Monocular Template-based Reconstruction of Inextensible Surfaces

Mathieu Perriollat[1] **Richard Hartley**[2] **Adrien Bartoli**[3]

[1] VI-Technology, Grenoble, France

[2] RSISE, ANU / NICTA*, Canberra, Australia

[3] Université d'Auvergne, Clermont-Ferrand, France

Adrien.Bartoli@gmail.com

Abstract

We present a monocular 3D reconstruction algorithm for inextensible deformable surfaces. It uses point correspondences between a single image of the deformed surface taken by a camera with known intrinsic parameters and a template. The main assumption we make is that the surface shape as seen in the template is known. Since the surface is inextensible, its deformations are isometric to the template. We exploit the distance preservation constraints to recover the 3D surface shape as seen in the image. Though the distance preservation constraints have already been investigated in the literature, we propose a new way to handle them. Spatial smoothness priors are easily incorporated, as well as temporal smoothness priors in the case of reconstruction from a video. The reconstruction can be used for 3D augmented reality purposes thanks to a fast implementation. We report results on synthetic and real data. Some of them are compared to stereo-based 3D reconstructions to demonstrate the efficiency of our method.

1 Introduction

Recovering the 3D shape of a deformable surface from a monocular video and a template (a 'reference' image of the surface) is a challenging problem, illustrated in figure 1. This problem has been addressed by researchers over the past few years and several algorithms have been proposed. The 3D shape as seen in the template is usually known. The problem of recovering the 3D shape as seen in the image is ill-posed due to depth ambiguities. Additional consistency constraints are thus required. Most commonly, *ad hoc* constraints are used. These include spatial and temporal surface smoothness (Gumerov et al., 2004; Prasad et al., 2006), the low-rank shape model (Bregler et al., 2000) and combinations of those (Bartoli et al., 2008; Del Bue, 2008).

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

We propose an algorithm dedicated to inextensible surfaces such as those shown in figure 2. It uses point correspondences to compute upper bounds on the points' depth using the surface inextensibility assumption. We show that these bounds directly provide a 'good' 3D reconstruction of the surface. As opposed to algorithms that iteratively refine an initial solution that must be 'close' to the optimal one, ours is standalone and easily handles additional constraints such as spatial and temporal smoothness. Our method was first published in a short version of this paper (Perriollat et al., 2008). The closest works to ours are (Ecker et al., 2008; Ferreira et al., 2009; Penna, 1992; Salzmann et al., 2008a). Our method improves the state of the art since it does not make any assumption about the surface deformation. It reconstructs an inextensible surface from a template and a single image[1] of the deformed surface from point correspondences only, though we demonstrate it on both single-image and video datasets. Our algorithm is simple and fast, and can therefore be used to provide a good initialization to local iterative algorithms.

This paper is organized as follows. Related work on monocular deformable reconstruction is reviewed in §2. The evaluation of upper bounds is presented in §3 and the surface recovery procedure in §4. An experimental study of the reconstruction error with simulated data is proposed in §5. Results on real datasets are reported in §6. Eventually, we give our conclusion and research perspectives in §7.

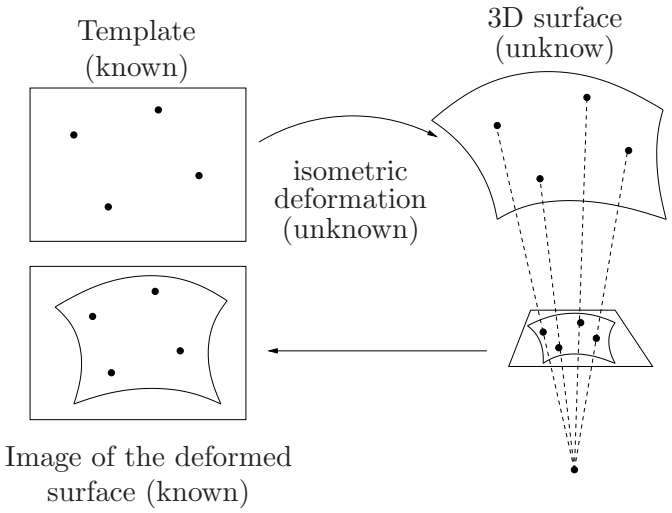


Figure 1: **Template-based monocular reconstruction of a deformable surface: problem setup.** We are given a *template* for which the surface shape is known, and an *image* for which the surface shape has to be reconstructed, through the estimation of an *isometric deformation* from the template surface shape to the image surface shape. Point (typically keypoint) correspondences between the template and the image are used. The template is often flat, but not always. As an example, the can shown in figure 2 (b) has a cylindrical template shape. An isometric deformation of the cylinder to 'explain' the image of the deformed shape is sought.

2 State of the Art

There are three main components used for monocular deformable scene reconstruction in the literature: the general *low-rank shape* model, the assumption that the object of interest is a *surface* and the knowledge of a *template*.

[1]For which the camera intrinsic parameters are known.

(a) (b)

Figure 2: **Examples of images from which our algorithm can reconstruct the 3D surface shape.** (a) Examples of a paper sheet: the template (left) and two deformed sheets, a smooth one (middle) and a creased one (right). (b) Example of a can: template image (left), the 3D shape associated with the template (middle), and the input image showing the can deformed (right). The reconstructions are shown in §6, figures 12, 13 and 14.

These components can be independently used or combined together, so as to handle the intrinsic ambiguities in monocular deformable reconstruction. *Surface* is a fairly broad term which in the context of this paper means a smooth (continuous and differentiable) shape. There also exist work on surface tracking, that can be used to provide input data to surface reconstruction methods, such as (Shen et al., 2009) that does inextensible surface tracking, and (Gay-Bellile et al., 2009) that uses a self-occlusion resistant smoothness constraint.

The low-rank factorization solution to the non-rigid shape recovery problem has been introduced by (Bregler et al., 2000) and used in (Bartoli et al., 2008; Brand, 2005; Del Bue, 2008; Olsen and Bartoli, 2008; Torresani et al., 2008; Vidal and Abretske, 2006; Xiao and Kanade, 2006). The 3D object shape is represented by a linear combination of unknown basis shapes. The algorithm recovers both the basis shapes and the configuration weights. The surface hypothesis has recently been incorporated in this framework through the use of priors (Bartoli et al., 2008; Del Bue, 2008; Olsen and Bartoli, 2008). These methods are batch: they need the whole video to compute a solution and are thus not suited for reconstruction on the fly.

Learning approaches have proven efficient to model deformable objects (Gay-Bellile et al., 2006; Salzmann et al., 2007, 2008b). The main drawback is the lack of generality when the trained model is too specific. So as to properly deal with videos, temporal consistency is used to smooth the deformations. The initial 3D shape *i.e.*, the surface shape for at least one frame of the video, must be known. These methods usually also need a template. In practice, the initial 3D shape and template constraints are met by acquiring the video such that the object deformation in the first frame is close to the one seen in the template.

Methods using only the surface assumption have been proposed. They require strong additional priors on the surface such as its developability, applicable to paper sheets. One of the motivations for these methods is to perform paper scanning from images of deformed paper sheets. For this kind of application, a template is obviously not available. Under the surface smoothness assumption, (Gumerov et al., 2004) solve a system of differential equations on the page borders to obtain the 3D shape. Other approaches such as (Liang et al., 2006) use textual information to evaluate the surface parameters. These methods perform well on smoothly bent paper but cannot be extended to

arbitrary inextensible objects.

The method we propose is dedicated to inextensible surfaces (that deform isometrically), uses a template and assumes the internal parameters of the camera to be known. The method in (Salzmann et al., 2008a) uses the same hypotheses as our method, but solves the problem differently, using Extended Linearization. The method in (Ferreira et al., 2009), however, is different in that it uses a scaled orthographic camera model, does not use a template, but multiple images so as to unfold the surface, thereby reconstructing the template itself. The method in (Ecker et al., 2008) also uses an affine approximation, more precisely, an orthographic camera model. These two methods enforce the inextensiblity constraints exactly. Finally, the method in (Penna, 1992) makes similar hypotheses to ours but also requires partial derivatives around the point correspondences. This extra piece of information is rarely available in practice.

In summary, our approach is more flexible than the others in several respects: it applies to any inextensible surface such as paper, garment or faces[2] and it uses only one frame to compute the reconstruction. When processing a video, it does not need that the 3D surface is known in advance for a particular, reference frame.

3 Finding Upper Bounds on the Surface Depth

We focus on inextensible deformable objects imaged by projective cameras. A surface template is assumed to be known. We describe our algorithm to compute upper bounds on the depth of the surface points. We first give the principle, then the bound initialization and finally their iterative refinement. Our notation is shown in table 1.

