

Single Image Deformable Surface 3D Recovery by Linear Programming

Wenjuan Ma

Department of Digital Media Technology, Zhejiang Sci-Tech University, Hangzhou, China

mawj2010@zstu.edu.cn

Keywords: deformable 3D reconstruction; single image reconstruction; linear programming

Abstract. We present a method for 3D shape reconstruction of inextensible deformable surfaces from a single image. The key of our approach is to represent the surface as a 3D triangulated mesh and formulate the reconstruction problem as a sequence of Linear Programming (LP) problems. We use a closed-form method to generate an initial structure, then refine this structure by solving the LP problem iteratively. The robustness and accuracy of our approach are evaluated qualitatively on real data.

Introduction

3D shape recovery of objects from 2D images is of central importance in computer vision. Many years of work in the field have led to several reliable approaches for reconstruction of rigid [1], multiple rigid [2] and articulated rigid objects [3]. However, many objects in the real world vary their shapes over time, such as faces, papers, clothes etc. The problem of reconstructing the shape of such deformable objects remains challenging.

Common methods for deformable structure recovery either introduce strong priors of deformations which makes it not adapted for objects undergoing complex deformations [4-7], or involve temporal consistency that requires a good initialization [8]. An alternative way is to build a deformation model using machine learning techniques [9-11], but this method lacks sufficient generality when the trained model is too specific.

In this paper, we describe a method to recover the 3D structure of a non-rigid object from a single image. More specifically, we dedicate to recover shapes of inextensible deformable surfaces. The central idea of our method is to represent the surface as a 3D triangulated mesh, and formulate the reconstruction problem as a sequence of Linear Programming (LP) problems. Compared with previous methods, ours neither involves smoothness constraints nor temporal consistency, which enables us to recover shapes of surfaces with various deformations from a single image.

Deformable Surface 3D Recovery by Linear Programming

Keypoint Constraints

The deformable surface is represented as a 3D triangulated mesh with n_v vertices. We denote the 3D coordinates of each vertex of the mesh by \mathbf{v}_i . The 3D structure of the mesh can be parameterized as a long vector \mathbf{V} by concatenating the three coordinates of all vertices, as: $\mathbf{V} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_{n_v}^T]$. Our method relies on 3D-to-2D keypoint correspondences between the mesh and the image. Assuming that a keypoint in the mesh is \mathbf{x}_i , it can be expressed in terms of its barycentric coordinate of the facet where this keypoint lies on, as:

$$\mathbf{x}_i = a_i \mathbf{v}_p + b_i \mathbf{v}_q + c_i \mathbf{v}_r = \mathbf{T}_i \quad (1)$$

Where \mathbf{v}_p , \mathbf{v}_q and \mathbf{v}_r are the vertices of the facet that \mathbf{x}_i lies on, a_i , b_i and c_i are the barycentric coordinate of \mathbf{x}_i , and \mathbf{T}_i is a transformation matrix dependent on the barycentric coordinate.

We assume that the camera to be calibrated, that is, the matrix of intrinsic parameters \mathbf{K} is known. Furthermore, we assume \mathbf{V} is in the camera coordinate, thus the projection of \mathbf{x}_i is:

$$\begin{bmatrix} u_i^1 \\ u_i^2 \\ 1 \end{bmatrix} = \mathbf{K} \mathbf{T}_i \mathbf{V}$$

The reprojection error with respect to image measurement $(\hat{u}_i^1, \hat{u}_i^2)^T$ is:

$$\left\| \begin{bmatrix} \frac{\mathbf{K}_1 \mathbf{T}_i \mathbf{V}}{\mathbf{K}_3 \mathbf{T}_i \mathbf{V}} - \hat{u}_i^1 \\ \frac{\mathbf{K}_2 \mathbf{T}_i \mathbf{V}}{\mathbf{K}_3 \mathbf{T}_i \mathbf{V}} - \hat{u}_i^2 \end{bmatrix} \right\| = \frac{1}{\mathbf{K}_3 \mathbf{T}_i \mathbf{V}} \left\| \begin{bmatrix} (\mathbf{K}_1 - \hat{u}_i^1 \mathbf{K}_3) \mathbf{T}_i \mathbf{V} \\ (\mathbf{K}_2 - \hat{u}_i^2 \mathbf{K}_3) \mathbf{T}_i \mathbf{V} \end{bmatrix} \right\| \quad (2)$$

