Deformable Surface 3D Reconstruction from a Single Image by Linear Programming

Wenjuan Ma, Shusen Sun
Department of Digital Media Technology,Zhejiang Sci-Tech University,
Hangzhou, China
[e-mail:mawj2010@zstu.edu.cn]

Received May 26, 2016; revised December 28, 2016; accepted February 7, 2017; published June 30, 2017

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. The LP problem consists of data constraints which are 3D-to-2D keypoint correspondences and shape constraints which are designed to retain original lengths of mesh edges. We use a closed-form method to generate an initial structure, then refine this structure by solving the LP problem iteratively. 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. The robustness and accuracy of our approach are evaluated quantitatively on synthetic data and qualitatively on real data.

Keywords: deformable 3D reconstruction, single image reconstruction, linear programming, convex optimization

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.

https://doi.org/10.3837/tiis.2017.06.018 **ISSN : 1976-7277**

1. 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[4-13], or involve temporal consistency that requires a good initialization [14]. An alternative way is to build a deformation model using machine learning techniques [15-18], 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. Our method has three main advantages compared with previous ones. Firstly, formulating the problem as an LP problem has great advantages since the LP can be solved quite reliably and efficiently. Secondly, our method does not involve any temporal consistency, which means that it can recover the structure from a single image. Thirdly, there are no smoothness constraints introduced in our method which makes it applicable for surfaces with various kinds of deformations such as those of **Fig. 1.**

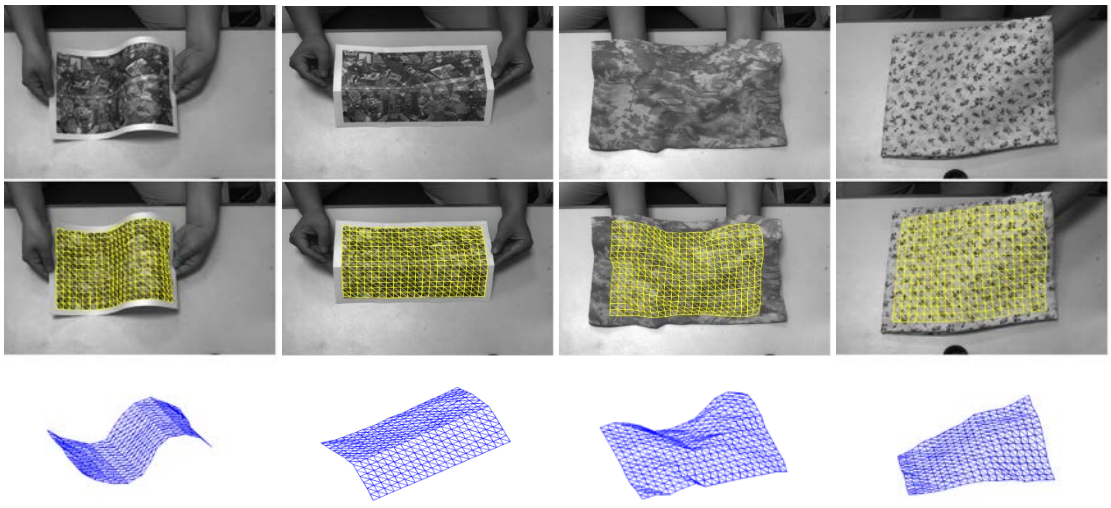


Fig. 1. Examples of deformable surface reconstruction from a single image using our approach. The top row are the original images. The middle row are the images with reprojected meshes. The bottom row are the reconstructed triangulated meshes seen from a different view

The paper is organized as follows. Previous work is reviewed in Section 2. The deformable reconstruction using LP is presented in Section 3. Some implementation details are stated in Section 4. Finally, experimental results on both synthetic and real data are reported in Section 5, followed by conclusions in Section 6.

2. Previous Work

Various methods have been proposed for the recovery of non-rigid 3D structures from 2D images. Generally, these approaches can be divided into three categories: structure-from-motion based methods, machine learning based methods and physics based methods.

