Deformable Surface Tracking Ambiguities

Mathieu Salzmann, Vincent Lepetit and Pascal Fua
Computer Vision Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland
{mathieu.salzmann,vincent.lepetit,pascal.fua}@epfl.ch

Abstract

We study from a theoretical standpoint the ambiguities that occur when tracking a generic deformable surface under monocular perspective projection given 3–D to 2–D correspondences. We show that, additionally to the known scale ambiguity, a set of potential ambiguities can be clearly identified.

From this, we deduce a minimal set of constraints required to disambiguate the problem and incorporate them into a working algorithm that runs on real noisy data.

1. Introduction

Without a strong model, 3–D shape recovery of non-rigid surfaces from monocular video sequences is a severely under-constrained problem. Prior models are required to resolve the inherent ambiguities.

Many approaches to creating such models have been proposed, such as physics-based models [10, 11, 3, 5, 9], feature point-based structure from motion algorithms [13, 7, 16] and machine learning techniques [2, 12, 8]. However, as will be discussed in Section 2, these methods typically make restrictive assumptions that prevent them from being completely general.

Furthermore, we are not aware of any formal study of the ambiguities when explicitly reconstructing deformable surfaces in the total absence of prior knowledge, or of the number of constraints that must be supplied to resolve them. In this paper, we address this issue from a theoretical standpoint and show how such a theoretical understanding can be translated into working algorithms that make minimal assumptions on the range of possible surface deformations.

As shown in Fig. 1, we focus here on surfaces that are textured enough to let us establish 3–D to 2–D correspondences between interest points on the surface and their image locations but whose physical properties may be very different. Requiring texture is a limiting assumption but our approach nevertheless represents a key step towards designing video-based tracking algorithms able to reconstruct the deformations of classes of deformable surfaces whose behavior is not known *a priori*: Given a robust algorithm able to recover the deformations of such a surface when it is sufficiently textured, it will become feasible to construct large training sets of such deformations; to use them to learn low-dimensional deformation models; and finally to use these models to recover the shape of surfaces of the same class even though they may be less textured.

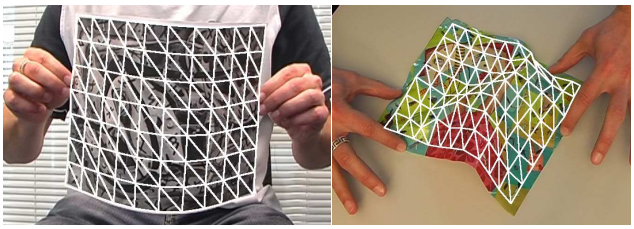


Figure 1. Reconstruction of deformable surfaces from video sequences with minimal *a priori* knowledge. We constrain the reconstruction of the deforming sheet of paper and of the much flexible plastic sheet in the same manner, even though they have very different physical properties.

More specifically, we model our surfaces as triangulated meshes seen under perspective projection. Computing the 3–D coordinates of its vertices can be achieved by solving a large linear system, whose rank and singular values we can easily compute. This will allow us to show that:

- Given sufficiently many noise-free correspondences, the coordinates can be retrieved up to a single scale ambiguity.
- In practice, the image locations of the correspondences are never perfect and the resulting ambiguities can be attributed to the presence of very small singular values in the linear system. These ambiguities actually correspond to those of a piecewise affine model, which introduces an extra depth ambiguity for each vertex.
- The ambiguities can be resolved by considering a sequence of images instead of a single one and imposing a very simple dynamics model that links the reconstructions in consecutive images. This results in a much larger linear system but of full rank thanks to the additional motion constraints.

1-4244-1180-7/07/$25.00 ©2007 IEEE

We will show that for surfaces with physical properties as different as those of the sheet of paper and the piece of plastic of Fig. 1, the same set of generic constraints allows us to resolve the ambiguities. As a result, we can retrieve their overall shape as they deform, even though the correspondences we use are automatically established and therefore contain many errors.

2. Related Work

Recovering the shape of a deforming surface in a monocular sequence requires prior knowledge to make the problem tractable. Many different approaches have been studied over the years, most of which make very strong and restrictive assumptions about the object of interest.

Physics-based deformation models have been used extensively to add a qualitative knowledge about the object's behaviour. The original 2–D models were first applied to shape recovery [6], but have also been used for 2–D surface registration [1]. They have rapidly been adapted to 3–D under the form of superquadrics [10], triangulated surfaces [3], or thin-plate splines [9]. To reduce the dimensionality of the problem, linearity assumptions have also been made on those models through modal analysis [11, 3, 5]. Even though these models have been extremely successful, the imply some knowledge of the pseudo-physical properties of the surface, which may not be available. Furthermore, the complexity of modeling a true nonlinear behavior tends to restrict them to cases where nonlinearities are small.

Structure from motion methods have also been shown to be effective. They rely on feature points tracked through a sequence to retrieve the deformed shape of a surface. A common assumption in such methods is to consider the deformations as being a linear combination of bases vectors [7], which can be learned during the process [13]. This of course does not correspond to the true behaviour of a surface which, by nature, deforms nonlinearly. A slightly different approach is to consider piecewise rigid deformations [16]. In this case, rigid objects are moving independently, and the motion of the whole scene is considered as a deformation. This, again, introduces a strong prior, which in general is not valid for a deformable surface.

Machine learning techniques have known an increasing popularity in the past few years. They make use of training data to build a model that can then be applied to track objects from monocular images. Even though nonlinear dimensionality reduction methods have proved successful in the case of human motion [14], most applications to deformable surfaces have been linear. Active appearance models [4] pioneered this approach in the 2–D case and have since been extended to 3–D [8]. Morphable models [2] rely on the same philosophy to build 3–D deformable face models. Recently [12] applied this idea to deformable surface tracking, and created a training set of deformed shapes by varying angles between the facets of a mesh. However, their training data are far from corresponding to reality. Machine learning methods have proved efficient, but suffer from the need of good training sets which might be hard to obtain, especially in the case of deformable surfaces.