$\mathcal{T}$	template
$q_i^{\mathcal{T}}$	point i in the template
$\mathcal{I}$	image of the deformed object
P	camera matrix for $\mathcal{I}$
C	camera centre for $\mathcal{I}$
$q_i^{\mathcal{I}}$	point i in the image
S_i	sightline for point $q_i^{\mathcal{I}}$
v_i	direction of the sightline S_i
α_{ij}	the angle between S_i and S_j
$\bar{q}_i$	point i in homogeneous coordinates
$\|.\|$	vector two-norm

$d_{ij} = d_{geo}(q_i^{\mathcal{T}}, q_j^{\mathcal{T}})$	geodesic distance between $q_i^{\mathcal{T}}$ and $q_j^{\mathcal{T}}$
μ_i	depth of point i
$Q_i = Q_i(\mu_i)$	3D point i
$\hat{\mu}_i$	true depth of point i
$\hat{Q}_i$	true 3D point i
$\tilde{\mu}_i$	reconstructed depth of point i
$\tilde{Q}_i$	reconstructed 3D point i
$i^\star$	index of the point constraining the depth of point i
$\breve{\mu}_i = \breve{\mu}_{ii^\star}$	maximal depth of point i
$\breve{Q}_i$	deepest 3D point i

Table 1: Our notation for this paper.

We assume that the template is composed of the 3D surface shape registered with an image of the object. Examples are shown in figure 2. For the paper sheets, the reference shape is a plane, and for the can, it is an open cylinder. Assuming that point correspondences are established between the image of the deformed object and the template, we show that the region of space containing the object is bounded. The internal camera parameters allow

[2]See (Bronstein et al., 2005) for more details on the 3D geometric properties of faces.

one to compute the backprojection of the matched feature points, known as *sightlines*. Since the camera is projective, the sightlines intersect at the camera center and are not parallel to each other. The consequence is that the distance between two points increases with their depths. The template gives the maximal distance between two points. when the real dimensions of the template are available, the 3D scale ambiguity is resolved. This is used to compute the maximal depth of the points.

First of all, correspondences are established between the image and the template using for instance KLT (Shi and Tomasi, 1994) or SIFT (Lowe, 2004), or a detection / tracking process designed for deformable objects (Gay-Bellile et al., 2009; Pilet et al., 2008). We assume that there is no point mismatches. The upper bounds are evaluated through a two step algorithm:

1. **Initialization.** (§3.1) A suboptimal solution is computed by using pairwise constraints.

2. **Refinement.** (§3.2) An iterative refinement process considers the upper bounds as a whole and tunes all of them to get a fully compatible set of bounds.

3.1 Initializing the Upper Depth Bounds

An initialization for the bounds is computed by considering the point correspondences pairwise. Two points and the inextensibility constraint are sufficient to bound the depth of these two points along their sightlines. For n correspondences, $n-1$ bounds are obtained for each point. The most restrictive bound (*i.e.*, the tightest one) is kept for each point. The sightlines are computed in the image of the deformed object $\mathcal{I}$ (details can be found in *e.g.* (Hartley and Zisserman, 2004)). The camera matrix $P = [M|p_4]$ is composed of a (3×3) matrix M and a (3×1) vector p_4. The camera center is $C = -M^{-1}p_4$. It will be seen that it is convenient to set the arbitrary world coordinate frame aligned with the camera coordinate frame, and we therefore set $C = 0$. The vector v_i orienting the sightline passing through the point $q_i^{\mathcal{I}}$ is:

$$v_i = \frac{M^{-1}\tilde{q}_i^{\mathcal{I}}}{\|M^{-1}\tilde{q}_i^{\mathcal{I}}\|}.$$

A 3D point Q_i on the sightline S_i can be parametrized by:

$$Q_i(\mu_i) = \mu_i \; v_i + C = \mu_i \; v_i.$$

The camera position and orientation for the image can in practice be chosen arbitrarily. This is thanks to the problem setup illustrated in figure 1. Indeed, placing the camera differently entails applying a Euclidean transformation. Since what we are computing is an isometric deformation of the surface, a Euclidean transformation has no influence on it.

The depth μ_i is the distance of the point to the camera center; it is positive (Hartley, 1998). As figure 3 illustrates, the inextensibility of the object gives the following constraint between the points: whatever the actual deformation, the Euclidean distance between two 3D points is lower than or equal to the geodesic distance between them on the

template:

$$\|\hat{Q}_i \ - \hat{Q}_j\| \leq d_{geo}(q_i^{\mathcal{T}}, q_j^{\mathcal{T}}) = d_{ij}.$$

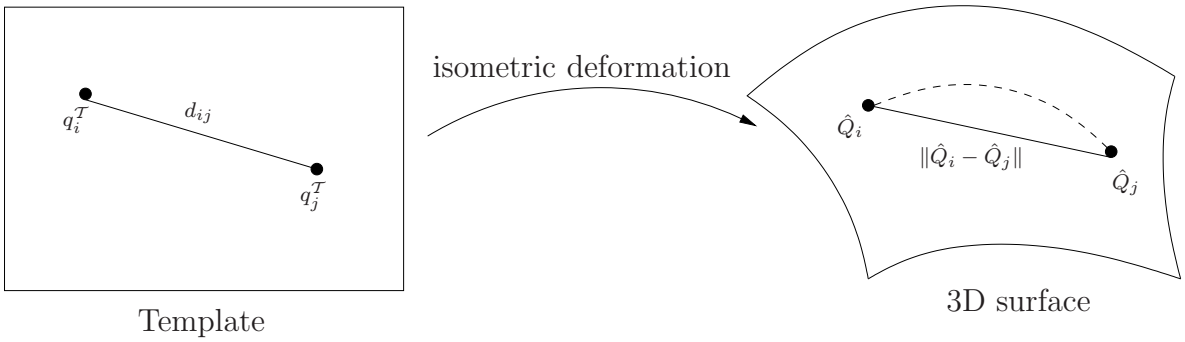


Figure 3: **Inextensible object deformation.** The template surface shape is deformed to the 3D surface by an unknown isometric transformation. The dashed line is the geodesic curve between $\hat{Q}_i$ and $\hat{Q}_j$. It has the same length d_{ij} as the known geodesic distance in the template. The Euclidean distance between the 3D points $\hat{Q}_i$ and $\hat{Q}_j$ is shorter than d_{ij} due to the surface deformation.

As figure 4 illustrates, the coordinate frame can be chosen such that:

$$Q_i = \begin{pmatrix} \mu_i \\ 0 \\ 0 \end{pmatrix} \qquad Q_j = \begin{pmatrix} \mu_j \ \cos(\alpha_{ij}) \\ \mu_j \ \sin(\alpha_{ij}) \\ 0 \end{pmatrix}.$$

Given μ_i, the two candidate points for Q_j such that $\|Q_i - Q_j\|$ equals d_{ij} are given by:

$$\mu_j\,(\mu_i) = \mu_i \ \cos(\alpha_{ij}) \pm \sqrt{d_{ij}^2 - \mu_i^2 \ \sin^2(\alpha_{ij})}, \tag{1}$$

where $\mu_j\,(\mu_i)$ gives the depth of the j-point as a function of the depth μ_i of the i-th point. So there exists a real solution if and only if:

$$\mu_i \leq \sqrt{\frac{d_{ij}^2}{\sin^2(\alpha_{ij})}}.$$

The upper bound $\breve{\mu}_i$ is then computed from the whole set of correspondences (we assume $\alpha_{ij} \leq \frac{\pi}{2}$ which holds with most of the common lenses):

$$\breve{\mu}_i = \breve{\mu}_{ii^\star} = \min_{\substack{j\,=\,1..n \\ j\,\neq\,i}} \left(\frac{d_{ij}}{\sin(\alpha_{ij})}\right).$$

The point that induces the minimum upper bound has index $i^\star$. We refer to this point $i^\star$ as the *anchor point* of point i. Note that the 'anchor' property is not symmetric: the anchor point of $i^\star$ is not necessarily i. It is one of the reasons why this initialization is suboptimal, as explained in the next paragraph.

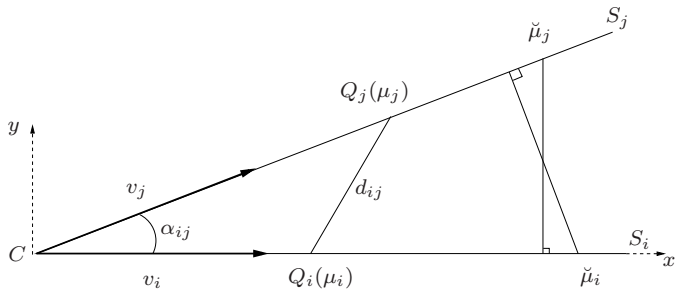


Figure 4: **Point parametrization along the sightlines.** Point Q_i is parametrized by its depth μ_i along the sightline passing through the camera projection center C and with direction S_i.

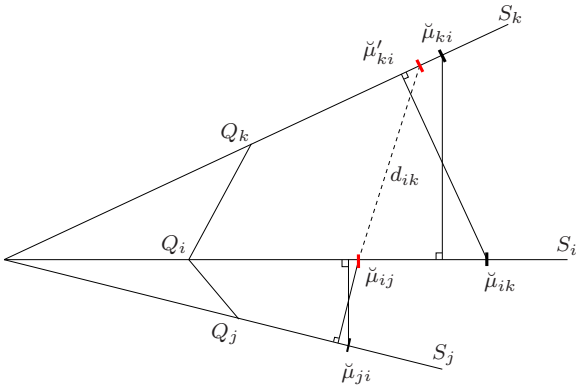


Figure 5: **Refinement of the upper bounds.** The initial bound $\breve{\mu}_{ki}$ gets refined to $\breve{\mu}'_{ki}$.

