Monocular Template-based Reconstruction of Inextensible Surfaces

Mathieu Perriollat[1] Richard Hartley[2] Adrien Bartoli[1]
[1] LASMEA, CNRS / UBP, Clermont-Ferrand, France
[2] RSISE, ANU, Canberra, Australia
Mathieu.Perriollat@gmail.com

Abstract

We present a monocular *3D* reconstruction algorithm for inextensible deformable surfaces. It is based on point correspondences between the actual image and a template. Since the surface is inextensible, its deformations are isometric to the template. We exploit the underlying distance constraints to recover the *3D* shape. Though these constraints have already been investigated in the literature, we propose a new way to handle them. As opposed to previous methods, ours does not require a known initial deformation. Spatial and temporal smoothness priors are easily incorporated. 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 faced 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 is a challenging problem, illustrated in figure 1 (a). It has been addressed for years and several algorithms have been proposed. The *3D* shape seen in the template is usually known. The general problem is ill-posed due to depth ambiguities [11]. Additional consistency constraints are thus required. Most commonly, *ad hoc* constraints are used. These include spatial and temporal surface smoothness [3, 4, 10] and the low-rank shape model [2, 3].

The algorithm we propose is dedicated to inextensible surfaces such as those shown in figure 1 (b,c). It uses point correspondences to compute upper bounds on the points' depth thanks to the inextensibility assumption. We show that these bounds directly provide a decent *3D* reconstruction of the surface. The algorithm does not require an initial guess and easily handles additional smoothness constraints.

There are two main differences between our method and the previous ones:

- the inextensibility constraints are treated as hard constraints instead of as a penalty. It makes the result less empirical because we guarantee to find an inextensible surface. Indeed, smoothing terms that can be used to handle other priors on the surface do not alter the inextensible property of the solution. Other methods usually mix different penalties and so generally have to carefully trade off various terms to get convincing results.

- our algorithm does not need any assumption about the surface deformation in the video, contrarily to other methods such as [10, 13] for which the first frame of the video must be 'similar' to the template. It is simple and fast, and can therefore be used to provide a good initialization to more sophisticated algorithms.

The 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 on the reconstruction error is proposed in §5. Results on synthetic and real data sets are reported in §6. Eventually, we give our conclusion and research perspectives in §7.

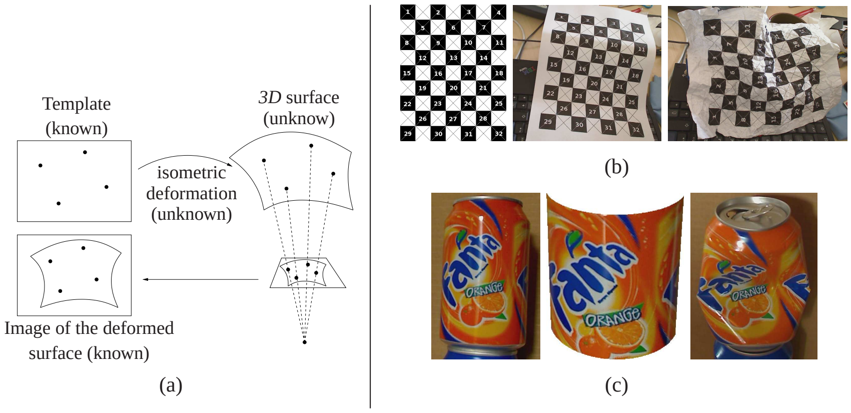

Figure 1: Monocular reconstruction of a deformable surface. (a) Problem setup. (b) Examples of paper sheets: the template (left) and two deformed sheets, a smooth one (middle) and a creased one (right). (c) Example of a can: template image (left), *3D* template (middle), and the input image of the deformed can (right).

2 State of the Art

There are three main components in monocular deformable scene reconstruction: the general low-rank shape model, the assumption that the object of interest is a surface and the knowledge of a template. They can be independently used or combined together so as to handle the monocular reconstruction ambiguities.

The low rank factorization solution to the non rigid shape recovery problem has been introduced by [2]. The object is represented by a combination of unknown basis shapes. The algorithm recovers both the basis shapes and the configuration weights for each frame of the video. The surface hypothesis has recently been incorporated through shape priors [3]. This method needs the whole video to compute the solution and thus is not suited for reconstruction on the fly.

Learning approaches have proven efficient to model deformable objects [12, 13]. The main drawback is the lack of generality when the trained model is too specific. To deal with videos, temporal consistency is used to smooth the deformations [10]. It requires to know the initial *3D* shape. It usually needs a template, and the video is such that the object deformation in the first frame is close to the one in the template.

