_{j=1}^{p \times q} \frac{1}{V(i,j)} (\mathbf{W}\mathbf{y}_{i,j}' - \mu^j(\mathbf{x}_i'))^2}{2\sigma^2(\mathbf{x}_i')} + \frac{D}{2} \ln \sigma^2(\mathbf{x}_i') + \frac{1}{2} \|\mathbf{x}_i'\|^2 \right) \ , \tag{8}$$

where $\mathbf{y}_{i,j}'$ is the j^{th} vertex of patch i and $\mu^j(\mathbf{x}_i')$ its corresponding mean prediction. $V(i,j)$ is the number of patches for a vertex, which depends on the index in the global representation of the j^{th} vertex of patch i.

Furthermore, we also introduced a 3×3 diagonal matrix $\mathbf{W}$ in Eq. 8 that defines the global scales along the x-, y- and z-axes, and accounts for the difference in scale between the training and testing surfaces. In practice, we allow for at most 10% scaling.

6. Monocular Reconstruction

We formulate our reconstruction algorithm as an optimization problem with respect to a state vector ϕ. At each time t, the state is defined as $\phi_t = \left[\mathbf{y}_t^T, \mathbf{x}_t^T\right]^T$, where $\tilde{\mathbf{y}}_t$ is the vector of the 3D coordinates of the global surface, and $\mathbf{x}_t = \left[\mathbf{x}_{1,t}^T, ..., \mathbf{x}_{S,t}^T\right]^T$ denotes the latent variables for the local models. Note that this formulation guarantees surface continuity, since the patches share a common vector of vertex coordinates. Given a new image $\mathbf{I}_t$ and a local deformation model $\mathcal{M}$, we seek to recover the MAP estimate ϕ_t. We therefore approximate the posterior

$$p(\phi_t|\mathbf{I}_t, \mathcal{M}) \propto p(\mathbf{I}_t|\phi_t) p(\phi_t|\mathcal{M}) \ , \text{ where} \tag{9}$$

$$p(\phi_t|\mathcal{M}) \approx p(\mathbf{y}_t|\mathbf{x}_t, \mathcal{M}) p(\mathbf{x}_t) \ , \tag{10}$$

with $p(\mathbf{I}_t|\phi_t)$ the image likelihood, and $-\ln p(\mathbf{x}_t) = \sum_{i=1}^{S} ||\mathbf{x}_{i,t}||^2$.

6.1. Image Likelihood

To estimate the image likelihood, we rely on texture and edge information. The latter constrains the boundary vertices which are not as well-constrained by texture as the interior ones. Assuming that both sources of information are independent given the state, we can write

$$p(\mathbf{I}_t|\phi_t) = p(\mathbf{T}_t|\phi_t) p(\mathbf{E}_t|\phi_t) \ . \tag{11}$$

To take advantage of the whole texture, we use template matching. Assuming we have a reference image $\mathbf{I}_{\text{ref}}$ in which we know the shape of the surface $\mathbf{y}_{\text{ref}}$, each facet of the surface mesh is treated as a separate template that we match in image $\mathbf{I}_t$. The negative log likelihood of such observations is given by

$$-\ln\, p(\mathbf{T}_t|\phi_t) = \frac{1}{\sigma_T^2}\sum_{i=1}^{N_f}\gamma(\Psi(P(\mathbf{y}_{\text{ref}},j,\mathbf{I}_{\text{ref}}),\phi_t),P(\mathbf{y}_t,j,\mathbf{I}_t)) \; , \tag{12}$$

where N_f is the number of facets, γ denotes the normalized cross-correlation function, $P(\mathbf{y},j,\mathbf{I})$ is the projection of facet j of surface $\mathbf{y}$ in image $\mathbf{I}$, and $\Psi(.,\phi)$ denotes the function that warps an image to another using parameters ϕ. σ_T is a constant set to the variance of the expected texture error. In practice, the results are relatively insensitive to its exact value.