Finally, it was recently shown that texture and shading information could be combined to retrieve the shape of a deformable surface [15]. However, very strong assumptions on the lighting environment must be made, and therefore the method lacks generality.

The ultimate goal of our research is to solve the problem of building training sets of deformable surfaces from textured objects and with minimal prior knowledge of the feasible deformations. This would constitute a good starting point to learn accurate deformation models that could then be applied to less textured surfaces of the same kind. We therefore see this theoretical study as a necessary step towards fully understanding the problem we are facing and showing that a few reasonable assumptions can make it tractable.

3. Single-Image Ambiguities

We represent surfaces as 3–D triangulated meshes and assume that we are given a set of 3–D to 2–D correspondences between surface points and image locations. In this section, we show that recovering the shape amounts to solving an ill-conditioned linear system. We then show that the degeneracies, or near-degeneracies, of this system correspond to depth ambiguities that can be explained in terms of a piecewise affine projection model. Since we use a single camera and assume its internal parameters to be known, we express all world coordinates in the camera referential for simplicity and without loss of generality.

3.1. Ambiguities under Perspective Projection

In this section, we formulate the computation of the 3–D mesh vertex coordinates given the correspondences in terms of solving a linear system and discuss its degeneracies. We start with a mesh containing a single triangle and extend our result to a complete one.

Projection of a 3–D Surface Point Let $\mathbf{x_i}$ be a 3–D point whose coordinates are expressed in the camera referential. We write its perspective projection as

$$\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \frac{1}{k_i} \mathbf{A} \left[\mathbf{I}_{3\times 3} \,\middle|\, \mathbf{0} \right] \begin{pmatrix} \mathbf{x_i} \\ 1 \end{pmatrix} \quad , \tag{1}$$

where $\mathbf{A}$ is the internal parameters matrix, and k_i a scalar.

If $\mathbf{x_i}$ lies on the facet of a triangulated mesh, it can be expressed as a weighted sum of the facet vertices. Eq. 1

becomes

$$\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \frac{1}{k_i}\mathbf{A}(a_i\mathbf{v_1} + b_i\mathbf{v_2} + c_i\mathbf{v_3}) \ , \qquad (2)$$

where $\mathbf{v_i}, \mathbf{1 \leq i \leq 3}$ are the 3-D coordinate vectors of the vertices and (a_i, b_i, c_i) the barycentric coordinates of $\mathbf{x_i}$.

Reconstructing a Single Facet Let us assume that we are given a list of n such 3–D to 2–D correspondences for points lying inside one single facet. The $\mathbf{v_i}, \mathbf{1 \leq i \leq 3}$ coordinates of its vertices can be computed by solving

$$\mathbf{M_f}\left(\mathbf{v_1}^{\mathrm{T}} \ \mathbf{v_2}^{\mathrm{T}} \ \mathbf{v_3}^{\mathrm{T}} \ k_1 \ ... \ k_i \ ... \ k_n\right)^{\mathrm{T}} = \mathbf{0} \text{ with} \qquad (3)$$

$$\mathbf{M_f} = \left(\begin{array}{ccc|cccccc} a_1\mathbf{A} & b_1\mathbf{A} & c_1\mathbf{A} & -\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} & 0 & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... \\ a_i\mathbf{A} & b_i\mathbf{A} & c_i\mathbf{A} & 0 & ... & -\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} & 0 & ... \\ ... & ... & ... & ... & ... & ... & ... & ... \\ a_n\mathbf{A} & b_n\mathbf{A} & c_n\mathbf{A} & 0 & ... & ... & ... & -\begin{pmatrix} u_n \\ v_n \\ 1 \end{pmatrix} \end{array}\right)$$

where the k_i are treated as auxiliary variables to be recovered as well.

We could have hoped that, for $n > 4$, the columns of $\mathbf{M_f}$ would become linearly independent and that the system would then have a unique solution. However, this is not what happens.

To prove that $\mathbf{M_f}$ is rank-deficient, we show that its last column can always be written as a linear combination of the others as follows. From Eq. 2 we can write

$$\begin{pmatrix} u_n \\ v_n \\ 1 \end{pmatrix} = a_n\mathbf{A}\lambda_\mathbf{1} + b_n\mathbf{A}\lambda_\mathbf{2} + c_n\mathbf{A}\lambda_\mathbf{3} \qquad (4)$$

where $\lambda_\mathbf{j} = -\mathbf{v_j}/k_n$ for $1 \leq j \leq 3$. For all $1 \leq i < n$, we have

$$\begin{aligned} a_i\mathbf{A}\lambda_\mathbf{1} + b_i\mathbf{A}\lambda_\mathbf{2} + c_i\mathbf{A}\lambda_\mathbf{3} &= -\frac{a_i}{k_n}\mathbf{A}\mathbf{v_1} - \frac{b_i}{k_n}\mathbf{A}\mathbf{v_2} - \frac{c_i}{k_n}\mathbf{A}\mathbf{v_3} \\ &= -\frac{k_i}{k_n}\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \ . \end{aligned}$$

This implies that the last column of the $\mathbf{M_f}$ matrix of Eq. 3 is indeed a linear combination of the previous ones with coefficients $(\lambda_\mathbf{1}^{\mathrm{T}}, \lambda_\mathbf{2}^{\mathrm{T}}, \lambda_\mathbf{3}^{\mathrm{T}}, -k_1/k_n, ..., -k_{n-1}/k_n)$. Thus, in general, $\mathbf{M_f}$ has full rank minus 1.