Most approaches [5-13]within the non-rigid structure from motion (NRSFM) framework stem from Tomasi and Kanade's factorization algorithm which is originally designed for rigid structures [4]. Bregler et al. [5] are the first to use a factorization-based method for the recovery of non-rigid structure and motion, in which the 3D shape in each frame is formulated as a linear combination of a set of basis shapes. Brand [6] recasts NRSFM as a constrained optimization problem and solves this problem by directly minimizing the metric geometric errors. Torresani et al. [7] model the time-varying shape as a rigid transformation combined with a non-rigid deformation. This model is a form of Probabilistic Principal Components Analysis (PPCA) shape model whose parameters can be learned in the reconstruction process. In the case of perspective cameras, Xiao et al. [8] present a closed-form solution for perspective reconstruction given the assumption that there exists a set of independent deformable basis shapes. Del Bue et al. [9] formulate NRSFM as a constrained non-linear minimization adding priors on the degree of deformability of each point and then optimize accordingly for the perspective and deformation parameters. Bartoli et al. [10] propose a low-rank structure-from-motion method which handles missing data, automatically selects the number of deformation modes and makes use of several different priors. Agudoet. al. [11-13] propose a series of methods to simultaneously recover camera pose and 3D shape of non-rigid and potentially extensiblesurfaces from a monocular image sequence. Most NRSFM methods make strong assumptions about the deformations which makes it more suitable for objects undergoing small deformations (although some NRSFM methods could deal with relatively large deformations such as human motion). Besides, NRSFM is a kind of template-free method which needs the whole image sequence to compute the solution and thus is not suited for reconstruction on the fly.

Machine learning based methods[15-18] try to learn a deformation model from the training data, and apply the model for new data. Active appearance models (AAMs) are typical generative models for nonrigid objects and have been successfully applied for 3D face reconstruction [15,16]. AAM consists of a linear combination of shape bases and a linear combination of appearance bases which can be learned from training samples. Fitting an AAM to an image is obtained by minimizing the error between the input image and the closest model instance, which is a nonlinear optimization problem. However, the underlying linearity assumption makes AAMs only suitable for smoothly deformed objects. In fact, training a model that can be applied for general deformations requires complex nonlinear learning techniques and a large number of training samples with all kinds of deformations. Recent work shows that this kind of model can be obtained by learning its local deformation models, and combining them together to reconstruct global shapes [17]. However, the training samples are still not easy to obtain even though local patches have fewer degrees of freedom.

Physics based methods[14,19-29] introduce a prior knowledge of deformations and formulate the problem as an optimization problem. These approaches have been widely used for modeling and animation purposes in computer graphics [19]. In computer vision, Gay-Bellile et al. [20] present an 2D intensity-based non-rigid registration method with self-occlusion reasoning. This method constrains the 2D warp to shrink in self-occluded

regions while detecting them based on this property and successfully deals with extreme self-occlusions. However, this method is only suited for 2D registration and hard to be generalized to 3D cases. McInerney et al. [21] present a physics based approach for recovering 3D shapes of nonrigid objects using a 3D elastically deformable balloon model that is based on a thin-plate under tension spline. Although this method is very effective, introducing smoothness constraints into the objective function limits its applicabilities. Salzmann et al. [22]express the deformations as a linear combination of modes and present a closed-form solution to recover the shape of a non-rigid inelastic surface from an individual image. This method obtains the modes by applying Principal Component Analysis to a matrix of registered training meshes in deformed configurations, which makes it only suitable for surfaces that have similar deformation modes with the training examples. Perriollat et al. [23] use 3D bounds on the keypoints as distance constraints to recover structures of inextensible deformable surfaces. This method uses pairwise constraints to get an initial bound for each keypoint and refines them as a whole iteratively. The effectiveness of this method is based on establishing perfect keypoint correspondences, that is, there shouldn't be mismatching between the template and the image which is rare in practice. Chhatkuli et al. [28] theoretically analysis why existing convex numerical and analytical solutions for isometricsurface reconstruction from a single image may be unstable under perspective and weak-perspective conditions, and propose a new algorithm which works under all imaging conditions. Salzmann et al. [14] represent surfaces as triangulated meshes and disallow large changes of edge orientation between two consecutive frames, and formulate the tracking problem as an second order cone programming (SOCP) feasibility problem which can be efficiently solved. However, this method introduces strong constraints to bound the vertex displacements from on frame to the next, which makes it only applicable for the tracking context in which the shape for the first frame is known.

3. Linear Programming for Deformable Surface Reconstruction

To formulate general deformations, the surface is represented as a 3D triangulated mesh in this paper, and the purpose of reconstruction is to retrieve the 3D position of each mesh vertex from a single image. We will show that the recovery of the mesh could be formulated as a sequence of LP problems which can be efficiently solved. In this section, we briefly introduce the LP problem first, then describe how to formulate the reconstruction problem as an LP formulation, and give a method to reduce the number of unknowns of this LP problem.