To constrain the boundary of the surface, we sample the border of the mesh and look in the direction of its normal for an edge point detected by Canny's algorithm that matches the projection of the sample. We allow for multiple hypotheses and retain all the matches within a distance r from the current reprojection, which decreases from 8 to 2 pixels as the optimization proceeds. The negative log likelihood of the edge observations is then

$$-\ln\, p(\mathbf{E}_t|\phi_t) = \frac{1}{\sigma_E^2}\left(\frac{1}{r^2}\sum_{i=1}^{N_e}\sum_{j=1}^{N_h(i)}\|\mathbf{u}_{i,j}-\mathbf{e}_i(\phi_t)\|^2\right) \; , \tag{13}$$

where N_e is the number of sampled boundary points, $\mathbf{e}_i$ denotes the boundary point projected in the image, $N_h(i)$ is the number of edge hypotheses for point i, and $\mathbf{u}_{i,j}$ is the corresponding image measurement. As for texture, σ_E is a constant corresponding to the variance of the expected error, and whose precise value has only little influence on the results.

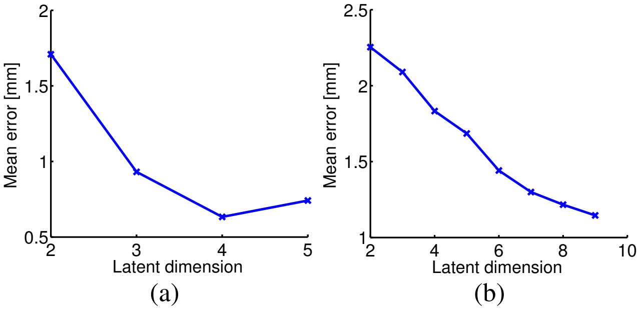
6.2. Optimization

Reconstruction is performed by minimizing the negative log of the approximate posterior of Eq. 9, which we write, up to an additive constant, as

$$\mathcal{L}_{global}(\phi_t) - \ln\, p(\mathbf{T}_t|\phi_t) - \ln\, p(\mathbf{E}_t|\phi_t) \; . \tag{14}$$

In practice, we assume that the projection matrix is known and remains constant throughout the sequence. This entails no loss of generality since the vertices are free to move rigidly. Furthermore, the reference image and shape may be those of the first image of the sequence, or of any model image.

When considering large surfaces, the number of degrees of freedom of our optimization problem quickly becomes large, since it includes the 3D positions of the vertices. To improve convergence, we introduce a coarse-to-fine approach to optimization. In the first step we only consider every other row and every other line of the grid representing the local patches. Therefore, we end up with patches of 3×3 vertices separated by 4cm instead of 5×5 vertices separated by 2cm. While not changing the number of local models that we use, this drastically reduces the number of vertices to optimize. Furthermore, this only changes the resolution of the patches, but not their size. Therefore we can still use the same local deformation models to represent the shape of the patches.



(a) (b)

Figure 3. Validating the deformation models. We compute the mean of the average vertex-to-vertex distances between test data and the model predictions, and plot it versus the latent dimension. (a) In the cardboard case, with 100 inducing variables, latent dimension 4 performs best, whereas dimension 5 overfits the training data. (b) In the case of the napkin, 200 inducing variables were sufficient for dimensions 2 to 7, but 400 were required for dimensions 8 and 9. The improvement between dimensions 7 and 8 is so small that we chose to use 7 to limit the computational burden.

In the first frame, we start from the reference shape and initialize the latent positions of the local models such that their mean predictions best correspond to the different patches of the reference shape. This is done by optimizing the negative log likelihood of Eq. 3. Then, at every following frame, we initialize the state with the MAP estimate of the previous time, obtain a reconstruction with a low resolution mesh, and use it to initialize the fine mesh that is then optimized as well.

7. Experimental Results

In this section, we first validate the local models we learned for cardboard and tissue paper, and then use both synthetic and real data to demonstrate that they sufficiently constrain the reconstruction to achieve accurate results, even when the lack of texture on the surfaces makes it difficult for texture-based approaches. The corresponding videos are submitted as supplemental material. We encourage the reader to watch them to check that the recovered 3D shapes match our perception of the deformations.

7.1. Local Models Validation

We used the technique of Section 4, to learn models for the two datasets discussed in Section 3 for latent dimensions ranging from 2 to 9. We then picked the dimensionality that best fitted our validation set.

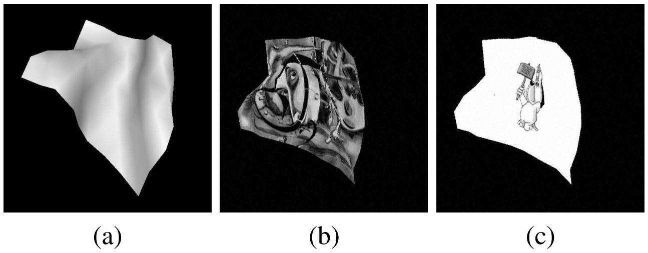


Figure 4. Synthetic images generated from optical motion capture data (a) Shaded view of a surface. (b,c) Images synthesized by texture-mapping using either a rich texture or a much more uniform one.