Reconstructing the Whole Mesh If we now consider a mesh made of $n_v > 3$ vertices with a total of m correspondences, Eq. 3 becomes

$$\mathbf{M_m}\begin{pmatrix} \mathbf{v_1} \\ ... \\ \mathbf{v_{n_v}} \\ k_1 \\ ... \\ k_m \end{pmatrix} = \mathbf{0}, \quad \mathbf{M_m} = \left[\ \mathbf{M_L} \ \middle|\ \mathbf{M_R}\ \right] \ , \qquad (5)$$

with

$$\mathbf{M_L} = \begin{pmatrix} a_1\mathbf{A} & b_1\mathbf{A} & c_1\mathbf{A} & 0 & ... & ... \\ ... & ... & ... & ... & ... & ... \\ 0 & b_j\mathbf{A} & c_j\mathbf{A} & d_j\mathbf{A} & 0 & ... \\ ... & ... & ... & ... & ... & ... \\ a_l\mathbf{A} & 0 & c_l\mathbf{A} & 0 & e_l\mathbf{A} & ... \\ ... & ... & ... & ... & ... & ... \end{pmatrix} \text{ and}$$

$$\mathbf{M_R} = \begin{pmatrix} -\left(u_1 \ v_1 \ 1\right)^{\mathrm{T}} & 0 & ... & ... & ... \\ ... & ... & ... & ... & ... \\ 0 & -\left(u_j \ v_j \ 1\right)^{\mathrm{T}} & 0 & ... & ... \\ ... & ... & ... & ... & ... \\ 0 & ... & -\left(u_l \ v_l \ 1\right)^{\mathrm{T}} & 0 & ... \\ ... & ... & ... & ... & ... \end{pmatrix} .$$

Coefficients similar to those of Eq. 4 can be derived to compute $(u_m, v_m, 1)^{\mathrm{T}}$ as a linear combination of the non-zero columns of the last row. Since these coefficients only depend on k_m, on the mesh vertices and on the projection matrix, it can easily be checked that, as in the single triangle case, the last column of the matrix can be expressed as a linear combination of the others.

Thus matrix $\mathbf{M_m}$ of Eq. 5 has still full rank minus 1. This was to be expected and reflects the well-known scale ambiguity in monocular vision.

Representing the problem as in Eq. 5 was convenient to discuss the rank of the matrix. However, in practice, we want to recover the vertex coordinates but are not interested in having the k_i as unknowns. We therefore eliminate them by rewriting Eq. 5 as

$$\begin{pmatrix} a_1\mathbf{T_1} & b_1\mathbf{T_1} & c_1\mathbf{T_1} & 0 & ... & ... \\ ... & ... & ... & ... & ... & ... \\ 0 & b_j\mathbf{T_j} & c_j\mathbf{T_j} & d_j\mathbf{T_j} & 0 & ... \\ ... & ... & ... & ... & ... & ... \\ a_l\mathbf{T_l} & 0 & c_l\mathbf{T_l} & 0 & e_l\mathbf{T_l} & ... \\ ... & ... & ... & ... & ... & ... \end{pmatrix}\begin{pmatrix} \mathbf{v_1} \\ ... \\ \mathbf{v_{n_v}} \end{pmatrix} = \mathbf{0} \qquad (6)$$

with

$$\mathbf{T_i} = \mathbf{A}_{2\times 3} - \begin{pmatrix} u_i\mathbf{A}_3 \\ v_i\mathbf{A}_3 \end{pmatrix},$$

where $\mathbf{A}_3$ represents the last row of matrix $\mathbf{A}$ and $\mathbf{A}_{2\times 3}$ its first two rows. By construction, the matrix in Eq. 6 has the same rank as matrix $\mathbf{M_m}$, therefore the following results are valid for both representations of the problem.

Effective Rank In the previous paragraph, we showed that $\mathbf{M_m}$ has at most full rank minus one. However, this does not tell the whole story: In general, it is ill-conditioned and many of its singular values are small enough so that, in practice, it should be treated as a matrix of even lower rank. To illustrate this point, we projected randomly sampled points on the facets of the synthetic 88 vertices mesh of the top row of Fig. 2 using a known camera model. We then computed the singular values of matrix of Eq. 6, which we plot in Fig. 3.

In Fig. 4, we show the effect of adding two of the corresponding singular vectors—one associated to the zero singular value and the other to a small one—to the mesh in its reference position.

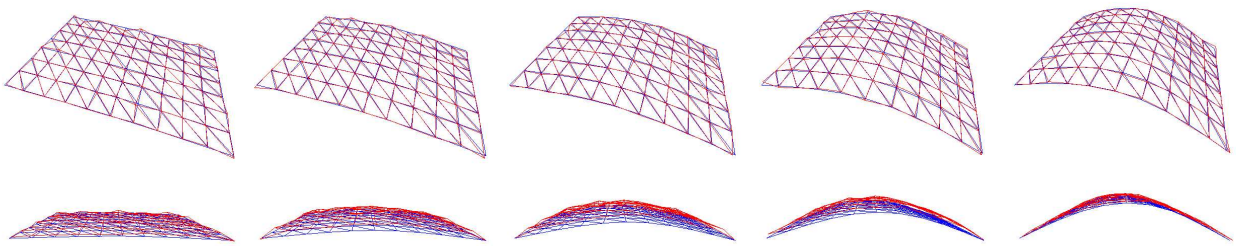


Figure 2. Reconstructing an 88-vertex mesh using perfect correspondences that were corrupted using zero-mean Gaussian noise with variance five, which is much larger than what can be expected of automated matching technique. **Top.** The original mesh and reconstructed one projected in the synthetic view used to create the correspondences. As expected, the projections match very closely. **Bottom.** The two meshes seen from a different viewpoint.