3.2 Refining the Upper Depth Bounds

The set of initial bounds is not optimal for the whole set of points, as illustrated in figure 5. As an example, we consider three points, and their pairwise computed bounds. The bounds for the points Q_j and Q_k are given by the point Q_i. The points Q_j and Q_k are used to compute two bounds for the point Q_i. Only the most restrictive one is kept *i.e.*, $\breve{\mu}_{ij}$. It means that the depth of the point Q_i cannot be greater than $\breve{\mu}_{ij}$. This gives the new bound $\breve{\mu}'_{ik}$ for the point Q_k.

We propose an iterative implementation of bound refinement. During one iteration, for each point, the upper bounds of the other points induced by the actual point are computed. If they are smaller than their actual bounds, these are updated. The iterations stop when there is no change during one iteration, meaning that the bounds are all coherent. So as to derive the update rule, we refer to equation (1) that links the depth of two points such that the distance between the points is equal to their distance measured in the template *i.e.*, the maximal distance between the two points. We study the upper bound on point j induced by point i. It is given by the largest value of μ_j:

$$\mu_j\left(\mu_i\right) = \mu_i \ \cos(\alpha_{ij}) + \sqrt{d_{ij}^2 - \mu_i^2 \ \sin^2(\alpha_{ij})}. \tag{2}$$

As figure 6 illustrates, this function has a global maximum:

$$\mu_i^{max} = \frac{d_{ij}}{\tan(\alpha_{ij})} \qquad \mu_j(\mu_i^{max}) = \frac{d_{ij}}{\sin(\alpha_{ij})}. \tag{3}$$

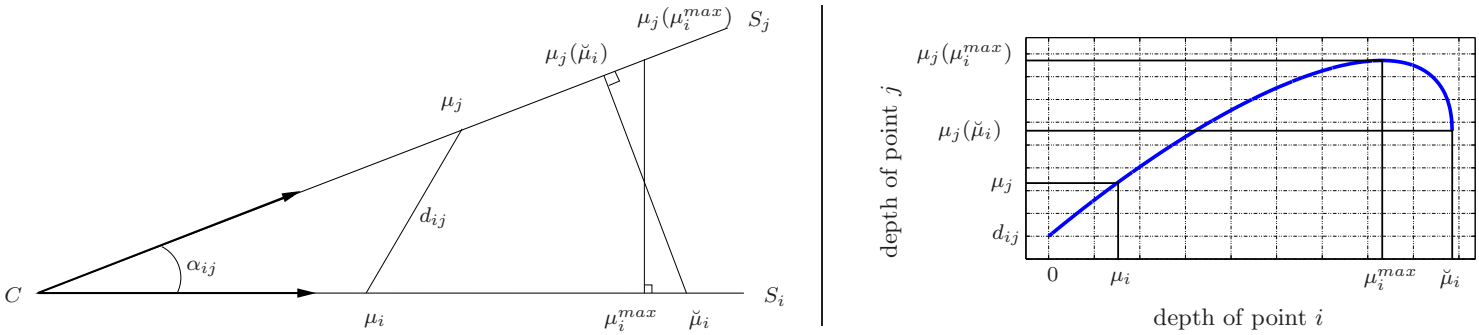


Figure 6: The function (2) giving the depth of point j against the depth of point i. (left) Parametrization of the points, illustrating how μ_i^{max} and $\mu_j(\mu_i^{max})$ are retrieved in equation (3). (right) Graph of the function.

The upper bound for point j with respect to point i is thus:

$$\breve{\mu}_{ji} = \begin{cases} \mu_i \ \cos(\alpha_{ij}) + \sqrt{d_{ij}^2 - \mu_i^2 \ \sin^2(\alpha_{ij})} & \text{if} \quad \breve{\mu}_i \leq \mu_i^{max} = \frac{d_{ij}}{\tan(\alpha_{ij})} \\ \mu_j(\mu_i^{max}) = \frac{d_{ij}}{\sin(\alpha_{ij})} & \text{otherwise,} \end{cases} \tag{4}$$

and the formula to update the bound is the following:

$$\breve{\mu}_j = \min \ (\breve{\mu}_{jj^\star}, \breve{\mu}_{ji}) \, .$$

In our experiments, this process converges in 3 or 4 iterations. It gives the upper bound and the anchor point of each point; both are used to recover a continuous surface. Algorithm 1 summarizes the upper depth bound refinement process.

Algorithm 1 Upper depth bound refinement.

Require: Initial upper depth bounds : $\breve{\mu}_{jj^\star}$.

```
c ← true
while c do
    c ← false
    for j = 1 to number of points do
        for i = 1 to number of points, i ≠ j do
            Compute μ̆_ji using equation (4).
            if μ̆_ji < μ̆_jj* then
                c ← true
                μ̆_jj* ← μ̆_ji
                j* ← i
            end if
        end for
    end for
end while
```

Ensure: Refined upper depth bounds : $\breve{\mu}_j$.

4 Recovering the Surface

Our surface recovery procedure has two main steps:

1. **Reconstruction of sparse 3D points.** (§4.1) The 3D points are computed using the upper bounds and the distances to their anchor points,

2. **Reconstruction of a continuous surface.** (§4.2) The surface is expressed as an interpolation of the recovered 3D points, possibly using surface smoothness priors.

4.1 Finding a Sparse Set of 3D Points

The previously computed set of upper bounds gives the maximal depth of the points. For a fast surface reconstruction algorithm, one can directly use the upper bounds as points on the surface:

$$\tilde{\mu}_i = \breve{\mu}_i. \tag{5}$$

In practice, the error due to this approximation is small, as our experimental error study of §5 shows. However, this is not fully satisfying when regarding surface inextensibility. Indeed, the distance between two upper bound points $\|Q(\breve{\mu}_i) - Q(\breve{\mu}_{i^\star})\|$ can be larger than their distance in the template $d_{ii^\star}$. In other words, we cannot individually assign each point to its upper bound without violating the constraint. For instance, when there is a symmetry between a point and its anchor point (in other words if i and $i^\star$ are mutual anchor points) it can be shown that the distance is equal to $d_{ii^\star} \cos^{-1}\left(\frac{1}{2}\alpha_{ii^\star}\right) \geq d_{ii^\star}$. To get a more consistent surface, we propose an optimization scheme to enforce the length equality between a point and its anchor point. Since the upper bounds give good results, the points' depth such that these length equalities are satisfied are sought near the upper bounds. The optimization can also handle other priors on the points. For instance when processing a video, a first order temporal smoother can be used to penalize the too 'strong' surface variations in time. The optimization problem thus takes the following form:

$$\tilde{\mu} \;=\; \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{n} \Big(\left(\breve{\mu}_i(t) - \mu_i(t)\right)^2 + \gamma \left(\mu_i(t) - \mu_i(t-1)\right)^2 + \eta(\|Q_i - Q_{i^\star}\| - d_{ii^\star})^2 \Big), \tag{6}$$

with μ the points' depth vector and $\mu_i(t)$ the depth of the $i-$th point for frame t (the current frame). The choice of the balancing weights γ and η are discussed in §6. This is a nonlinear least squares problem that we solve with the Levenberg-Marquardt algorithm (Hartley and Zisserman, 2004) (the initial solution is given by equation (5)). We use the MATLAB implementation provided by the `lsqnonlin` function. The Cheirality constraints imply that the depths in vector μ must be positive, and this is easily incorporated in the minimization.

4.2 Interpolating to a Continuous Surface

The reconstructed 3D points are eventually treated as control points of a mapping Γ from the template to the 3D space. This allows us to represent the surface by mapping a regular mesh from the template. In practice the mapping

we choose is composed of three 2D to 1D Thin-Plate Splines (Bookstein, 1989):

$$\Gamma(q) = Aq + \sum_{i=1}^{n} \rho(\|q - q_i\|)Q_k \quad \text{with} \quad \rho(d) = d^2 \log(d),$$

where A is a (3×2) matrix that represents the affine part of the Thin-Plate Splines. These have proven efficient in the representation of deformable objects. Getting a continuous surface makes it possible to deal with surface priors, such as surface smoothness. At this stage, another optimization process can be used to include these priors. They are written as penalty terms of a cost function that is minimized with respect to the depth of the control points. For priors on the temporal and geometric smoothness of the surface (modeled by a penalty on the squared second spatial derivatives of the surface, also called the bending energy), one can write this optimization problem as:

$$\tilde{\mu} \;=\; \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{n} \left((\breve{\mu}_i(t) - \mu_i(t))^2 + \eta(\|Q_i - Q_{i^\star}\| - d_{ii^\star})^2 \right) + \lambda \sum_{j=1}^{m} \left\| \frac{\partial^2 \Gamma}{\partial q^2}(c_j) \right\|^2 + \gamma \sum_{j=1}^{m} \|C_j(t) - C_j(t-1)\|^2, \tag{7}$$

with C_j a vertex of a surface mesh (C_j depends on the unknown depths since $C_j = \Gamma(c_j)$, where c_j is a point defining the mesh position in the template), m the number of vertices of the mesh and λ, γ and η balancing weights controlling the trade-off between the distance to the bounds, the geometric and temporal smoothness terms and the inextensibility constraints (the influence of the smoothing weight λ on the reconstructed surface will be experimentally tested in §5.3). Fixing the deformation centers of the Thin-Plate Splines in the template, problem (7) shows to be nonlinear least squares. It can be solved similarly as problem (6). The spatial smoothing term is evaluated in closed-form, as described in (Bartoli et al., 2010).