Methods using only the surface assumption have been proposed. They require strong priors on the surface. One of the motivations for these methods is to perform paper scanning from images of deformed paper sheets. For this kind of applications, a template is obviously not available. Under the surface smoothness assumption, [4] solves a system of differential equations on the page borders to obtain the *3D* shape. Other approaches such as [7] 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 surfaces and uses a template. It assumes the internal parameters of the camera to be known. It is more flexible than other approaches since it applies to any inextensible surface such as paper, garment or faces. Only one frame is needed to compute the reconstruction and there is no need for a reference image in the video, *i.e.* an image for which the *3D* surface is known in advance.

3 Finding Upper Bounds on the Surface Depth

3.1 Principle

We focus on inextensible deformable objects imaged by projective cameras. A surface template is assumed to be known. The template is composed of the *3D* surface shape registered with an image of the object. Examples are shown on figure 1. For the paper sheet and the piece of garment, the reference shape is a plane, and for the can, it is a 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 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 along the sightlines increases with their depths. The template gives us the maximal distance between two points (when the real dimensions of the template are available, the scale ambiguity can be 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 SIFT [8] or a detection process designed for deformable objects [9]. We assume that there is no mismatch. The evaluation of the bounds is then done through a two step algorithm:

- **Initialization.** (§3.2) A suboptimal solution is computed by using pairwise constraints.
- **Refinement.** (§3.3) An iterative refinement process considers the upper bounds as a whole and tunes all of them to get a fully compatible set of bounds.

Our notations are given in table 1.

3.2 Initializing the Bounds

The initialization of the bounds is computed pairwise. Two points and the inextensibility constraint are sufficient to bound the position of these two points along their sightlines.

Notation	Meaning
$\mathbf{T}$	template
$q_i^{\mathbf{T}}$	point *i* in the template
$\mathbf{I}$	image of the deformed object
P	camera matrix for **I**
C	camera centre for **I**
$q_i^{\mathbf{I}}$	point *i* in the image
S_i	sightline for point $q_i^{\mathbf{I}}$
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 coordinate
$\|\|.\|\|$	vector two norm

$d_{ij} = \|\|q_i^{\mathbf{T}} - q_j^{\mathbf{T}}\|\|$	Euclidean distance between $q_i^{\mathbf{T}}$ and $q_j^{\mathbf{T}}$
μ_i	depth of the point *i*
$Q_i = Q_i(\mu_i)$	*3D* point *i*
$\hat{\mu}_i$	true depth of the point *i*
$\hat{Q}_i$	true *3D* point *i*
$\tilde{\mu}_i$	reconstructed depth of the point *i*
$\tilde{Q}_i$	reconstructed *3D* point *i*
$i^\star$	indice of the point constraining the depth of point *i*
$\breve{\mu}_i = \breve{\mu}_{ii^\star}$	maximal depth of the point *i*
$\breve{Q}_i$	deeper *3D* point *i*

Table 1: Our notations for this paper.

For n correspondences, it gives $n-1$ bounds for each point. Only the most restrictive bound (*i.e.* the tightest one) is kept as the initial bound.

The sightlines are computed in the image of the deformed object **I**, (details can be found in *e.g.* [6]). The camera matrix $P = [M|\mathbf{p}_4]$ is composed of a (3×3) matrix M and a (3×1) vector $\mathbf{p}_4$. The camera center is $C = -M^{-1}\mathbf{p}_4$. The vector v_i orienting the sightline passing trough the point $q_i^{\mathbf{I}}$ is:

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

A *3D* point Q_i on the sightline S_i can be expressed as:

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

The depth μ_i is the distance of the point to the camera center. The cheirality property states that μ_i is positive [5].

As figure 2 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 or equal to the Euclidean distance between them on the template:

$$\|\hat{Q}_i - \hat{Q}_j\| \leq \|q_i^{\mathbf{T}} - q_j^{\mathbf{T}}\| = d_{ij}.$$

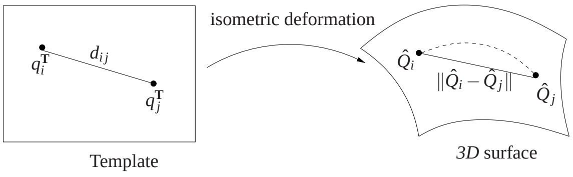

Figure 2: Inextensible object deformation. The template is deformed to the *3D* surface by an unknown isometric transformation. The dashed line is the geodesic curve between $\hat{Q}_i$ and $\hat{Q}_j$, it has the same length d_{ij} as the geodesic distance in the template. The Euclidean distance between the *3D* points is shorter due to the deformation.