where \mathbf{K}_1 , \mathbf{K}_2 and \mathbf{K}_3 are the first, second and third rows of \mathbf{K} respectively. Due to image noise, Eq.(2) can not be zero, and a variable γ is used as its upper bound. If γ is considered to be known, we have:

$$\left\| \begin{bmatrix} (\mathbf{K}_1 - \hat{u}_i^1 \mathbf{K}_3) \mathbf{T}_i \mathbf{V} \\ (\mathbf{K}_2 - \hat{u}_i^2 \mathbf{K}_3) \mathbf{T}_i \mathbf{V} \end{bmatrix} \right\| \leq \gamma \mathbf{K}_3 \mathbf{T}_i \mathbf{V}, \quad i = 1, \dots, n, \quad (3)$$

where n is the number of 3D-to-2D keypoint correspondences. As shown in [8], constraints in Eq. (3) are convex constraints and the recovery of \mathbf{V} can be solved by SOCP. Let us consider to replace the L_2 norm in Eq. (3) by L_∞ norm, we have:

$$\left\| \begin{bmatrix} (\mathbf{K}_1 - \hat{u}_i^1 \mathbf{K}_3) \mathbf{T}_i \mathbf{V} \\ (\mathbf{K}_2 - \hat{u}_i^2 \mathbf{K}_3) \mathbf{T}_i \mathbf{V} \end{bmatrix} \right\|_\infty \leq \gamma \mathbf{K}_3 \mathbf{T}_i \mathbf{V}, \quad i = 1, \dots, n, \quad (4)$$

or equivalently:

$$|(\mathbf{K}_j - \hat{u}_i^j \mathbf{K}_3) \mathbf{T}_i \mathbf{V}| \leq \gamma \mathbf{K}_3 \mathbf{T}_i \mathbf{V}, \quad i = 1, \dots, n, \quad j = 1, 2 \quad (5)$$

Eq. (5) gives $4n$ linear inequalities that represent n square pyramids centered in the camera, which means the n second order cones in (3) are replaced by n square pyramids in (5) when we use L_∞ instead of L_2 image error. Then the reconstruction can be achieved by solving the following LP problem:

find \mathbf{V}

$$\text{subject to } |(\mathbf{K}_j - \hat{u}_i^j \mathbf{K}_3) \mathbf{T}_i \mathbf{V}| \leq \gamma \mathbf{K}_3 \mathbf{T}_i \mathbf{V}, \quad i = 1, \dots, n, \quad j = 1, 2 \quad (6)$$

The minimal γ in (6) could be found using a *bisection* algorithm [12]. However, the results are always unacceptable due to image noise and ambiguities of perspective projection. Therefore, other constraints should be introduced to regularize the mesh shape.

Shape Constraints

For an inextensible surface, the most generic constraints that preserve original lengths of mesh edges are in the form:

$$\|\mathbf{v}_p - \mathbf{v}_q\| = l_r, \quad \langle p, q \rangle \in C, \quad (7)$$

where l_r is the original length of the edge linking vertices \mathbf{v}_p and \mathbf{v}_q , and $C = \{ \langle p, q \rangle \mid \mathbf{v}_p \text{ and } \mathbf{v}_q \text{ are neighboring vertices of the mesh} \}$. Since $\mathbf{v}_p - \mathbf{v}_q$ is a linear transformation of \mathbf{V} , we denote:

$$\mathbf{v}_p - \mathbf{v}_q = \mathbf{E}_r \mathbf{V}$$

where \mathbf{E}_r is a transformation matrix. Then the constraints in Eq. (7) can be expressed as:

$$\|\mathbf{E}_r \mathbf{V}\| = l_r, \quad r = 1, \dots, m, \quad (8)$$

where m is the number of mesh edges. The constraints in Eq. (8) are typically non-convex constraints and cannot be involved in the LP problem (6) directly. Below we will describe a linearization method that allows us to deal with these non-convex terms using an efficient LP solver in a sequential manner.