5 Error Analysis

The quality of the reconstruction depends on the number of correspondences and the noise in the images. Though the latter has been ignored in the theoretical derivation, we show how to deal with it in the reconstruction algorithm. The experiments to assess the reconstruction error against the number of points or the noise magnitude are performed on synthetic surfaces. They are modeled by developable surfaces, which are isometric to the plane. In practice we use a 200 mm wide square shape, that we randomly deform using the generative developable surface model we proposed (Perriollat and Bartoli, 2007). The feature points are randomly drawn on the shape. Examples of simulated shapes are shown in figure 7. The 3D reconstruction error $e(i)$ for the $i-$th feature point is monitored and averaged over all points. It is defined as the distance of the reconstructed to the true 3D point:

$$e(i) = \|\tilde{Q}_i - \hat{Q}_i\|. \tag{8}$$

The dashed curves represent the fast implementation error (equation (5)) and the bold curves correspond to the optimized points under length penalty (equation (6)). We also plot bars showing the standard deviation of the 3D error. As will be seen, the standard deviation is important, showing that the accuracy of the algorithms significantly depends on the surface that is being considered. In general, the point optimization has a lower 3D error and a slightly larger standard deviation than the fast implementation.

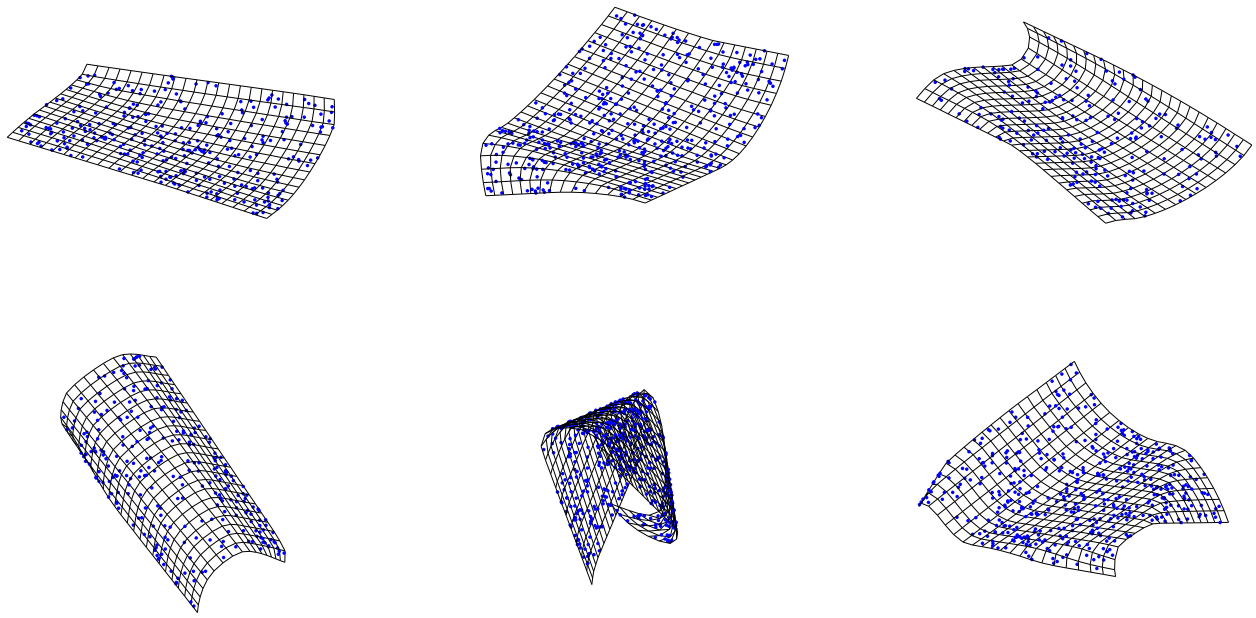


Figure 7: **Simulated data.** Example of simulated developable surfaces and points that we use in our experiments.

5.1 Number of Points

Figure 8 shows the average reconstruction error against the number of correspondences. As expected, the error decreases thanks to the length constraints. For both algorithms the higher the number of points the lower the error.

The accuracy of the reconstruction is related to two quantities: the amount of deformation between the points and the 'orientation' of the points with respect to the camera. Their respective influence will be explained below. While deforming, the Euclidean distance between the 3D points decreases. Since our algorithm is based on the preservation of the Euclidean distance between a point and its anchor point, the less it deforms between these point pairs, the better the results. The 3D orientation of a point and its anchor point changes the relative position of their projections in the image. There exist a configuration for which the angle between the sightlines of the two points is maximum. This is the optimal orientation since it leads to a tighter upper bound, and thus minimizes the reconstruction error. For both situations, the increasing number of points gives more chance to get an optimal situation *i.e.*, for which the points and their anchor points are well-oriented and the surface is not deformed too much between them.

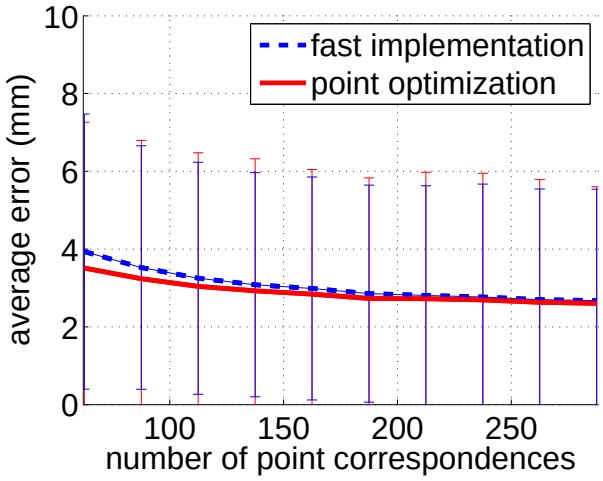


Figure 8: **Simulated data.** Error against the number of point correspondences.

5.2 Noise on the Point Positions

The point correspondences we use between the template and the image have positions corrupted by noise. Since we do not use special points, there are two ways for one to look at the noise. Indeed, if one considers a single pair of corresponding points, and fixes one of these two points, say the image point, there always exist an 'ideal' template point that exactly satisfies the constraints we use. Consequently, we may interpret a measured image point as being noise-free, and the measured template point as a noisy version of the ideal corresponding point. The opposite is obviously also true. Therefore, one can arbitrarily choose which observations (in the template or in the image) are exact and which ones are noisy. This choice induces differences in our algorithm: 'noise in the image' changes the orientation of the sightlines whereas 'noise in the template' modifies the reference distances d_{ij} between the points. Since our 3D points are parametrized along their sightlines, we choose the second possibility, that yields a simple yet efficient solution to understand and handle noise. The noisy distances measured in the template lead to tighter upper bounds if they are under-estimated. With the refinement process on the bounds, this error is propagated to other points, spoiling the reconstruction accuracy. To avoid this, we add a constant corrective term k to the reference distances:

$$d_{ij} \longleftarrow d_{ij} + k. \tag{9}$$

This term reflects how reliable the distances are. Its efficiency is related to the noise level, as shown in figure 9. The curve presents a minimum at 55% of the average noise magnitude, giving an empirical way to choose the value of the term k. This curve shows also that it is better to over-estimate this parameter than to under-estimate it. However it is difficult in practice to evaluate the noise magnitude. This term is fixed to one pixel in our experiments. The precision of the reconstruction gracefully degrades with the noise magnitude, as shown in figure 10. The relation between the noise magnitude and the reconstruction error is nearly linear. For a noise magnitude of 5 pixels, the average error is below 5.5 mm.

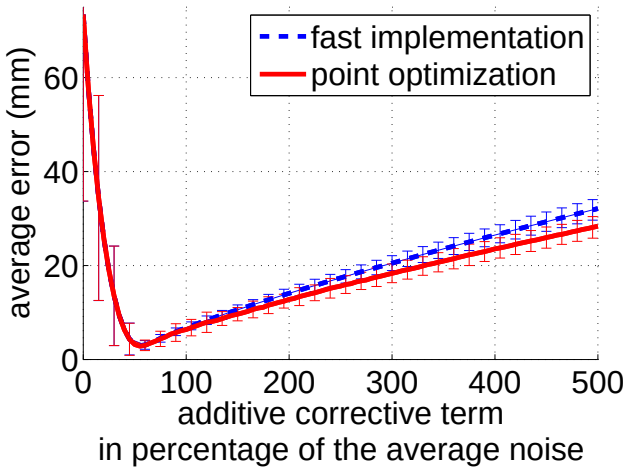


Figure 9: **Simulated data.** Influence of the corrective term on the 3D error.

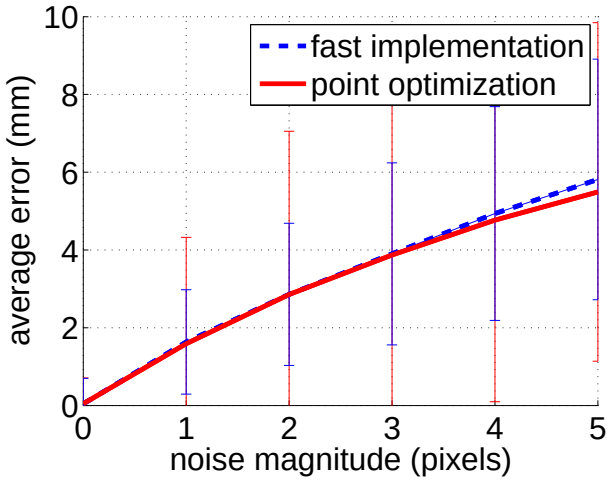


Figure 10: **Simulated data.** 3D error versus the noise magnitude in the image.