3.1 Linear Programming

Recently, there has been interest in solving geometric vision problems such as triangulation and camera resectioning using L_∞ minimization [30,31]. The key advantage of using the L_∞ is that the problem can be formulated as an SOCP feasibility problem with a single minimum and can be effectively solved. A general SOCP problem has the form:

$$\begin{aligned} &\text{minimize } \mathbf{f}^{\mathrm{T}}\mathbf{x} \\ &\text{subject to } \|\mathbf{A}_i\mathbf{x}+\mathbf{b}_i\| \le \mathbf{c}_i^{\mathrm{T}}\mathbf{x}+d_i i=1,\ldots,m \end{aligned}$$

There are great advantages to recognize or formulate a problem as an SOCP problem. The most basic advantage is that the problem can then be solved very reliably and efficiently [32]. In [33], it shows that LP could be adopted in place of the SOCP in these geometric vision

problems. The SOCP includes LP as a special case, and a general LP has the form:

minimize $\mathbf{f}^{\mathrm{T}}\mathbf{x}$
subject to $\mathbf{Ax} \leq \mathbf{b}$

In the following of this paper, $\|\cdot\|$ represents the L_2 norm, and $\|\cdot\|_\infty$ represents the L_∞ norm.

3.2 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: $\mathbf{V} = \left[\mathbf{v}_1^T, \mathbf{v}_2^{\mathrm{T}}, \cdots, \mathbf{v}_{n_v}^{\mathrm{T}}\right]$. 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:

$$\mathbf{x}_i = a_i\mathbf{v}_p + b_i\mathbf{v}_q + c_i\mathbf{v}_r = \mathbf{T}_i\mathbf{V}, \tag{1}$$

where $\mathbf{v}_p$, $\mathbf{v}_q$ and $\mathbf{v}_r$ are the vertices of the facet that $\mathbf{x}_{\mathrm{i}}$ 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 $\mathbf{x}_{\mathrm{i}}$ 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 $(\hat{\mathrm{u}}_i^1, \hat{\mathrm{u}}_i^2)^{\mathrm{T}}$ is:

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

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:

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

where n is the number of 3D-to-2D keypoint correspondences. As shown in [23,24], 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:

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

or equivalently:

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

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:

$$\begin{aligned} &\text{find} && \mathbf{V} \\ &\text{subject to} && \left| (\mathbf{K}_\mathrm{j} - \hat{\mathrm{u}}_i^j \mathbf{K}_3)\mathbf{T}_i\mathbf{V} \right| \leq \gamma \mathbf{K}_3 \mathbf{T}_i \mathbf{V}, \qquad i = 1, \cdots, n, \quad j = 1,2 \end{aligned} \tag{6}$$

The minimalγin (6) could be found using a *bisection* algorithm [30]. 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.

3.3 Shape Constraints

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

$$\left\| \mathbf{v}_p - \mathbf{v}_q \right\| = l_r, \qquad < p, q > \in C, \tag{7}$$

where l_r is the original length of the edge linking vertices $\mathbf{v}_p$ and $\mathbf{v}_q$, and C={$< p, q >$ $\left| \mathbf{v}_p \right.$ and $\mathbf{v}_q$ are neighboring vertices of the mesh}. Note that Eq. (7) indicates that the nodal connectivity is used to define the shape constraints. Since $\mathbf{v}_p - \mathbf{v}_q$ is a linear transformation of $\mathbf{V}$, we denote:

$$\mathbf{v}_p - \mathbf{v}_q = \mathbf{E}_\mathrm{r}\mathbf{V}$$

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

$$\| \mathbf{E}_r \mathbf{V} \| = l_r, \qquad r = 1, \cdots, m, \tag{8}$$

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:

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

which can be expressed as:

$$2\mathbf{V}_k^T\mathbf{E}_r^T\mathbf{E}_r\delta_k + \delta_k^T\mathbf{E}_r^T\mathbf{E}_r\delta_k = l_r^2 - \mathbf{V}_k^T\mathbf{E}_r^T\mathbf{E}_r\mathbf{V}_k, \qquad r = 1, \cdots, m \tag{9}$$

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

$$2\mathbf{V}_k^T\mathbf{E}_r^T\mathbf{E}_r\delta_k \approx l_r^2 - \mathbf{V}_k^T\mathbf{E}_r^T\mathbf{E}_r\mathbf{V}_k, \qquad r = 1, \cdots, m \tag{10}$$

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

$$\mathbf{F}_k\delta_k = \mathbf{g}_k \tag{11}$$

Where

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

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

$$\|\mathbf{E}_r\delta_k\|_\infty \le \eta, \qquad r = 1, \cdots, m \tag{13}$$

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

$$\left\|\mathbf{E}_r^j\delta_k\right\|_\infty \le \eta, \qquad r = 1, \cdots, m,\ j = 1,2,3, \tag{14}$$