More specifically, for each patch $\mathbf{y}'_i$ extracted from the validation sequence, we infered the corresponding latent variable $\mathbf{x}'_i$ by minimizing the negative log likelihood of Eq. 3, and computed the mean vertex-to-vertex distance between $\mathbf{y}'_i$ and the model prediction $\mu(\mathbf{x}'_i)$. In Fig. 3, we depicts the mean of these distances as a function of the latent dimension. For the cardboard, the models were all trained using 100 inducing variables. $d = 4$ yields the smallest average distance. Larger values of d overfit to the training data and yield worse results for the validation set. For the napkin, that has more samples and a greater variety of observed shapes, we had to use 200 inducing variables for $2 \leq d \leq 7$ and 400 for $8 \leq d \leq 9$ to make the training process converge. In this case the higher values of d yield slightly better results. However, since using 400 inducing variables instead of 200 carries a severe computational penalty, in our experiments we use $d = 7$, which we will show to be sufficient for our purposes.

In any event, using a larger latent dimension for tissue paper than for cardboard tallies with our intuition that the manifold of potential deformations of the former is larger than that of the latter.

7.2. Synthetic Data

We measured the accuracy of our method, and compared it on synthetically generated images against regularizing either via deformation modes [9] or using a standard quadratic term [6]. Modal analysis was performed by computing the covariance matrix of our optical motion capture data, and modeling the surface as a linear combination of its eigenvectors, whose weights became the unknown of our optimization problem. Regularization was achieved by introducing a term that penalizes the modes weights with their corresponding inverse squared eigenvalues. Using a subset of the napkin validation data, such as the surface in Fig. 4(a), we formed a sequence of deforming meshes, textured them and projected them with known perspective camera to obtain noise-free images. We then added i.i.d. noise in the range $[-10, 10]$ to the image intensities. We reconstructed surfaces from a well-textured sequence and from a more uniform one. As can be seen in Fig. 5, where we plot reconstruction errors, our method gives substantially better results than both other approaches.

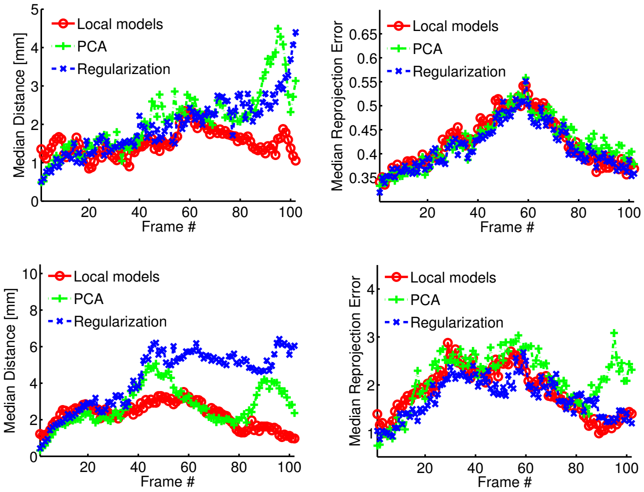


Figure 5. Comparison of our approach (red circles) against principal component analysis (green pluses) and standard quadratic regularization (blue crosses) using sequences of images such as those of Fig. 4. **Top Row** For each of the well-textured images, we plot, on the left, the median 3D vertex-to-ground-truth-surface distance, and, on the right, the median reprojection error of randomly sampled surface points. **Bottom Row** Same plots for the much less textured images. While the reprojection errors are similar for all approaches, the surfaces we recover correspond more closely to the true ones, especially when there is little texture. This confirms that our deformation model better approximates the true object behavior.

7.3. Real Sequences

We first applied our approach to the sheet of carboard and the paper napkin of Figs. 6 and 7. We needed to combine local deformation models to represent their shape eventhough they are rectangular, because their size is different from the one of the training data.