As figure 3 illustrates, the coordinate frame system can be choosen such that the points Q_i and Q_j are parameterized by:

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

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

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

So there exists a real solution if and only if:

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

The bound μ_i is then computed from the whole set of correspondences (without loss of generality, 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*. The notation $i \longrightarrow i^\star$ reads 'point $i^\star$ constraints the upper bound of point i'. One can remark that this property is not symmetric: $i \longrightarrow j$ does not imply $j \longrightarrow i$. It is one of the reasons why this initialization is suboptimal, as explained in the next paragraph.

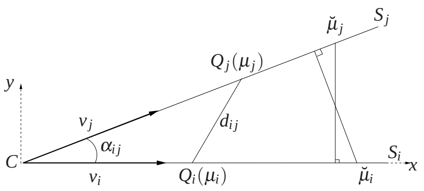

Figure 3: Point parameterization along the sightlines.

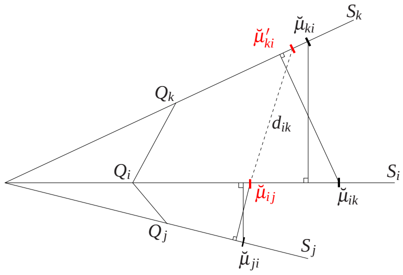

Figure 4: Bound refinement. Initial bound $\breve{\mu}_{ki}$ is refined to $\breve{\mu}'_{ki}$ by the process.

3.3 Refining the Bounds

The set of initial bounds is not optimal for the whole set of points. A suboptimal situation is represented on figure 4. 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 point Q_i can not be deeper than $\breve{\mu}_{ij}$. This gives the new bound $\breve{\mu}'_{ik}$ for the point Q_k.

We propose an iterative implementation of the 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 coherent.

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 the point i, it is thus given by the largest value of μ_j:

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

As figure 5 illustrates, this function has a global maximum:

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

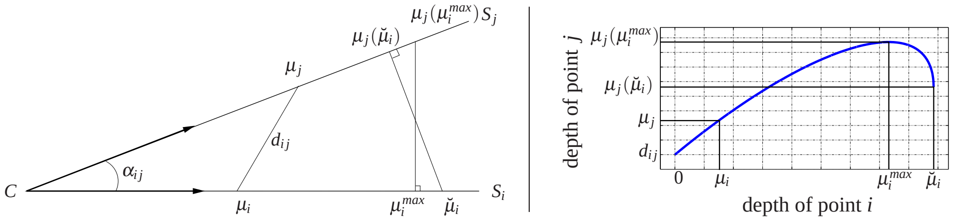

Figure 5: Relationship between the depth of the points under an inextensibility constraint. (left) The parameterization. (left) Graph of the function giving depth of point j against depth of point i when the distance between points i and j is equal to d_{ij}.

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

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

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. The outputs of the bounding process are the upper bound and the anchor point of each point. Both are used to recover the surface, as explained below.

4 Recovering the Surface

Our surface recovery procedure has two main steps:

- **Reconstruction of sparse *3D* points.** (§4.1) The *3D* points are computed using the upper bounds and the distances to their anchor points,
- **Reconstruction of a continuous surface.** (§4.2) The surface is expressed as an interpolation of these points, possibly using surface priors.

4.1 Finding a Sparse Set of *3D* Points

The set of 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:

$$\tilde{\mu}_i = \check{\mu}_i. \tag{4}$$

In practice, the error due to this approximation is small, as shown on figures 6, 7 and 8.

However, this is not satisfying from the inextensible point of view. Indeed, the distance between two upper bounds $\|Q(\check{\mu}_i) - Q(\check{\mu}_{i^\star})\|$ can be larger than their distance in the template $d_{ii^\star}$. For instance, when there is a symmetry between a point and its anchor point: $i \longrightarrow i^\star$ and $i^\star \longrightarrow i$, the distance is equal to $d_{ii^\star} \cdot \cos^{-1}\left(\frac{1}{2}\alpha_{ii^\star}\right)$. 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 decent results, the points depth such that these length equalities are satisfied are searched near the upper bounds.

The optimization can also handle other priors on the points. For instance, with a first order temporal smoother, it has the following form:

$$\begin{aligned} \tilde{\mu} \quad &= \quad \underset{\mu}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (\check{\mu}_i - \mu_i)^2 + \gamma(\mu_i(t) - \mu_i(t-1))^2 \right) \\ &\text{subject to} \quad \|Q_i - Q_{i^\star}\| = d_{ii^\star} \quad \text{for } i = 1..n \ , \end{aligned} \tag{5}$$

with μ the points depth vector, $\mu_i(t)$ the depth of the $i-$th point for the frame t and γ the balancing weight. This is a linear least squares problem under non-linear quadratic constraints, solved with the Levenberg-Marquardt algorithm [6] (the initial solution is given by equation (4)).