Suppose we start with an initial point \mathbf{V}_0 and seek for a better point \mathbf{V}_1 in the neighborhood of \mathbf{V}_0 . Point \mathbf{V}_1 can be expressed as $\mathbf{V}_1 = \mathbf{V}_0 + \delta_0$. So the problem now is to identify an appropriate vector δ_0 . In general, consider a scenario where we are in the k -th iteration and try to update point \mathbf{V}_k to point $\mathbf{V}_{k+1} = \mathbf{V}_k + \delta_k$. The constraints in (8) in this case become:

$$\|\mathbf{E}_r (\mathbf{V}_k + \delta_k)\| = l_r, \quad r = 1, \dots, m,$$

which can be expressed as:

$$2\mathbf{V}_k^T \mathbf{E}_r^T \mathbf{E}_r \delta_k + \delta_k^T \mathbf{E}_r^T \mathbf{E}_r \delta_k = l_r^2 - \mathbf{V}_k^T \mathbf{E}_r^T \mathbf{E}_r \mathbf{V}_k, \quad r = 1, \dots, m \quad (9)$$

Now if we remove the second term on the left-hand side of Eq. (9), we have:

$$2\mathbf{V}_k^T \mathbf{E}_r^T \mathbf{E}_r \delta_k \approx l_r^2 - \mathbf{V}_k^T \mathbf{E}_r^T \mathbf{E}_r \mathbf{V}_k, \quad r = 1, \dots, m \quad (10)$$

The m linear equality constraints in (10) can be put together as:

$$\mathbf{F}_k \delta_k = \mathbf{g}_k \quad (11)$$

where

$$\mathbf{F}_k = \begin{bmatrix} 2\mathbf{V}_k^T \mathbf{E}_1^T \mathbf{E}_1 \\ 2\mathbf{V}_k^T \mathbf{E}_2^T \mathbf{E}_2 \\ \vdots \\ 2\mathbf{V}_k^T \mathbf{E}_m^T \mathbf{E}_m \end{bmatrix}, \quad \mathbf{g}_k = \begin{bmatrix} l_1^2 - \mathbf{V}_k^T \mathbf{E}_1^T \mathbf{E}_1 \mathbf{V}_k \\ l_2^2 - \mathbf{V}_k^T \mathbf{E}_2^T \mathbf{E}_2 \mathbf{V}_k \\ \vdots \\ l_m^2 - \mathbf{V}_k^T \mathbf{E}_m^T \mathbf{E}_m \mathbf{V}_k \end{bmatrix} \quad (12)$$

Since Eq. (11) is valid as long as $\delta_k^T \mathbf{E}_r^T \mathbf{E}_r \delta_k \rightarrow 0$, we add another set of constraints as:

$$\|\mathbf{E}_r \delta_k\|_\infty \leq \eta, \quad r = 1, \dots, m \quad (13)$$

where η is an upper bound. Using L_∞ norm here makes (13) be a set of linear constraints, as:

$$\|\mathbf{E}_r^j \delta_k\|_\infty \leq \eta, \quad r = 1, \dots, m, \quad j = 1, 2, 3, \quad (14)$$

where \mathbf{E}_r^j represents the j -th row of \mathbf{E}_r . Now the reconstruction problem can be formulated as:
minimize η

subject to $|(\mathbf{K}_j - \hat{\mathbf{u}}_i^j \mathbf{K}_3) \mathbf{T}_i (\mathbf{V}_k + \delta_k)| \leq \gamma \mathbf{K}_3 \mathbf{T}_i (\mathbf{V}_k + \delta_k), \quad i = 1, \dots, n, \quad j = 1,$

$$\|\mathbf{E}_r^j \delta_k\|_\infty \leq \eta, \quad r = 1, \dots, m, \quad j = 1, 2, 3$$

$$\mathbf{F}_k \delta_k = \mathbf{g}_k \quad (15)$$

Eq. (15) defines an LP problem whose optimization variable is $[\delta_k^T, \eta]^T$ of dimension $3n_v + 1$, where $3n_v$ is the dimension of δ_k as well as \mathbf{V}_k . From an initial point \mathbf{V}_0 , we iteratively update \mathbf{V}_k , i.e. set $\mathbf{V}_{k+1} = \mathbf{V}_k + \delta_k$, by solving (15). This iteration continues until $\eta \rightarrow 0$.

Initialization

To some extent the effectiveness of our algorithm depends on how an initial point \mathbf{V}_0 is chosen. We use an idea that is similar to [13] to generate the initial point.