The top row of Fig. 6 shows the behavior of our technique when there is absolutely no texture to anchor the surface. The recovered surface belongs to a family of equally-likely shapes whose vertices can slide across the surface, while their boudaries reproject correctly. Nothing in the image likelihood prevents this, since all facets look similar. Note that, without using of shading clues, even a human eye could hardly differentiate between two such shapes. However, as shown in the second example of the figure, adding only very little texture disambiguates the reconstruction. Finally, when incrasing only slighly the amount of texture, even more complex deformations can be recovered accurately, as shown in the third example.

Our technique can be applied to very different shapes and topologies, e.g. a circular shape and a surface with a hole, as shown in Fig. 8. Our models being made of rectangular patches, the meshes we use only roughly approximate

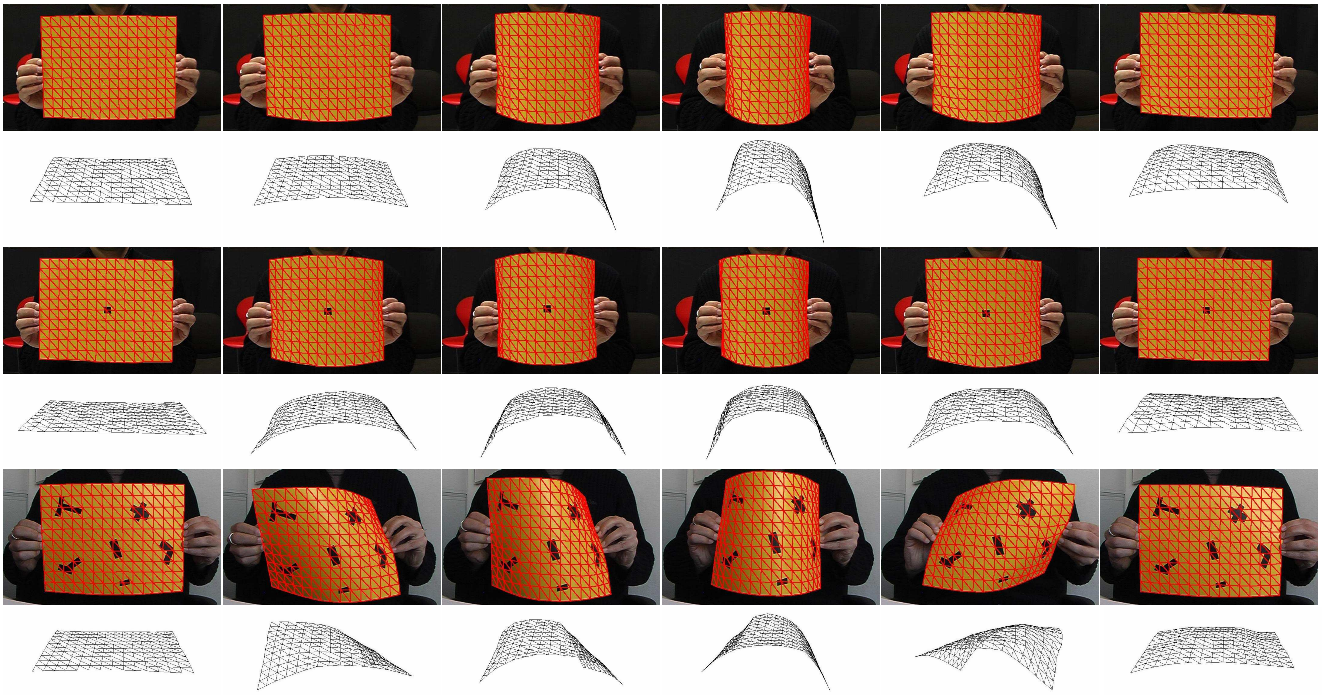


Figure 6. Reconstructing a rectangular piece of cardboard from a single video. In each of the three examples, we show the recovered surface overlaid in red on the original images, and the surface seen from a different viewpoint. As shown in the top rows, a complete absence of texture leads us to retrieve a surface that is plausible, but not necessary accurate. It is only one of a whole family of equally likely solutions. However, this problem is fixed by adding very little image information, as shown in the other two examples. We then recover deformations that match the real ones.