Table 2 shows the reconstruction time with and without the optimization process. The values given here are measured on our not optimal Matlab implementation.

number of points	50	100	250
reconstruction time without optimisation (ms)	13	35	176
reconstruction time with optimisation (ms)	82	143	444

Table 2: Reconstruction time against the number of points.

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 transferring a regular mesh designed on the template. In practice the mapping we choose is composed of three *2D* to *1D* Thin-Plate-Splines. These have proven efficient in the representation of deformable objects [1]. Getting a continuous surface makes it possible to deal with surface priors. 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, one can write this optimization as:

$$
\begin{aligned}
\tilde{\mu} \quad &= \quad \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{n} (\check{\mu}_i - \mu_i)^2 + \lambda \sum_{i=1}^{m} \left\| \frac{\partial^2 \Gamma}{\partial q^2}(q_i) \right\|^2 + \gamma \, \|q_i(t) - q_i(t-1)\|^2 \\
& \quad\; \text{subject to} \quad \|Q_i - Q_{i^\star}\| = d_{ii^\star} \quad \text{for } i = 1..n \; ,
\end{aligned}
\tag{6}
$$

with q a vertex of the mesh, m the number of vertices of the mesh and λ, γ the balancing weights controlling the trade-off between the distance to the bounds, the geometric smoothness and the temporal one. Fixing the deformation centers of the Thin-Plate-Splines in the template, problem (6) shows to be linear least squares under non-linear quadratic constaints. It can be similarly solved as problem (5).

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 square shape width 200mm. The feature points are randomly drawn on the shape. The reconstruction error for the $i-$th feature point is defined as:

$$
e(i) = \|\tilde{Q}_i - \hat{Q}_i\|. \tag{7}
$$

5.1 Number of Points

Figure 6 shows the average reconstruction error against the number of correspondences. The dashed curve represents the fast implementation error (equation (4)) and the full one corresponds to the optimized points under length constraints only (equation (5)). As expected, the error decreases thanks to the point optimization. The curves are decreasing: the higher the number of points, the lower the error. The accuracy of the reconstruction is related to two situations: the amount of deformation between the points and the orientation of the points with respect to the camera. Their respective influences are explained below. Due to lack of space, we do not show any quantitative results.

While deforming, the Euclidean distance between the *3D* points decreases. Since our algorithm is somehow 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 where the angle between the sightlines of the two points is maximum. This is the optimal orientation since it leads to a closer 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.* the points and their anchor points are well-oriented and the surface is not deformed too much between them.

5.2 Influence of the Noise

There are two ways to see the noise on our point primitives because one can arbitrarily choose in which image (the template or the image) the exact points are and in which one they are noisy. This choice induces differences in our algorithm: the ‘noise in the image’ changes the orientation of the sightlines whereas the ‘noise in the template’ modifies the reference distances d_{ij} between the points. Since our *3D* points are parameterized along their sightlines, we choose the second possibility. The noisy distances measured in the template lead to lower upper bounds if they are under evaluated. 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 computed distances in the template:

$$d_{ij} \longleftarrow d_{ij} + k. \tag{8}$$

This term reflects how reliable the distances are. Its efficiency is related to the noise level, as shown on figure 7. The curve presents a nice minimum at around 55% of the average noise magnitude, giving an empirical way to choose the term. This experimental curve shows also that it is better to over-estimate this parameter than to under-estimate it. However it is still difficult in practice to evaluate the noise magnitude. We ran all our real experiments with this parameter fixed to one pixel.

The precision of the reconstruction gracefully degrades with the noise magnitude, as shown on figure 8. 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 5.5mm.

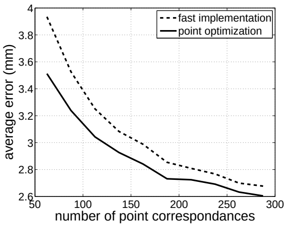

Figure 6: Error against number of correspondences.

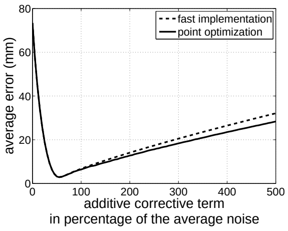

Figure 7: Influence of the corrective term on the error.