Suppose \mathbf{c} is the camera center, \mathbf{x}_i and \mathbf{x}_j are two keypoints in the camera coordinate $(\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathbf{z})$, \mathbf{u}_i and \mathbf{u}_j are image measurements of \mathbf{x}_i and \mathbf{x}_j respectively in the image coordinate $(\mathbf{o}, \mathbf{u}, \mathbf{v})$. Apparently, \mathbf{x}_i lies on the sightline coming through \mathbf{c} and \mathbf{u}_i , and \mathbf{x}_j lies on the sightline coming through \mathbf{c} and \mathbf{u}_j , which gives us a triangle $\Delta \mathbf{x}_i \mathbf{c} \mathbf{x}_j$. We denote $\|\mathbf{x}_i\| = s_i$, $\|\mathbf{x}_j\| = s_j$, $\angle \mathbf{x}_i \mathbf{c} \mathbf{x}_j = \alpha_{ij}$, and we have:

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = s_i^2 + s_j^2 - 2s_i s_j \cos \alpha_{ij}$$

where α_{ij} can be computed from \mathbf{u}_i and \mathbf{u}_j . For an inextensible deformable surface, the Euclidean distance between \mathbf{x}_i and \mathbf{x}_j is lower or equal to the geodesic distance on the surface due to the deformation. We denote this geodesic distance by d_{ij} , and we have:

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = s_i^2 + s_j^2 - 2s_i s_j \cos \alpha_{ij} \leq d_{ij}^2 \Rightarrow (s_j - s_i \cos \alpha_{ij})^2 \leq d_{ij}^2 - s_i^2 \sin^2 \alpha_{ij}$$

It can be shown s_j has a real solution if and only if:

$$d_{ij}^2 - s_i^2 \sin^2 \alpha_{ij} \geq 0 \Rightarrow s_i \leq \frac{d_{ij}}{|\sin \alpha_{ij}|} \quad (18)$$

Eq. (18) gives an upper bound of s_i , and the minimum upper bound of s_i can be computed from the whole set of keypoints, as:

$$s_i^* \leq \min_{\substack{j=1, \dots, n \\ j \neq i}} \frac{d_{ij}}{|\sin \alpha_{ij}|} \quad (19)$$

As has been pointed out in [13], $s_i^* \mathbf{e}_i$ can be seen as an estimation of \mathbf{x}_i , where \mathbf{e}_i is the unit vector in the direction of the sightline from \mathbf{c} to \mathbf{u}_i . Since $\mathbf{x}_i = \mathbf{T}_i \mathbf{V}$, we have the following linear system:

$$\mathbf{M} \mathbf{V} = \mathbf{N}, \quad (20)$$

where

$$\mathbf{M} = \begin{bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_n \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} s_1^* \mathbf{e}_1 \\ \vdots \\ s_n^* \mathbf{e}_n \end{bmatrix}.$$

However, solving this linear system may have ambiguities. For each unconstrained vertex, we introduce a linear equality. Adding all these equalities to (20) gives an extended linear system:

$$\tilde{\mathbf{M}}\mathbf{V} = \tilde{\mathbf{N}}, \quad (21)$$

where $\tilde{\mathbf{M}}$ is a full-rank matrix. Solving the linear system (21) yields an initial structure.

Experimental Results

Our approach is evaluated on real images. We use two objects for experiments: a paper sheet and a piece of cloth. The experiments are implemented under the Matlab environment and SeDuMi[14] is used as the LP solver. For each object, we capture an image sequence using a calibrated camera with a resolution of 1024×768. The keypoints on the mesh and their barycentric coordinates are extracted from a reference image in which the surface is in front of the camera without deformations. Then the 3D-to-2D keypoint correspondences are established between the reference image and the input one using SIFT [15]. Some reconstruction results are shown in Fig. 1. The results show that our approach can correctly recover 3D structures of surfaces with various deformations.

Conclusions

In this paper we present a method for 3D shape recovery of inextensible deformable surfaces. In our approach, the reconstruction problem is formulated as a sequence of LP problems. The LP problem consists of data constraints which are 3D-to-2D keypoints correspondences and shape constraints which are designed to retain original lengths of mesh edges. A closed-form linear method is used to generate an initial structure, and then the structure is refined by solving the LP problem iteratively. Compared with previous methods, our approach involves neither smoothness constraints nor temporal consistency, which enables us to reconstruct structures of surfaces with various kinds of deformations from a single image.