5.3 Smoothing Weight

Our reconstruction engine uses a smoothing prior, as equation (7) shows. To this prior, a smoothing weight λ is associated, that controls the strength of the surface smoothness. This prior is likely to have a higher impact on the recovered surface when the number of available point correspondences is fewer. We thus looked at how the reconstructed surface error changes as a function of both the number of point correspondences and the value of λ. Figure 11 shows the results we obtained. What this reveals is that, the influence of the smoothing weight on the quality of the reconstructed surface is marginal. It is indeed slightly more important when the number of points is low but not significant when the number of points is above about 30. We also observe that the quality of the reconstructed surface gracefully increases with the number of points, which is consistent with our previous experimental results. It is thus in general not necessary to automatically tune the smoothing weight for our algorithm using a complexity selection criterion such as the one used in (Bartoli, 2008).

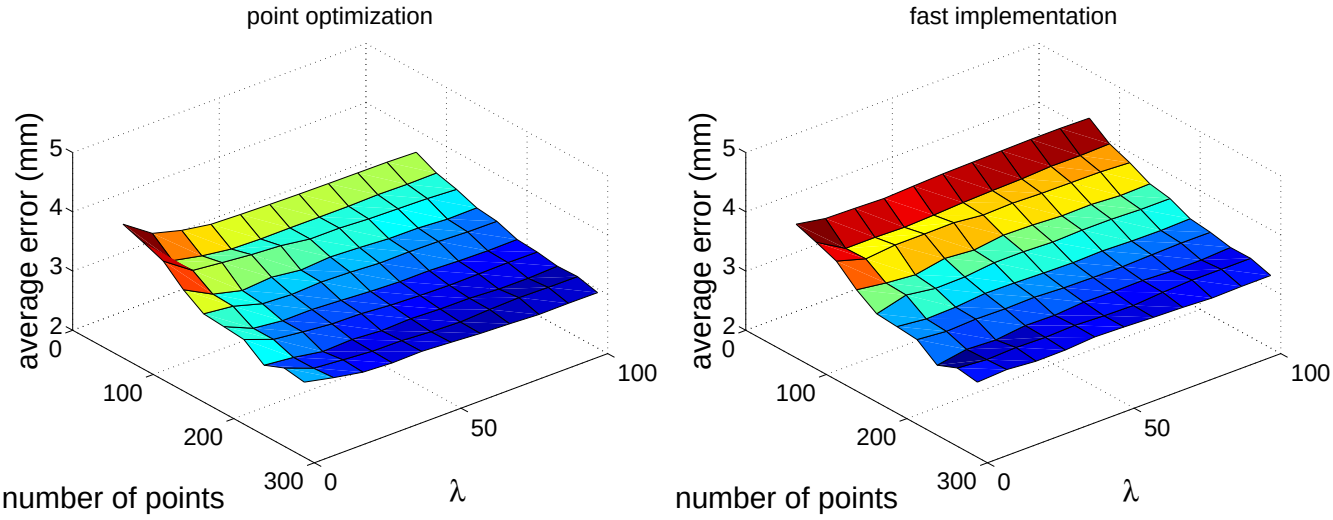


Figure 11: **Simulated data.** 3D error versus the smoothing weight λ and the number of point correspondences.

6 Experimental Results on Real Data

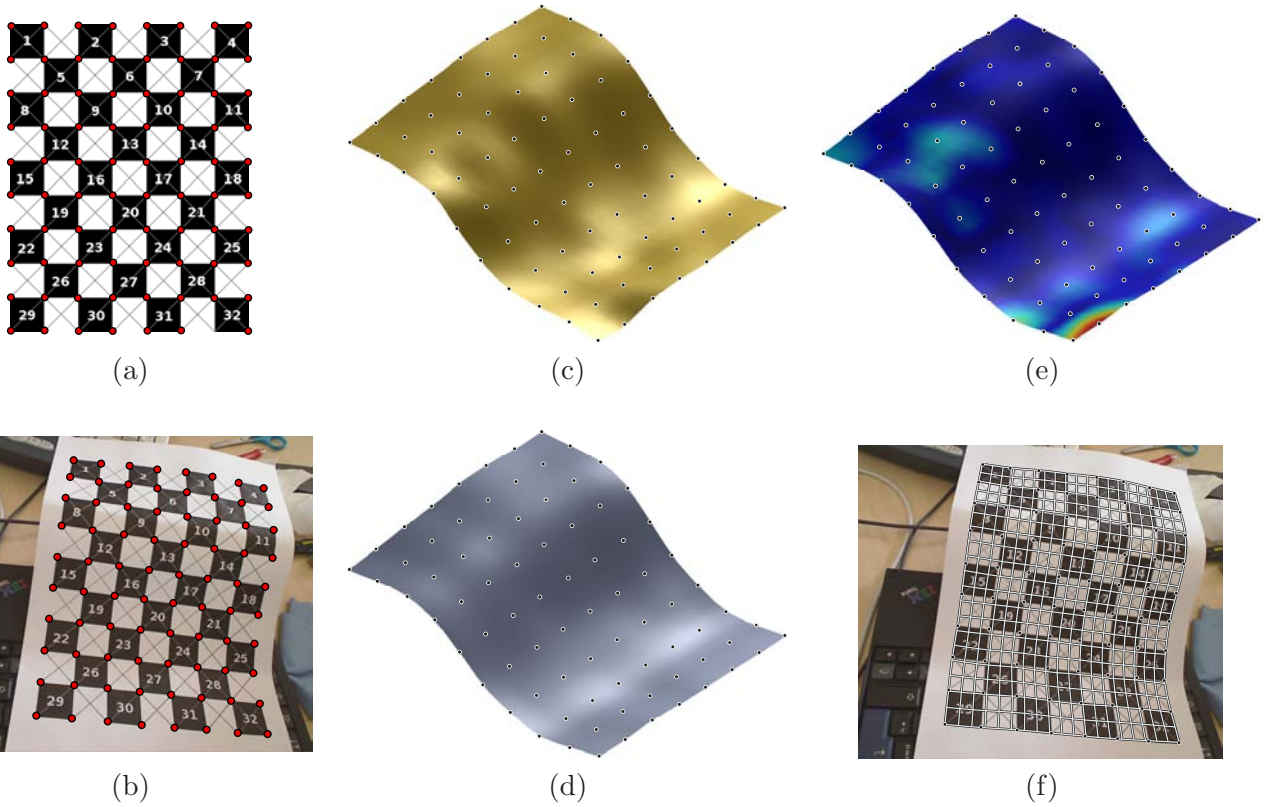
This section reports the experimental results we obtained by applying our algorithms to real datasets. We first show results obtained for single-image datasets, and then results for videos. In the latter case, the temporal consistency of the time-varying 3D structure is enforced.

6.1 Reconstruction from Single Images

The results we show in this section were obtained from a single image of the deformed object of interest. The reference models, *i.e.*, the templates, are shown in figure 2. So as to evaluate the quality of the 3D reconstructions, we compare them to 3D measurements obtained from two images of the deformed object, in a stereovision manner. The surfaces we reconstruct with our algorithm are first registered with a scaled euclidean transformation to the

stereo reconstructions, for which the scale is properly normalized so as to obtain metric measurements that we report in mm. We then compute the surface discrepancy, as a measure of the quality that our algorithm reaches. We note that since the points are reconstructed by our algorithm along their sightlines, the reprojection error they induce always vanishes. The stereo reconstruction is obtained as follows. We use a stereo rig, that we accurately calibrate (internally and externally) using (Bouguet, 2008). Points are then triangulated in a maximum likelihood manner by minimizing their reprojection error as in (Hartley and Sturm, 1997).

Reconstruction of a slightly deformed paper sheet. The data and results we obtained are shown in figure 12. We use 80 point correspondences, both for the monocular and stereo algorithms. The paper shape is well reconstructed by our algorithm. It closely resembles the shape obtained with stereovision. The inter-surface distance is 1.2 mm. It means that our 3D reconstruction is very close to the stereo, reference reconstruction. The average stereo reprojection error was 0.83 pixels.