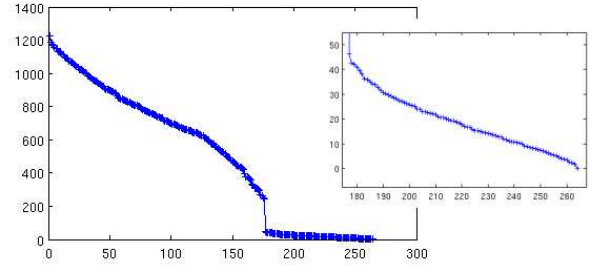
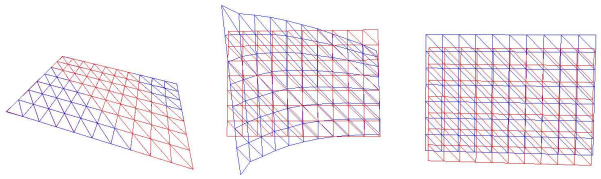


Figure 3. Singular values of the matrix of Eq. 6 for the 88 vertex mesh of Fig. 1. Note how the values drop down after the $2n_v = 176^{\text{th}}$ one, as predicted by the affine model of Section 3.2. The small graph on the right is a magnified version of the part of the graph containing the small singular values. The last one is zero up to the precision of the matlab routine used to compute it and the others are not very much larger.



(a) (b) (c)

Figure 4. Visualizing vectors associated to small singular values. (a) Reference mesh and mesh to which one the vectors has been added seen from the original viewpoint, in which they are almost indistinguishable. (b) The same two meshes seen from a different viewpoint. (c) The reference mesh modified by adding the vector associated to the zero singular value. Note that the resulting deformation is close to being a scaling.

Even though only one of these values is exactly zero, we can see that they drop down drastically after the first $2n_v = 176$. This shows that, even though the matrix may have full rank minus 1, the solution of the linear system would be very sensitive to noise. Therefore, in a real situation, we would actually be closer to having n_v ambiguities, which can be understood in terms of the piecewise affine model we introduce below.

3.2. Ambiguities under Piecewise Affine Projection

A piecewise affine camera model is one that involves an affine transform for each facet of the mesh. This approximation is warranted if the facets are small enough to neglect depth variations across them.

Projection of a 3–D Surface Point Let $\mathbf{x_i}$ be a 3–D point whose coordinates are again expressed in the camera referential. We write its projection to a 2–D image plane as

$$k \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \mathbf{P}\mathbf{x_i} \ , \ \mathbf{P} = \mathbf{A}' \left[\mathbf{I}_{2\times 2} \,\middle|\, \mathbf{0} \right] \ , \qquad (7)$$

where k is a depth factor associated to the affine camera and $\mathbf{A}'$ is a 2×2 matrix representing the internal parameters. As in Section 3.1, we study the ambiguities for a mesh containing first a single triangle and then many.

Reconstructing a Single Facet We can again write a linear system for a single triangle containing n 3–D to 2–D correspondences, with 3–D points given by their barycentric coordinates

$$\left(\begin{array}{ccc|c} a_1\mathbf{P} & b_1\mathbf{P} & c_1\mathbf{P} & -\begin{pmatrix} u_1 \\ v_1 \end{pmatrix} \\ \dots & \dots & \dots & \dots \\ a_i\mathbf{P} & b_i\mathbf{P} & c_i\mathbf{P} & -\begin{pmatrix} u_i \\ v_i \end{pmatrix} \\ \dots & \dots & \dots & \dots \\ a_n\mathbf{P} & b_n\mathbf{P} & c_n\mathbf{P} & -\begin{pmatrix} u_n \\ v_n \end{pmatrix} \end{array}\right) \begin{pmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \mathbf{v_3} \\ k \end{pmatrix} = \mathbf{0} \ . \qquad (8)$$

Since we only have one facet, we also only have one projection matrix, therefore a single k corresponding to the average depth of the facet is necessary, and all $(u_i, v_i)^{\mathrm{T}}$ can be put in the same column.

Since $\mathbf{P}$ is of size 2×3, it has at most rank 2. Moreover, we can show that the last column of the global matrix also is a linear combination of the two first columns of $\mathbf{P}$

$$\begin{aligned} \begin{pmatrix} u_i \\ v_i \end{pmatrix} &= \mathbf{P}\frac{1}{k}(a_i\mathbf{v_1} + b_i\mathbf{v_2} + c_i\mathbf{v_3}) \\ &= \left[\mathbf{A}' \,\middle|\, \mathbf{0}\right] \frac{1}{k}(a_i\mathbf{v_1} + b_i\mathbf{v_2} + c_i\mathbf{v_3}) \\ &= \frac{a_i}{k}\mathbf{A}' \begin{pmatrix} \mathbf{v_1}_1 \\ \mathbf{v_1}_2 \end{pmatrix} + \frac{b_i}{k}\mathbf{A}' \begin{pmatrix} \mathbf{v_2}_1 \\ \mathbf{v_2}_2 \end{pmatrix} + \frac{c_i}{k}\mathbf{A}' \begin{pmatrix} \mathbf{v_3}_1 \\ \mathbf{v_3}_2 \end{pmatrix} \ . \end{aligned} \qquad (9)$$

The coefficients of Eq. 9 are independent of the correspondence considered and are therefore valid for any row i of the matrix. This finally means that, when $n \geq 3$, the rank of the matrix of Eq. 8 is always 6.