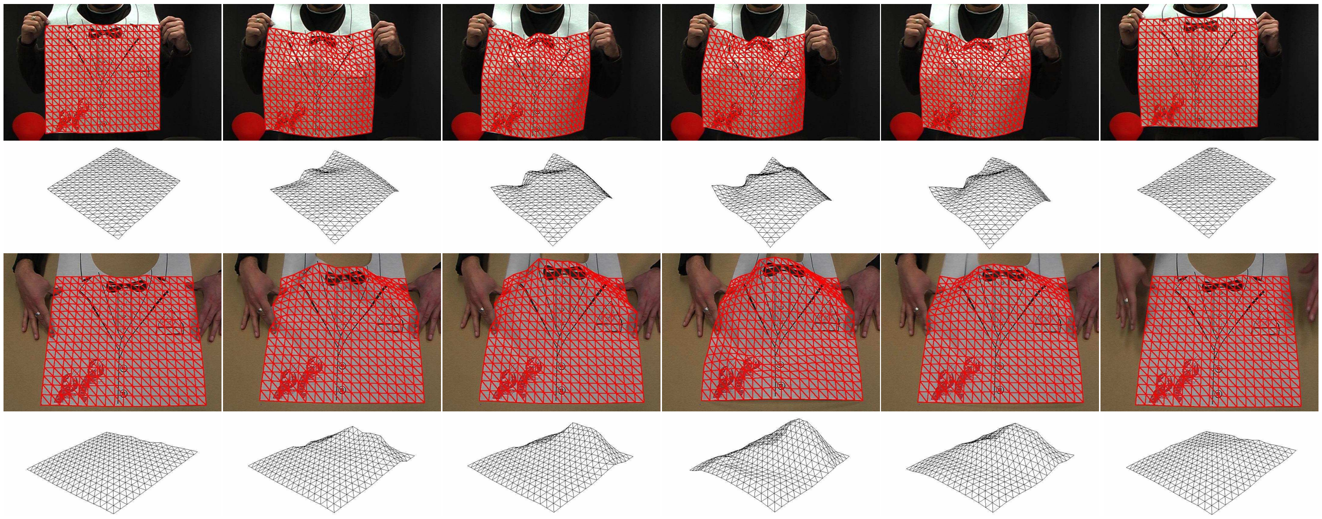


Figure 7. Reconstructing a much more flexible paper napkin. Even though there is little texture, the 3D of the surface is correctly recovered, as shown in the bottom row where the surface is seen from a different perspective.

the surface boundaries, which prevents us from using edge information. We nevertheless recover the 3D deformations in both cases. Finally, in Fig. 9, we show that our models make our approach robust to occlusions.

8. Conclusion

In this paper, we have presented an approach to recovering the 3D shape of a poorly-textured surface from a single camera. We have introduced local deformation models that can be learned from a relatively small amount of training data, and have shown that they can be combined to model arbitrary global shapes. Furthermore, in the limit of the size of the local patches, our method can be interpreted as either a local smoothness or a global shape prior, and therefore subsumes these two earlier approaches.

In future work, we plan to capture and learn models from

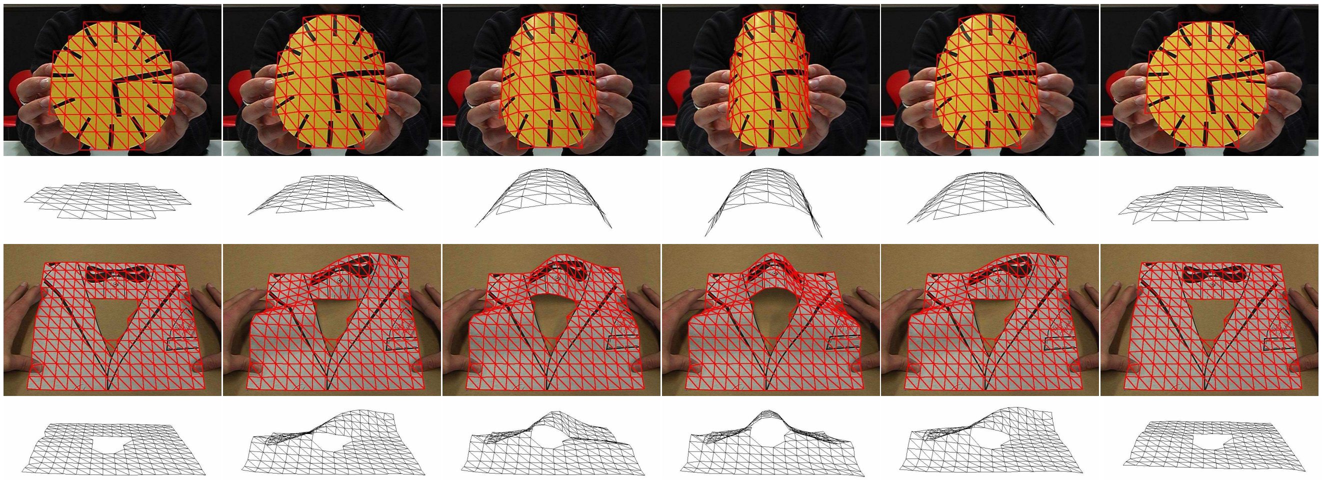


Figure 8. Reconstructing objects of different shape and topology. Note that assembling square patches only allows us to approximate object outline. This prevents us from using some image edges, but does not stop us from successfully recovering the deformations of a circular shape and of one with a hole.