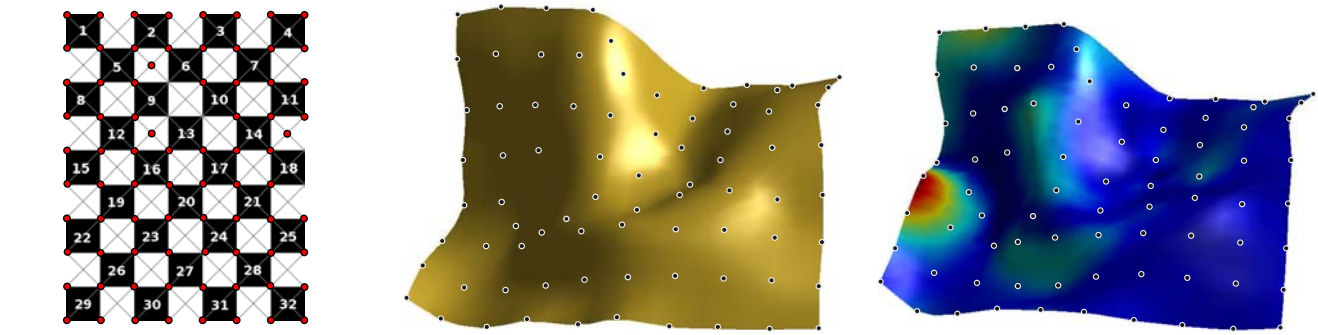
(a) (c) (e)

(b) (d) (f)

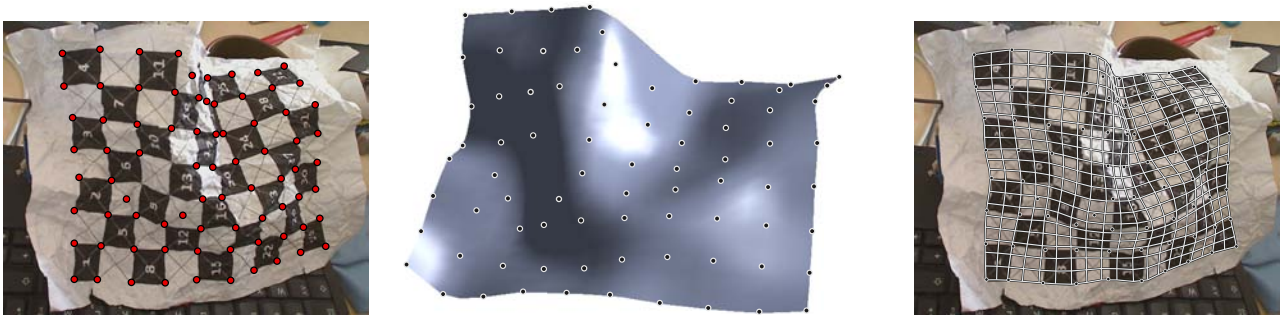
Figure 12: **Reconstruction of a slightly deformed paper sheet.** (a) Point correspondences on the template. (b) Point correspondences on the image of the deformed paper sheet. (c) Reconstruction obtained with our monocular algorithm. (d) Reconstruction obtained by stereovision (the second image of the deformed paper sheet is not shown). (e) Color-coded surface discrepancy between the monocular and stereo reconstructions. (f) The reconstructed surface reprojected in the image.

Reconstruction of a creased paper sheet. The data and results we obtained are shown in figure 13. We use 78 point correspondences, both for the monocular and stereo algorithms. The paper shape is similar for the two algorithms. The surface discrepancy is 3.3 mm. It is larger than for the slightly deformed paper example, but is still

of an acceptable magnitude. This slight degradation of accuracy is due to the creases that make the deformations more difficult to estimate. The average stereo reprojection error was 0.99 pixels.



(a) (c) (e)



(b) (d) (f)

Figure 13: **Reconstruction of a creased paper sheet.** (a) Point correspondences on the template. (b) Point correspondences on the image of the creased paper sheet. (c) Reconstruction obtained with our monocular algorithm. (d) Reconstruction obtained by stereovision (the second image of the deformed paper sheet is not shown). (e) Color-coded surface discrepancy between the monocular and stereo reconstructions. (f) The reconstructed surface reprojected in the image.

Reconstruction of a deformed can. The data and results we obtained are shown in figure 14. The reference, template model is in this case an open cylinder, and is thus different from the two previous datasets, for which the reference model is a plane. The global surface shape is correctly estimated by our algorithm. For 72 point correspondences, we get a surface error between our monocular and the stereo algorithms of 1.6 mm. The average stereo reprojection error was 0.57 pixels.

6.2 Reconstruction from Videos

Videos could be simply handled by applying our algorithm to each frame independently. However, it is possible to enforce temporal consistency of the frame-varying (*i.e.*, time-varying) reconstructed surface. This is done by introducing a term penalizing abrupt temporal surface changes in the cost function, as is done in equations (6) and (7).

Reconstruction of a bending paper sheet. Figure 15 shows some frames (over a total of 208 frames) of a paper sheet filmed while being manually bent. The figure also shows a warp visualization grid. Indeed, for this example,

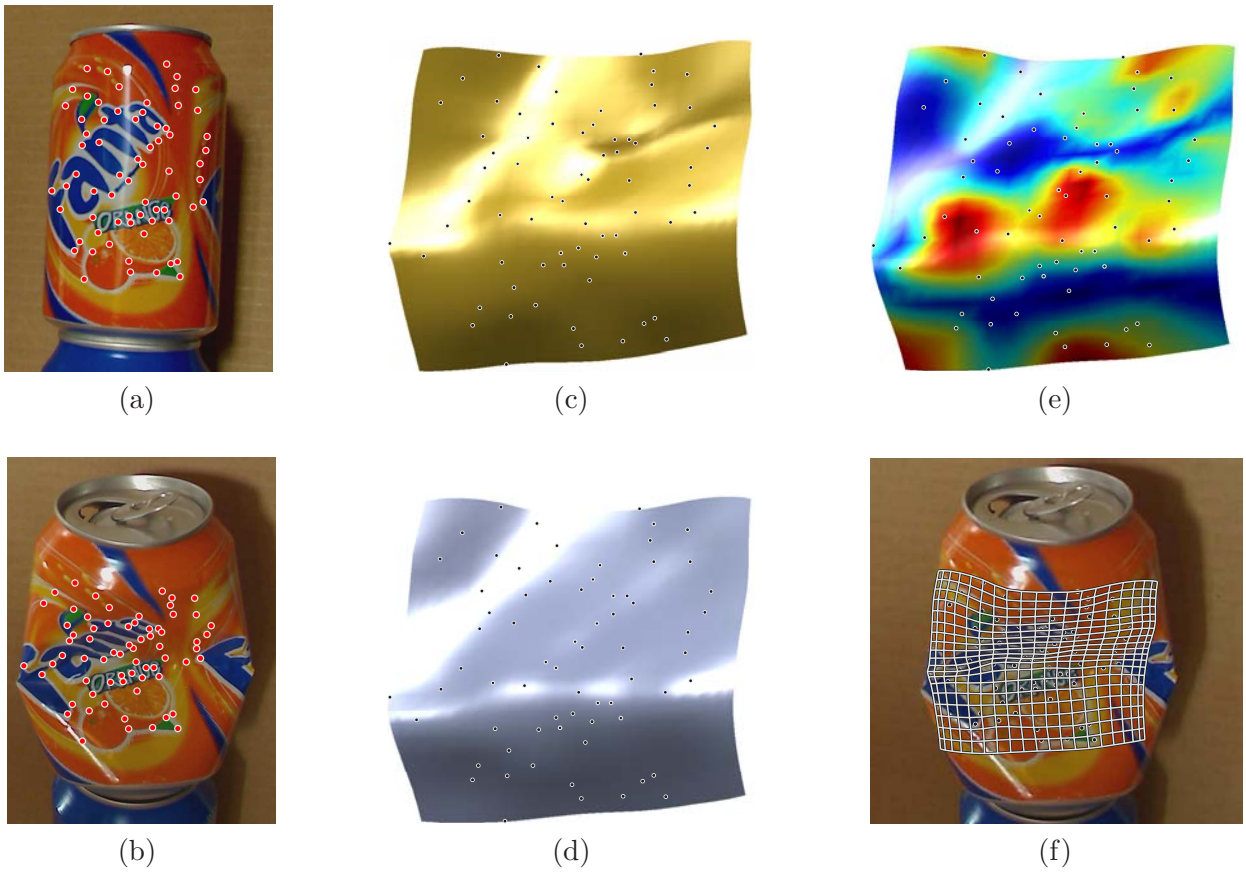


Figure 14: **Reconstruction of a deformed can.** (a) Point correspondences on the template. (b) Point correspondences on the image of the deformed can. (c) Reconstruction obtained with our monocular algorithm. (d) Reconstruction obtained by stereovision (the second image of the deformed paper sheet is not shown). (e) Color-coded surface discrepancy between the monocular and stereo reconstructions. (f) The reconstructed surface reprojected in the image.

we used the image registration method of (Gay-Bellile et al., 2009). This provided us with a set of image warps, mapping points from the template to each of the video frames. We then drew a set of 140 points on a regular grid in the template, that we mapped to each of the video frames to serve as a dataset for our algorithm.[3]