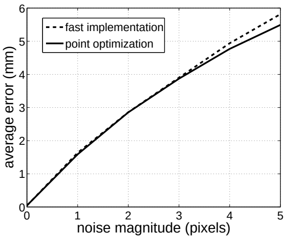

Figure 8: Error against noise magnitude in the image.

6 Experimental Results on Real Data

To evaluate the quality of our reconstructions, we have compared them to stereo-based *3D* reconstructions given by a calibrated pair of cameras. We report results on three objects: two A4-paper sheets and a can. The templates and the images of the deformations are shown in figure 1. The reconstructions are registered to the stereo-based reconstructions using a rigid transformation and a scale factor before evaluating the discrepancy.

The shape of the smoothly bent paper sheet is well recovered by our algorithm and looks like the stereo-based reconstruction. The reconstruction has been performed using 80 point correspondences. Both are shown in figure 9. The Root Mean Squared error is equal to 1.2mm, meaning that our reconstruction is closed to the stereo one.

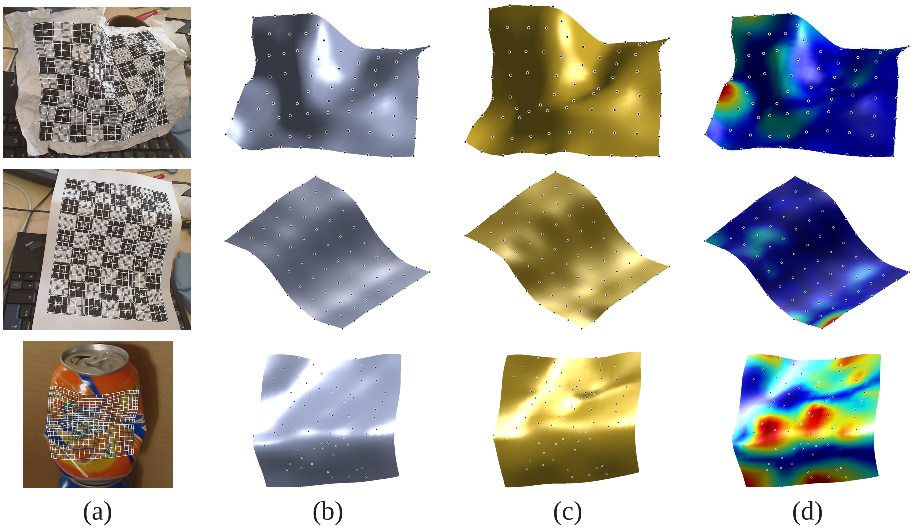

Figure 9: Reconstruction of the deformed paper sheets and the can. (a) Reprojections of our estimated surfaces. (b) Stereo reconstructions. (c) Our reconstructions. (d) Discrepancy map between the reconstructions.

The results we obtained with the creased paper sheet are shown in figure 9. The reconstruction has been done using 78 point correspondences. The *3D* shape from the stereo algorithm and from our method are similar. The RMS error is equal to 3.3mm. It is larger than the one of the smooth deformation. Actually, the creases make the deformations less adapted to our algorithm. However, the reconstruction accuracy we obtain is still very satisfaying.

We also used our method to reconstruct the deformed can shown in figure 1. For this example the template is not planar. We successfully recovered the shape using 72 point correspondences: the RMS error is 1.6mm. The results are shown on figure 9.

7 Conclusions

The algorithm we presented has been designed for 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. The results we obtained are convincing, and show that our algorithm brings a simple and effective solution to the monocular deformable reconstruction problem. Possible extensions include coupling our algorithm with a matching process for deformable environments such as [9]. The surface is represented by a *3D* warp parameterized by the correspondences. Better results could possibly be achieved by tuning these warp parameters through a cross-validation process to select optimal smoothing parameters.

References

[1] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *PAMI'89*.

[2] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR'00*.

[3] A. Del Bue. A factorization approach to structure from motion with shape priors. In *CVPR'08*.

[4] N. A. Gumerov, A. Zandifar, R. Duraiswami, and L. S. Davis. Structure of applicable surfaces from single views. In *ECCV'04*.

[5] R. Hartley. Cheirality. In *IJCV'97*.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Number ISBN: 0521540518. Cambridge University Press, 2004.

[7] J. Liang, D. DeMenthon, and D. Doermann. Geometric rectification of camera-captured document images. *PAMI'06*.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV'04*.

[9] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *IJCV'07*.

[10] M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3D tracking. In *ICCV'07*.

[11] M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *CVPR'07*.

[12] M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface deformation models for nonrigid 3D shape recovery. *PAMI'07*.

[13] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3D shape recovery. In *CVPR'08*.