Reconstructing the Whole Mesh As discussed above, when there are several triangles, using the piecewise affine model amounts to introducing a projection matrix per facet. However, since in reality we only have one camera, its internal parameters, rotation matrix, and center are bound to be the same for each triangle. This only lets us with a variable depth factor k_i for each facet i among the n_f facets of the mesh. We can then write the system

$$\mathbf{M}'_{\mathbf{m}} \left({\mathbf{v_1}}^{\mathrm{T}} \ldots {\mathbf{v_{n_v}}}^{\mathrm{T}} \; k_1 \ldots k_{n_f} \right)^{\mathrm{T}} = \mathbf{0}, \quad \mathbf{M}'_{\mathbf{m}} = \left[\, \mathbf{M}'_{\mathbf{L}} \mid \mathbf{M}'_{\mathbf{R}} \, \right] \tag{10}$$

with

$$\mathbf{M}'_{\mathbf{m}} = \left(\begin{array}{cccccc|cccccc} a_1\mathbf{P} & b_1\mathbf{P} & c_1\mathbf{P} & 0 & \ldots & \ldots & -\binom{u_1}{v_1} & 0 & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & b_j\mathbf{P} & c_j\mathbf{P} & d_j\mathbf{P} & 0 & \ldots & 0 & -\binom{u_j}{v_j} & 0 & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_l\mathbf{P} & 0 & c_l\mathbf{P} & 0 & e_l\mathbf{P} & \ldots & 0 & \ldots & -\binom{u_l}{v_l} & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{array} \right) .$$

$\mathbf{M}'_{\mathbf{L}}$, which is of size $2m \times 3n_v$, m being the number of correspondences, has at most rank $2n_v$ because $\mathbf{P}$ has rank 2. Similarly, $\mathbf{M}'_{\mathbf{R}}$, which is of size $2m \times n_f$, has at most rank $n_f - 1$, because we can again show that its last column is a linear combination of the previous one in a similar manner as was done for the perspective case, with the coefficients of Eq. 9. This means that for a full mesh, $\mathbf{M}'_{\mathbf{m}}$ has at most rank $2n_v + n_f - 1$. This leaves us with $n_v + 1$ ambiguities. This again seems natural due first to the same scale ambiguity as in the perspective case, and second to the fact that now each vertex is free to move along the line of sight. This number corresponds to the number observed in the perspective case of Section 3.1, except that, in the affine case, a global scale is different from all vertices sliding along the line of sight, which produces an extra singular value.

4. Weak but Broadly Applicable Constraints

Since the linear systems of Section 3 are rank-deficient, we need to introduce additional constraints to obtain acceptable solutions. In essence, this is what all the model-based methods discussed in Section 2 do. However, they typically involve very specific assumptions about either the physical properties or the range of possible deformations of the surfaces at hand, which is very restrictive.

In this section, we show that a much weaker and more broadly applicable set of constraints suffices: Since we deal with video sequences, we can assume that the surface does not move randomly between two frames, whatever the physical properties of the target surface. We therefore perform the reconstruction over several frames simultaneously and simply limit the range of motion from frame to frame.

We show here that this can be expressed as a set of additional linear constraints that make our linear systems well-conditioned, first in the affine case and then in the projective one.

4.1. Constraining the Affine Reconstruction

Given a temporal sequence of n_I images and the corresponding matrices ${\mathbf{M}'_{\mathbf{m}}}^{t} \; , \; 1 \leq t \leq n_I$ of Eq. 10, we can create a block diagonal matrix whose blocks are the ${\mathbf{M}'_{\mathbf{m}}}^{t}$ and use it to write a big linear system that the vertex coordinates in all frames must satisfy simultaneously. However, without temporal consistency constraints, the ambiguities remain: As discussed in Section 3.2, when the camera coordinates are aligned with the world coordinates, reconstruction is only possible up to an unknown motion along the z-axis for each vertex at each time step. To mitigate this problem, it is therefore natural to link the z value of vertices across time. The simplest way to do this is to write

$$\mathbf{v}_z^{t+1} - \mathbf{v}_z^t = 0 \tag{11}$$

for all vertices and all times. These constraints and those imposed by the 3–D to 2–D correspondences can then be imposed simultaneously by solving with respect to Θ

$$\mathbf{M_s}\Theta = \mathbf{b} \;\; , \tag{12}$$

where

$$\begin{aligned} \Theta &= \left({\mathbf{v}_1^1}^{\mathrm{T}} \ldots {\mathbf{v}_{n_v}^1}^{\mathrm{T}} \; k_1^1 \ldots k_{n_f}^1 \ldots {\mathbf{v}_1^{n_I}}^{\mathrm{T}} \ldots {\mathbf{v}_{n_v}^{n_I}}^{\mathrm{T}} \; k_1^{n_I} \ldots k_{n_f}^{n_I} \right)^{\mathrm{T}} , \\ \mathbf{M_s} &= \left[\begin{array}{c} \mathbf{M_U} \\ \mathbf{M_B} \end{array} \right] , \; \mathbf{b} = \left(\mathbf{0} \; z_1^{first} \ldots z_{n_v}^{first} \; \mathbf{0} \; z_1^{last} \ldots z_{n_v}^{last} \right)^{\mathrm{T}} , \\ \mathbf{M_U} &= \left(\begin{array}{cccccc} {\mathbf{M}'_{\mathbf{m}}}^{1} & \mathbf{0} & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \mathbf{0} & {\mathbf{M}'_{\mathbf{m}}}^{t} & \mathbf{0} & \ldots & \ldots \\ \ldots & \ldots & \mathbf{0} & {\mathbf{M}'_{\mathbf{m}}}^{t+1} & \mathbf{0} & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \mathbf{0} & {\mathbf{M}'_{\mathbf{m}}}^{n_I} \end{array} \right) , \\ \mathbf{M_B} &= \left(\begin{array}{cccccc} \mathbf{C} & \mathbf{0} & \ldots & \ldots & \ldots & \ldots \\ -\mathbf{C} & \mathbf{C} & \mathbf{0} & \ldots & \ldots & \ldots \\ \mathbf{0} & -\mathbf{C} & \mathbf{C} & \mathbf{0} & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \mathbf{0} & \ldots & \ldots & \mathbf{0} & -\mathbf{C} & \mathbf{C} \\ \mathbf{0} & \ldots & \ldots & \ldots & \mathbf{0} & \mathbf{C} \end{array} \right) , \end{aligned}$$