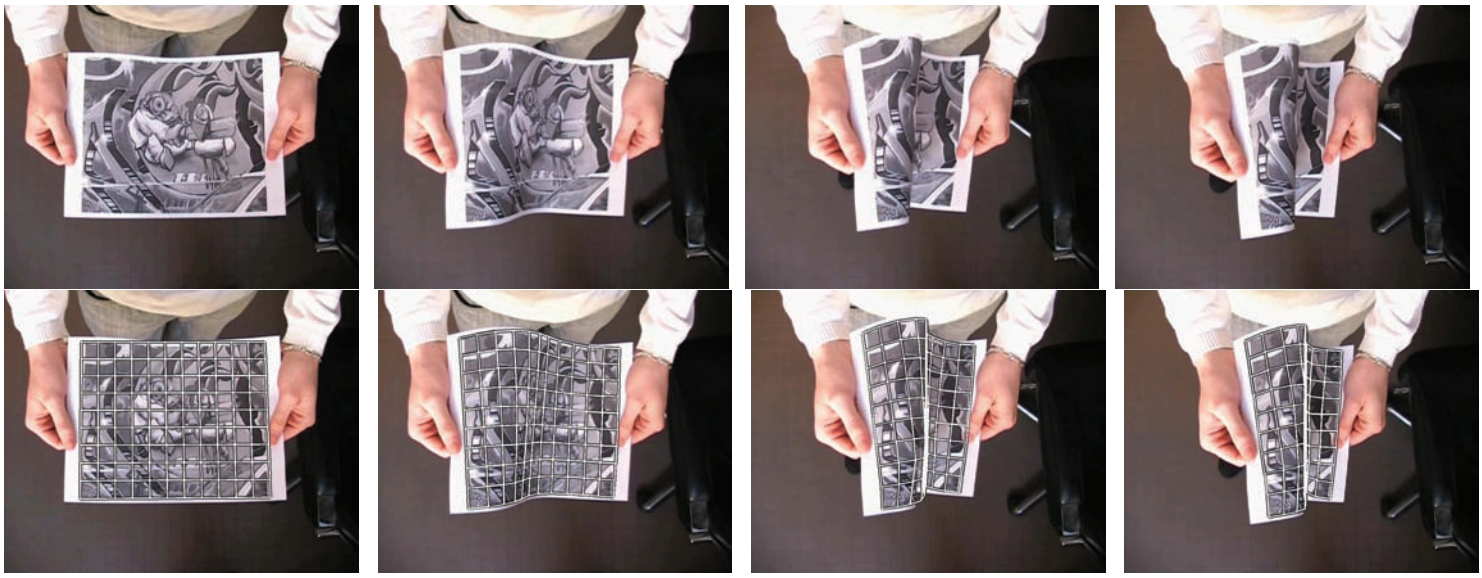


Figure 15: **Reconstruction of a bending paper sheet.** (top) Four frames from the 208-frames video. (bottom) The warp visualization grid from which point correspondences were extracted (see main text for details).

Once the 3D surface is reconstructed, it can be used for augmented reality purposes. As figure 16 shows, we were able to augment the video by inserting new, virtual objects. Those were manually set on the template, and automatically rendered in the original video frames, giving a highly convincing visual impression. As figure 16 also shows, we can render the reconstructed surface from a viewpoint different from the original one, and with an arbitrary appearance for the surface. This illustrates that our algorithm opens the possibility for full monocular video surface deformation capture.

The results we obtained are very satisfying. Indeed, the optic flow field induced by the paper sheet between the template and some of the video frames significantly collapses on the self-occlusion boundary. This makes 3D reconstruction difficult, since all the points located in the self-occluded area are non-informative on the surface shape, which still, is pretty well recovered by our algorithm, even in these areas where it is hidden.

Reconstruction of a waved tee-shirt. Some frames of a 898-frames video of a tee-shirt being waved are shown in figure 17. In this video, 43 point tracks were on average obtained using the Kanade-Lucas-Tomasi (KLT) tracker (Shi and Tomasi, 1994), from which we manually removed the erroneous tracks. Loss of point tracks and newly detected points caused the number of visible points to vary between 26 and 71 over the video.[4] Figure 18 shows a view of the reconstructed camera and surface. While shown on the figure, the camera is not explicitly reconstructed. Indeed, a full 3D surface is computed by our algorithm for each frame, with the camera is at a fixed position. What we did

[3]For that video, we used typical values for the balancing weights, $\lambda = 500$, $\gamma = 0.5$ and $\eta = 1.5$, in equation (7).
[4]For that video, we used typical values for the balancing weights, $\lambda = 300$, $\gamma = 0.25$ and $\eta = 1.5$, in equation (7).

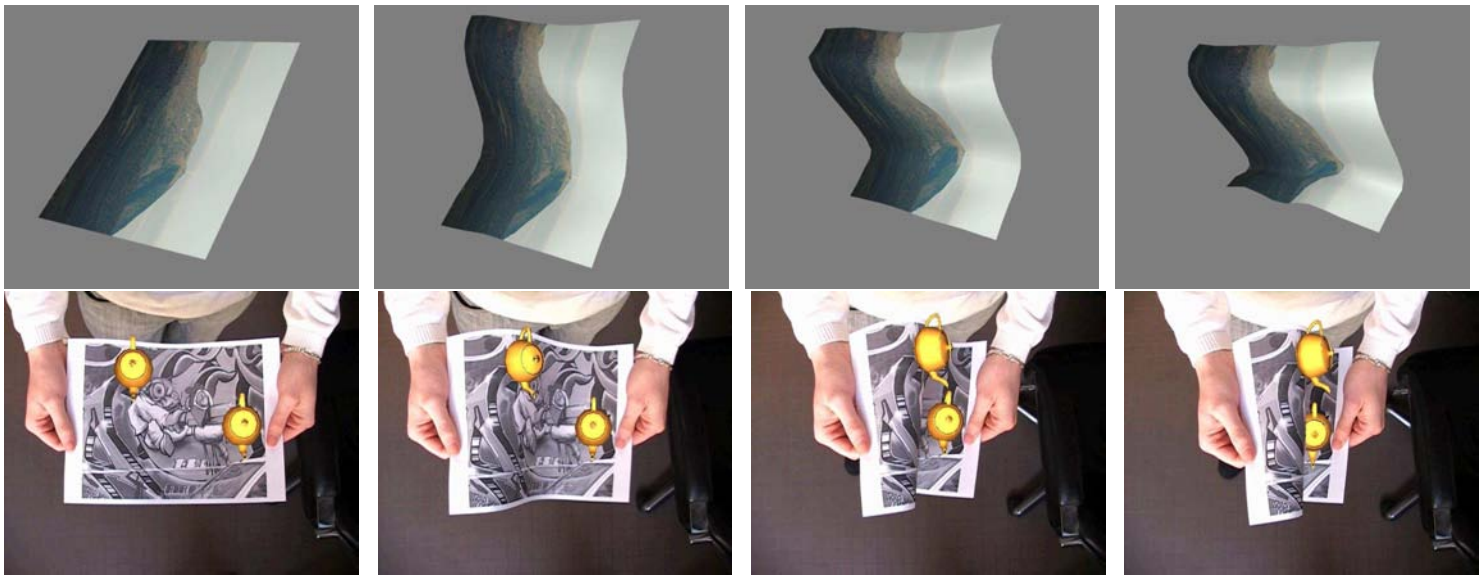


Figure 16: **Augmentation of a bending paper sheet.** (top) The reconstructed surface rendered from a different camera viewpoint as the original one and retextured. (bottom) The video augmented with virtual objects. Knowing the 3D surface allows us to correctly place the teapot and to take the surface self-occlusions into account. All but the teapot placement on the template by the user is automatically done.

to infer the camera pose with respect to the surface, was to standardize the surface's centre of gravity and principal axes, and interpret the rigid standardizing transformation as the camera pose.

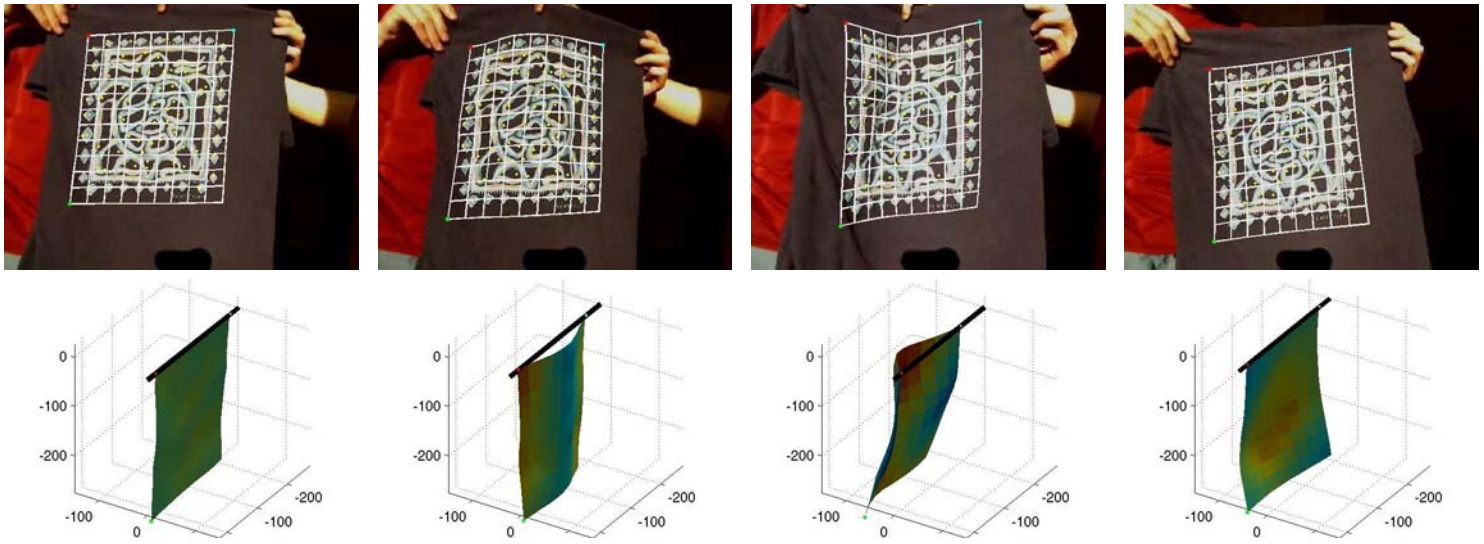


Figure 17: **Reconstruction of a waved tee-shirt.** (top) Four frames from the 898-frames video, overlaid with the tracked points and a grid reprojected from the reconstructed surface. (bottom) The reconstructed surface rendered from a new viewpoint.