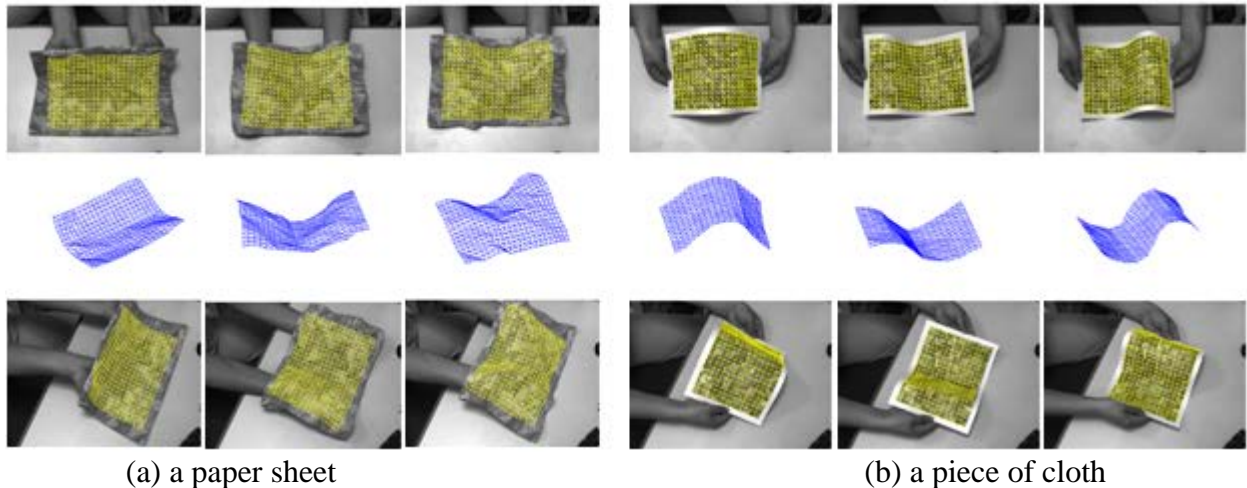


Fig. 1 Reconstruction of objects with smooth deformations.

The top row are the original images with reprojected mesh. The middle row are the reconstructed meshes seen from a different view. The bottom row are images of another camera with reprojected mesh.

Acknowledgement

This work was supported by Zhejiang Provincial Natural Science Foundation Youth Project

(LQ12F03012) and the Open Project Program of the National Laboratory of Pattern Recognition.

References

- [1] Vincent Lepetit and Pascal Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.
- [2] Rene Vidal, Yi Ma, Stefano Soatto, and Shankar Sastry. Two-view multibody structure from motion. *International Journal of Computer Vision*, 68(1):7–25, Jun. 2006.
- [3] David Forsyth, Okan Arıkan, Leslie Ikemoto, James Brien, and Deva Ramanan. Computational studies of human motion: Part 1, tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision*, 1(2):77–254, 2006.
- [4] Alessio Del Bue and Lourdes Agapito. Non-rigid stereo factorization. *International Journal of Computer Vision*, 66(3):193–207, Feb. 2006.
- [5] Lorenzo Torresani, Aaron Hertzmann, and Christoph 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, 2008.
- [6] Florent Brunet, Adrien Bartoli, Richard I. Hartley, Monocular template-based 3D surface reconstruction: Convex inextensible and nonconvex isometric methods. *Computer Vision and Image Understanding* 125: 138-154, 2014.
- [7] E. Serradell, M. Pinheiro, R. Sznitman, J. Kybic and F. Moreno-Noguer et al. Non-Rigid Graph Registration using Active Testing Search, accepted in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, num. 3, p. 625-638, 2015.
- [8] Mathieu Salzmann, Richard Hartley, and Pascal Fua. Convex optimization for deformable surface 3-d tracking. In *Proceedings of International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.
- [9] Iain Matthews and Simon Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, Nov. 2004.
- [10] Mathieu Salzmann, Raquel Urtasun, and Pascal Fua. Local deformation models for monocular 3d shape recovery. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, USA, 2008.
- [11] T. D. Ngo, J. Östlund and P. Fua. Template-based Monocular 3D Shape Recovery using Laplacian Meshes, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, num. 1, p. 172-187, 2016.
- [12] Fredrik Kahl. Multiple view geometry and the L1 norm. In *Proceedings of International Conference on Computer Vision*, Beijing, China, 2005.
- [13] Mathieu Perriollat, Richard Hartley, and Adrien Bartoli. Monocular template-based reconstruction of inextensible surfaces. In *Proceedings of the 19th British Machine Vision Conference*, Leeds, UK, 2008.
- [14] Jos Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.
- [15] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.