z_i^{first} and z_i^{last} are the z-coordinate of vertex i in the first and last frames, in which we assume that the shape is known, and $\mathbf{C}$ is an $n_v \times 3n_v$ matrix containing a single 1 in each row, which corresponds to the z-coordinate of one vertex.

The number of constraints we add in this manner is equal to the number n_v of ambiguities that we derived in Section 3.2. Therefore it affects the rank of $\mathbf{M_s}$, and reduces the number of ambiguities to zero as shown in Fig. 5. Moreover, these constraints do not overlap with the ones imposed by the correspondences and can then be considered as minimal.

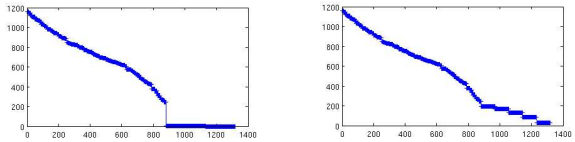


Figure 5. Singular values for a 5 frames sequence under affine projection. **Left** Without temporal consistency constraints between frames, the linear system has many zero singular values, which implies severe reconstruction ambiguities. **Right** Constraining the z coordinates as discussed in Section 4.1 leaves the non zero singular values unchanged but increase the value of the others, thus removing the ambiguities.

In practice the correspondences are never perfect and include noise and outliers. We therefore solve Eq. 12 in the least-squares sense and take Θ to be

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \, (\mathbf{M_s}\Theta - \mathbf{b})^{\mathrm{T}}\mathbf{W}(\mathbf{M_s}\Theta - \mathbf{b}) \ , \qquad (13)$$

where $\mathbf{W}$ is a diagonal matrix of ones for the lines corresponding to projection constraints and a user-defined weight for those that correspond to the depth constraints. The weight is designed to give comparable influence to both classes of constraints and directly affects how much the small singular values increase.

4.2. Constraining the Perspective Reconstruction

In Section 3.2, we showed that ambiguities under perspective projection are similar to those under piecewise affine projection. It is therefore natural to constrain the reconstruction in a similar way, that is by limiting the motion along the line-or-sight. However, since it is not parallel to the z-axis anymore, the constraints become more difficult to express.

Let us consider one vertex $\mathbf{v}$ of the mesh at times t and $t+1$. We can try minimizing $d = \mathbf{v}^t\mathbf{v}^{t+1} \cdot \mathbf{e}$, the length of the projection on the line-of-sight of the $\mathbf{v}^t\mathbf{v}^{t+1}$ vector, where $\mathbf{e}$ is the vector $\mathbf{c}\mathbf{v}^t$ after normalization, and $\mathbf{c}$ represents the optical center of the camera. The difficulty comes from the fact that this constraint is nonlinear and can therefore not be introduced into our linear formulation. We overcome this problem by replacing the exact formulation of d by an upper bound that can be expressed linearly as follows:

$$\begin{aligned}
d^2 &= (\mathbf{v}^t\mathbf{v}^{t+1} \cdot \mathbf{e})^2 \ , \\
&= (\mathbf{e}_x(x_c^{t+1} - x_c^t) + \mathbf{e}_y(y_c^{t+1} - y_c^t) + \mathbf{e}_z(z_c^{t+1} - z_c^t))^2 \ , \\
&\leq (\mathbf{e}_x(x_c^{t+1} - x_c^t))^2 + (\mathbf{e}_y(y_c^{t+1} - y_c^t))^2 + (\mathbf{e}_z(z_c^{t+1} - z_c^t))^2 \\
&\quad + (\sqrt{2}\mathbf{e}_x(x_c^{t+1} - x_c^t))^2 + (\sqrt{2}\mathbf{e}_y(y_c^{t+1} - y_c^t))^2 \\
&\quad + (\sqrt{2}\mathbf{e}_x(x_c^{t+1} - x_c^t))^2 + (\sqrt{2}\mathbf{e}_z(z_c^{t+1} - z_c^t))^2 \\
&\quad + (\sqrt{2}\mathbf{e}_y(y_c^{t+1} - y_c^t))^2 + (\sqrt{2}\mathbf{e}_z(z_c^{t+1} - z_c^t))^2 \ , \\
&\leq (\sin(\theta_x^{max})(1 + 2\sqrt{2})(x_c^{t+1} - x_c^t))^2 \\
&\quad + (\sin(\theta_y^{max})(1 + 2\sqrt{2})(y_c^{t+1} - y_c^t))^2 \\
&\quad + ((1 + 2\sqrt{2})(z_c^{t+1} - z_c^t))^2 \ , \qquad (14)
\end{aligned}$$

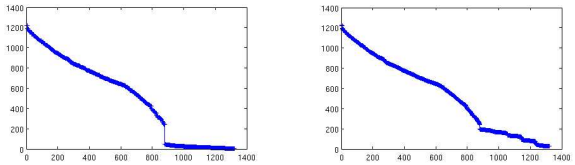


Figure 6. Singular values for a 5 frames sequence under perspective projection. **Left** Without temporal consistency constraints between frames, the linear system is rank-deficient. **Right** Bounding the frame-to-frame displacements along the line of sight using the linear expression of Eq. 14 transforms the ill-conditioned linear system into a well-conditioned one. The smaller singular values have increased and are now clearly non-zero. Since our motion model introduces more equations than strictly necessary, the other values are also affected, but only very slightly.

where x_c, y_c and z_c are the coordinates of a vertex in the camera reference system, and θ_x^{max} and θ_y^{max} are the maximum angles between the camera center and the points projecting on the left/right, and upper/lower border of the image, respectively.