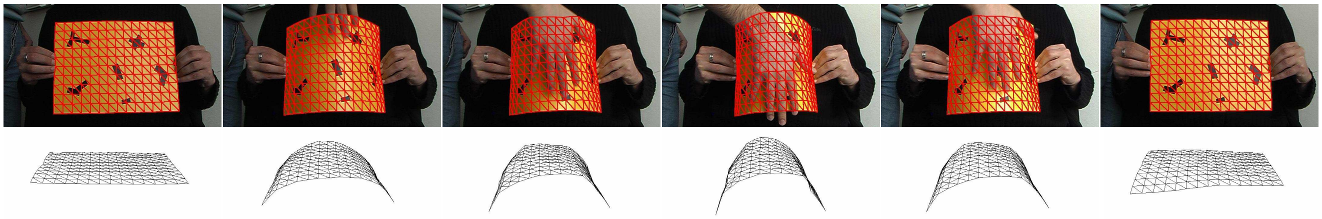


Figure 9. Despite a very large occlusion, we manage to reconstruct a deforming piece of cardboard in each frame of a sequence. Note that even if some small reconstrution errors occur, the global shape nevertheless matches the true one.

additional materials. This will allow us to study the problem of material recognition from video sequences by choosing the model that yields the best reconstruction. Finally, we plan to study the influence of dynamics by replacing our current patch shape models with models trained on short motion sequences, which should further improve the stability of the reconstruction.

References

[1] V. Blanz and T. Vetter. A Morphable Model for The Synthesis of 3–D Faces. In *ACM SIGGRAPH*, pages 187–194, 1999.

[2] L. Cohen and I. Cohen. Deformable models for 3-d medical images using finite elements and balloons. In *CVPR*, pages 592–598, 1992.

[3] T. Cootes, G. Edwards, and C. Taylor. Active Appearance Models. In *ECCV*, pages 484–498, 1998.

[4] H. Delingette, M. Hebert, and K. Ikeuchi. Deformable surfaces: A free-form shape representation. In *Geometric Methods in Computer Vision*, volume 1570, pages 21–30, 1991.

[5] G. Hinton. Products of experts. In *ICANN*, pages 1– 6, 1999.

[6] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *IJCV*, 1(4):321–331, 1988.

[7] N. D. Lawrence. Gaussian Process Models for Visualisation of High Dimensional Data. In *NIPS*, 2004.

[8] N. D. Lawrence. Learning for Larger Datasets with the Gaussian Process Latent Variable Model. In *AISTATS*, 2007.

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

[10] I. Matthews and S. Baker. Active Appearance Models Revisited. *IJCV*, 60:135–164, 2004.

[11] T. McInerney and D. Terzopoulos. A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking. In *ICCV*, pages 518–523, 1993.

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

[13] R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The Joint Manifold Model for Semi-supervised Multi-valued Regression. In *ICCV*, 2007.

[14] A. Pentland. Automatic extraction of deformable part models. *IJCV*, 4(2):107–126, 1990.

[15] M. Salzmann, V. Lepetit, and P. Fua. Deformable Surface Tracking Ambiguities. In *CVPR*, 2007.

[16] M. Salzmann, J. Pilet, S. Ilić, and P. Fua. Surface Deformation Models for NonRigid 3D Shape Recovery. *PAMI*, 29(8):1481–1487, 2007.

[17] L. Torresani, A. Hertzmann, and C. Bregler. Learning non-rigid 3d shape from 2d motion. In *NIPS*, 2003.

[18] R. Urtasun, D. Fleet, A. Hertzman, and P. Fua. Priors for people tracking from small training sets. In *ICCV*, 2005.

[19] R. White and D. Forsyth. Combining cues: Shape from shading and texture. In *CVPR*, 2006.