7 Conclusions

The algorithm we presented has been designed for the reconstruction of inextensible surfaces imaged by a perspective camera. It evaluates the 3D bounds on the points such that the inextensible constraints can be satisfied. A surface optimization can then be run to handle priors such as surface smoothness or temporal consistency. Our results

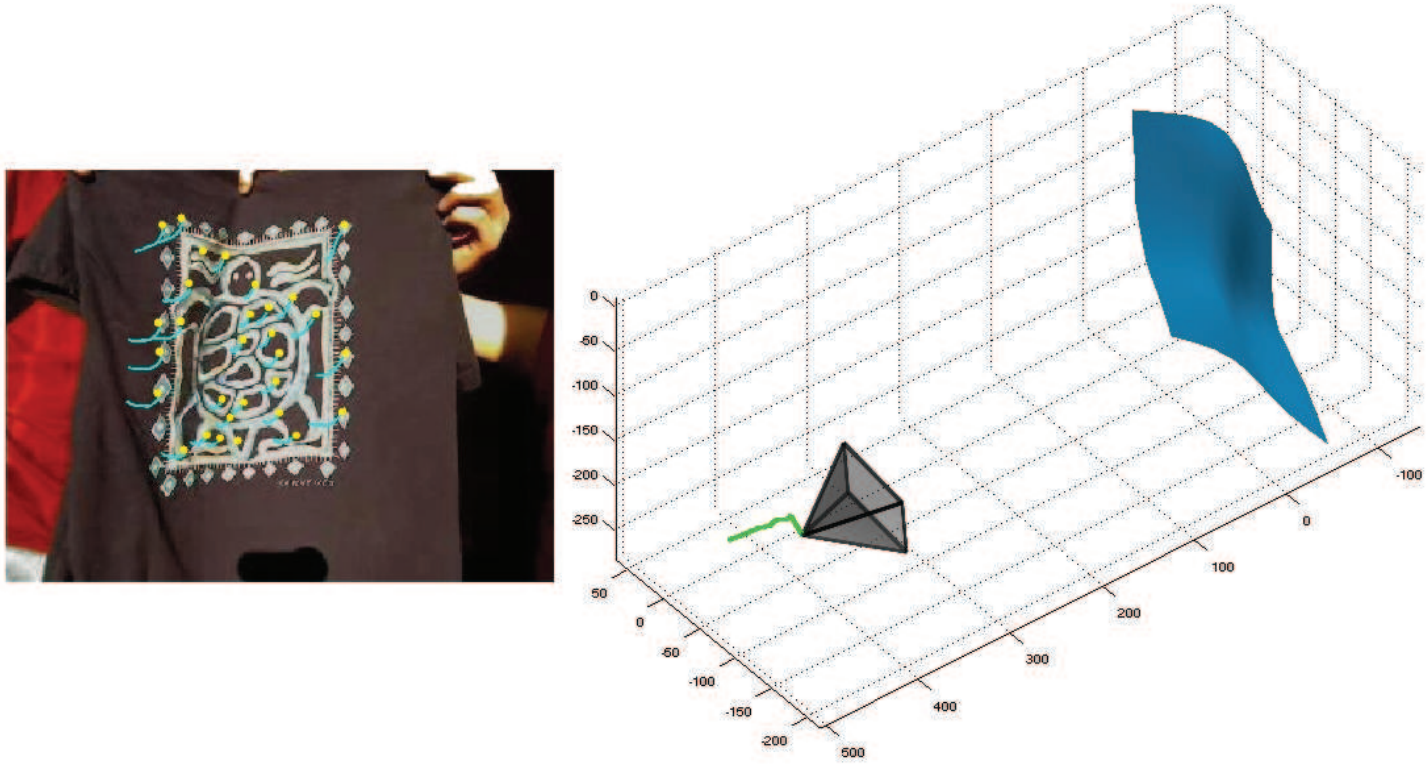


Figure 18: **Reconstruction of a waved tee-shirt.** (left) One of the video frames, overlaid with the point tracks and their trajectories over the past few frames. (right) The reconstructed surface and camera (shown as a pyramid), with its trajectory over the past few frames.

are convincing, and show that our algorithm brings a simple and effective solution to the monocular deformable reconstruction problem.

There are several possible extensions of our work. One of them regards the continuous surface reconstruction, obtained by interpolating the reconstructed points. The surface we compute does not strictly speaking satisfy the inextensibility constraints, because they are used as a weighted penalty in our cost function. However, they could be used as hard inequality constraints, since the euclidean distance between two points can decrease but not increase from the template to the image. Alternatively, one could use the link that exist between our corrective term and the preservation of the euclidean distance or detect and inhibitate those constraints that get the most violated (for instance over a paper crease). This may improve the accuracy of the final result. Finally, we assume that there is no point mismatches between the template and the input image. This is a real practical problem that should be addressed in future work.

Acknowledgements

We are indebted to Vincent Gay-Bellile for the registration results he provided on the bending paper video.

References

A. Bartoli. Maximizing the predictivity of smooth deformable image warps through cross-validation. *Journal of Mathematical Imaging and Vision*, 31(2-3):133–145, July 2008.

A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *International Conference on Computer Vision and Pattern Recognition*, 2008.

A. Bartoli, M. Perriollat, and S. Chambon. Generalized thin-plate spline warps. *International Journal of Computer Vision*, 88(1):85–110, May 2010.

F. L. Bookstein. Principal warps: Thin-Plate Splines and the decomposition of deformations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

J.-Y. Bouguet. Camera calibration toolbox for matlab. www.vision.caltech.edu/bouguetj, 2008.

M. Brand. A direct method for 3D factorization of nonrigid motion observed in 2D. In *International Conference on Computer Vision and Pattern Recognition*, 2005.

C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *International Conference on Computer Vision and Pattern Recognition*, 2000.

A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *International Journal of Computer Vision*, 64(1):5–30, 2005.

A. Del Bue. A factorization approach to structure from motion with shape priors. In *International Conference on Computer Vision and Pattern Recognition*, 2008.

A. Ecker, K. Kutulakos, and A. Jepson. Semidefinite programming heuristics for surface reconstruction ambiguities. In *European Conference on Computer Vision*, 2008.

R. Ferreira, J. Xavier, and J. Costeira. Reconstruction of isometrically deformable flat surfaces in 3D from multiple camera images. In *International Conference on Acoustics, Speech, and Signal Processing*, 2009.

V. Gay-Bellile, M. Perriollat, A. Bartoli, and P. Sayd. Image registration by combining thin-plate splines with a 3D morphable model. In *International Conference on Image Processing*, 2006.

V. Gay-Bellile, A. Bartoli, and P. Sayd. Direct estimation of non-rigid registrations with image-based self-occlusion reasoning. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2009. To appear.

N. A. Gumerov, A. Zandifar, R. Duraiswami, and L. S. Davis. Structure of applicable surfaces from single views. In *European Conference on Computer Vision*, 2004.

R. Hartley. Cheirality. *International Journal of Computer Vision*, 26(1):41–61, 1998.

R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

J. Liang, D. DeMenthon, and D. Doermann. Geometric rectification of camera-captured document images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30(4):591–605, 2006.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

S. Olsen and A. Bartoli. Implicit non-rigid structure-from-motion with priors. *Journal of Mathematical Imaging and Vision*, 31(2-3):233–244, July 2008.

M. A. Penna. Non-rigid motion analysis: isometric motion. *CVGIP: Image Understanding*, 56:366–380, 1992.

M. Perriollat and A. Bartoli. A quasi-minimal model for paper-like surfaces. In *Workshop "Towards Benmarking Automated Calibration, Orientation and Surface Reconstruction from Images"*, 2007.

M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. In *British Machine Vision Conference*, 2008.

J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, February 2008.

M. Prasad, A. Zisserman, and A. Fitzgibbon. Single view reconstruction of curved surfaces. In *International Conference on Computer Vision and Pattern Recognition*, 2006.

M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface deformation models for nonrigid 3D shape recovery. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1–7, August 2007.

M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3D surface registration. In *European Conference on Computer Vision*, 2008a.

M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3D shape recovery. In *International Conference on Computer Vision and Pattern Recognition*, 2008b.

S. Shen, W. Shi, and Y. Liu. Monocular template-based tracking of inextensible deformable surfaces under l_2-norm. In *Asian Conference on Computer Vision*, 2009.

J. Shi and C. Tomasi. Good features to track. In *International Conference on Computer Vision and Pattern Recognition*, 1994.

L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, May 2008.

R. Vidal and D. Abretske. Nonrigid shape and motion from multiple perspective views. In *European Conference on Computer Vision*, 2006.

J. Xiao and T. Kanade. A linear closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 67(2):233–246, March 2006.