As in Section 4.1, these constraints and those imposed by the 3–D to 2–D correspondences can be imposed simultaneously. We rewrite Eq. 12 by replacing the ${\mathbf{M}'_{\mathbf{m}}}^t$ matrices of Eq. 10 by the matrix of Eq. 6 and the $\mathbf{C}$ matrices by $3n_v \times 3n_v$ matrices, containing a single value in each row that will constrain the x-, y-, or z-coordinate of one vertex. This value is set to one of the three coefficients of Eq. 14, depending on which coordinate the row corresponds to.

Fig. 6 shows how the singular values of the system are affected by introducing our depth constraints. As in the affine case, we can see that the smaller singular values have increased and now clearly different from zero. Since this was our only goal in adding constraints, this justifies our approach to liberalization by minimizing the upper bound of d of Eq. 14 instead of d itself. Note that because we added more equations than was strictly necessary, the other singular values also increased, but only very slightly.

5. Experiments

In the previous sections, we developed theoretical basis for reconstructing the shape of a deformable surface from 3–D to 2–D correspondences in a video sequence. We showed that constraining the variations in depth from frame to frame is sufficient, in theory, to formulate the reconstruction problem in terms of solving a well-conditioned linear system. In this section, we show that this indeed produces valid reconstructions in practice.

We present results obtained using both synthetic data and real images. In both cases, the deformations of the meshes were retrieved by solving the linear system of Section 4 for whole sequences with known deformations in the first and last frames. This was done using `Matlab`'s implementation of sparse matrices and resolution of linear systems

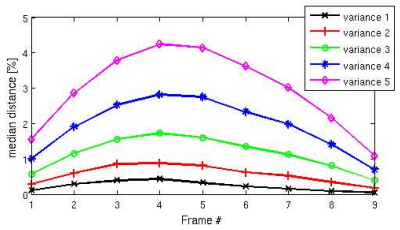


Figure 7. Distance between the original mesh and its reconstruction for each one of the 9 deformed versions of the mesh of Fig. 2. We plot five curves corresponding to vertex-to-surface distances obtained with variance one to five gaussian noise on the correspondences. The distances are expressed as percentages of the length of the mesh largest side.

with known covariance matrix in the least square sense. In our experiments, the covariance matrix simply is the weight matrix of Eq. 13, which weighs differently the correspondences equations and the constraints.

5.1. Synthetic Data

We deformed the 88-vertex mesh of Fig. 4(a) to produce 9 different shapes and 9 corresponding sets of 3–D to 2–D correspondences using a perspective projection matrix. We then added Gaussian noise with mean zero and variance ranging from one to five to the image locations of these correspondences. Fig. 2 depicts the reconstruction results overlaid on the original mesh with noise variance five. The differences are hard to see, even though this represents far lower precision than what can be expected of good feature-point matching algorithms.

To quantify the differences between the meshes, we plot the distances between the two meshes in Fig. 7 for each one of 9 different shapes, given increasing noise variance. The distances are expressed as percentages of the mesh largest side. With a noise variance one, they are of the order of 0.25% for vertex-to-surface distance, which works out to 0.025cm for a 10cm$\times$7cm mesh. This is very small given that we incorporate very little *a priori* knowledge into our reconstruction algorithm.

5.2. Real Data

We now present results on two real monocular video sequences acquired with an ordinary digital camera. The longest one is 250 frames long, which shows that, even though our approach involves solving a very large system, it is sparse enough to use a standard `Matlab` routine. In both cases, we automatically establish 3–D to 2–D correspondences between the first frame, where the 3–D pose is assumed to be known, and the others by first tracking the surface in 2–D using normalized cross-correlation. We then compute correspondences by picking 10 random samples in each facet and looking in each frame in an area limited by the 2–D tracking result for 2–D points matching their projections in the first frame. To this end we use standard cross-correlation, which results in noisy correspondences with a number of mismatches at places where there is not enough texture to guarantee reliable matches.

Fig. 8 depicts our reconstruction results for a relatively inelastic piece of paper in a 250-frame sequence and Fig. 9 those for a much more flexible sheet of plastic in a 147-frame sequence. In both cases, the global shape is correct, which confirms that the ambiguities have been correctly handled. However, because we impose no smoothness constraint of any kind, there are also local errors that are caused by the mismatches present in our input data. If the goal were to derive a perfect shape from a set of noisy correspondences, we could mitigate the effect of erroneous matches by introducing a robust estimator into the least-squares minimization of Eq. 13. However, we will argue in the following section that this may not actually be necessary for the application we have in mind.

Since our technique does not introduce any prior on the physical properties of the target surface, we were able to reconstruct both the paper and plastic without changing anything to our system. It is not clear that this would have been the case had we used a physics-based approach or any other that implicitly limits the range of deformation of the surface.

6. Conclusion

In this paper we have presented a theoretical study of the ambiguities that arise when reconstructing deformable 3–D surfaces from monocular video sequences. We showed that they can be interpreted in terms of those inherent to a piecewise affine model and can be removed by simply constraining the frame-to-frame variation in depth. These are very weak constraints that are broadly applicable because they do restrict the range of possible surface deformations.

When used in conjunction with real correspondences, including noise and outright mismatches, these constraints are sufficient to recover the surface, not perfectly, but with good accuracy nevertheless. More specifically, we do not smooth our results at all because it would defeat our basic purpose, which is to introduce as little *a priori* knowledge of the surface's physical properties as possible. As a result, our reconstructions may contain local deviations from the true surface. However, we do not believe this to be a major issue given our ultimate purpose: If the goal is to track many surfaces to create a motion database from which a motion model can be learned, the deviations can be treated as random perturbations that will be eliminated when observing a large number of sequences. Proving this to be the case will be the focus of our future work.

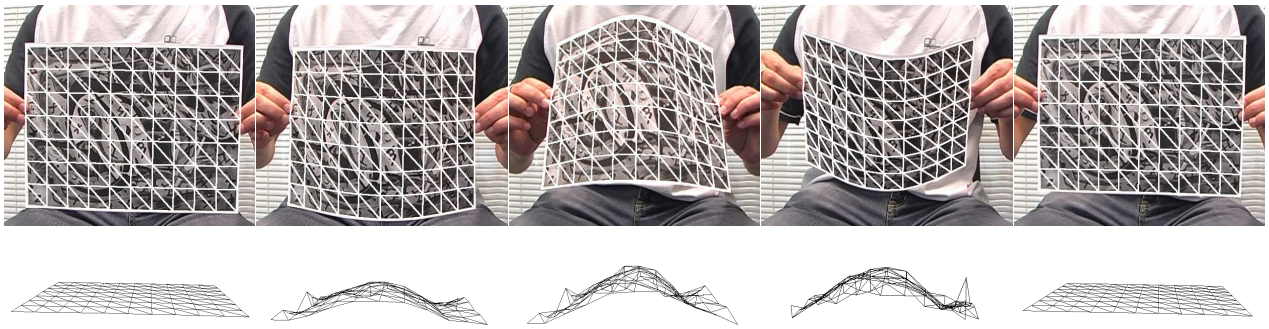


Figure 8. Reconstructing a deforming sheet of a paper from a 250-frames sequence. **Top** The reconstructed mesh is reprojected into the original images and closely matches the outline of the paper. **Bottom** The same mesh seen from the side. In spite of local inaccuracies in depth, the overall shape is correct, which indicates that the ambiguities have been successfully resolved. A complete video is submitted as supplementary material.

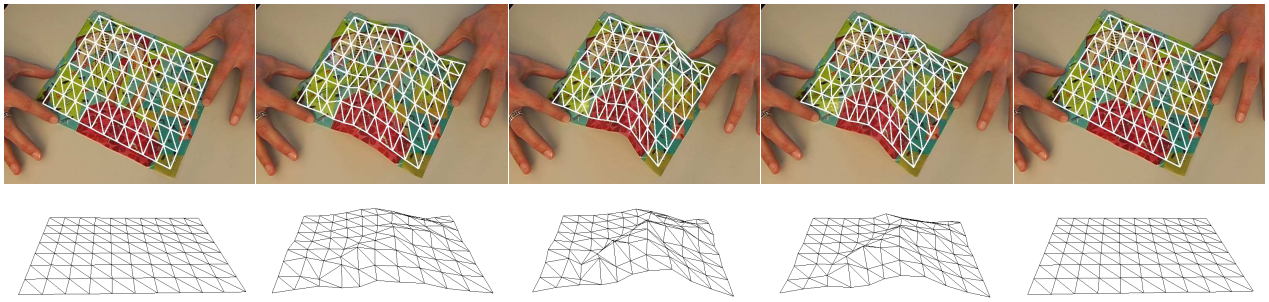


Figure 9. Reconstruction results for a plastic sheet, which is much more flexible than the sheet of paper of Fig. 8. In spite of this, the overall shape is again correctly recovered up to small errors due to erroneous correspondences. A complete video is submitted as supplementary material.

References

[1] A. Bartoli and A. Zisserman. Direct Estimation of Non-Rigid Registration. In *British Machine Vision Conference*, Kingston, UK, September 2004.

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

[3] L. Cohen and I. Cohen. Deformable models for 3-d medical images using finite elements and balloons. In *Conference on Computer Vision and Pattern Recognition*, pages 592–598, 1992.

[4] T. Cootes, G. Edwards, and C. Taylor. Active Appearance Models. In *European Conference on Computer Vision*, pages 484–498, Freiburg, Germany, June 1998.

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

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

[7] X. Llado, A. D. Bue, and L. Agapito. Non-rigid 3D Factorization for Projective Reconstruction. In *British Machine Vision Conference*, Oxford, UK, September 2005.

[8] I. Matthews and S. Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60:135–164, November 2004.

[9] T. McInerney and D. Terzopoulos. A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking. In *International Conference on Computer Vision*, pages 518–523, Berlin, Germany, 1993.

[10] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

[11] A. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4(2):107–126, 1990.

[12] M. Salzmann, S. Ilić, and P. Fua. Physically Valid Shape Parameterization for Monocular 3–D Deformable Surface Tracking. In *British Machine Vision Conference*, Oxford, UK, September 2005.

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

[14] R. Urtasun, D. Fleet, A. Hertzman, and P. Fua. Priors for people tracking from small training sets. In *International Conference on Computer Vision*, Beijing, China, October 2005.

[15] R. White and D. Forsyth. Combining cues: Shape from shading and texture. In *Conference on Computer Vision and Pattern Recognition*, 2006.

[16] J. Xiao and T. Kanade. Uncalibrated perspective reconstruction of deformable structures. In *International Conference on Computer Vision*, 2005.