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

$$\begin{aligned} &\text{minimize } \eta \\ &\text{subject to } \left|\left(\mathbf{K}_\mathrm{j} - \hat{\mathrm{u}}_i^j\mathbf{K}_3\right)\mathbf{T}_i(\mathbf{V}_k + \delta_k)\right| \le \gamma\mathbf{K}_3\mathbf{T}_i(\mathbf{V}_k + \delta_k), \qquad i = 1, \cdots, n, \quad j = 1,2 \\ &\qquad\qquad \left\|\mathbf{E}_r^j\delta_k\right\|_\infty \le \eta, \qquad r = 1, \cdots, m, \qquad j = 1,2,3 \\ &\qquad \mathbf{F}_k\delta_k = \mathbf{g}_k \end{aligned} \tag{15}$$

Eq. (15) defines an LP problem whose optimization variable is $\left[\delta_k^T, \eta\right]^T$ 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 $\eta \to 0$.

3.4 Reducing the Number of Unknowns

Since the last constraint in (15) is a set of m linear equations with $3n_v$ unknowns

($3n_v > m$), there are infinitely many solutions for these equations. It is well known that all these solutions can be linearly parameterized using singular value decomposition (SVD) of the coefficient matrix $\mathbf{F}_k$. Since the m linear equations in (15) are independent, the rank of matrix $\mathbf{F}_k$ is m and its SVD is given by $\mathbf{F}_k = \mathbf{U}\sum\mathbf{S}$. All the solutions of $\mathbf{F}_k\delta_k = \mathbf{g}_k$ are then given by:

$$\delta_k = \mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k \tag{16}$$

where $\mathbf{F}_k^+$ denotes the Moore-Penrose pseudo inverse of $\mathbf{F}_k$, which can also be computed using the SVD of $\mathbf{F}_k$, $\mathbf{S}_k$ is a matrix of size $3n_v$ by $3n_v - m$ which consists of the last $3n_v - m$ columns of matrix $\mathbf{S}$, and φ_k is an arbitrary vector of dimension $3n_v - m$. With (16), the problem in (15) becomes:

$$\begin{aligned} &\text{minimize} \quad \eta \\ &\text{subject to} \quad \left|\left(\mathbf{K}_j - \hat{\mathrm{u}}_i^j\mathbf{K}_3\right)\mathbf{T}_i(\mathbf{V}_k + \mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k)\right| \le \gamma\mathbf{K}_3\mathbf{T}_i(\mathbf{V}_k + \mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k), \\ &\qquad\qquad i = 1,\cdots,n, \quad j = 1,2 \\ &\qquad \left|\mathbf{E}_r^j(\mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k)\right| \le \eta,\ r = 1,\cdots,m,\ j = 1,2,3 \end{aligned} \tag{17}$$

Eq. (17) is an LP problem whose optimization variable is $\left[\delta_k^T, \eta\right]^T$ of dimension $3n_v - m + 1$. Take a 16×22 vertex rectangular mesh for example, the dimension of $\mathbf{V}$ is $3n_v = 3 \times 16 \times 22 = 1056$, so (15) is an LP problem of dimension 1057. This mesh has 981 edges, i.e.$m = 981$, so (17) is an LP problem of dimension 76. As we can see, the dimension of (17) is much smaller than that of (15) after reducing the number of unknowns.

4. Implementation Details

In this section, we address some implementation details when we solving the LP problem in (17). These include a closed-form method to identify an initial point $\mathbf{V}_0$, and a refinement process to find the final result $\mathbf{V}_k$.Note that when dealing with videos, the temporal consistency is usually employed for initialization, i.e. using the structure recovered in the previous frame as the initialization in current frame. In this paper we focus on deformable reconstruction from a single Image, thus we omit this temporal consistency prior here.

4.1 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 [23] to generate the initial point.

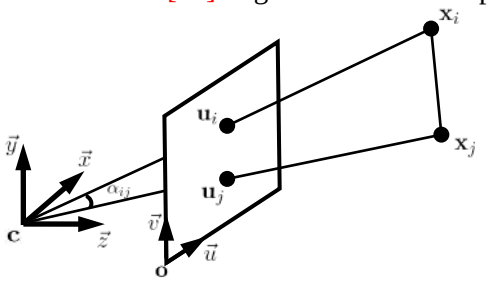


Fig. 2. The 3D location of a keypoint is on the sightline coming through its 2D image measurement and the camera center.

As shown in **Fig. 2**, $\mathbf{c}$ 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 $\triangle \mathbf{x}_i\mathbf{c}\mathbf{x}_j$. 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:

$$\begin{gathered}\|\mathbf{x}_i - \mathbf{x}_j\|^2 = s_i^2 + s_j^2 - 2s_i s_j \cos\alpha_{ij} \le d_{ij}^2 \Longrightarrow \\ \left(s_j - s_i \cos\alpha_{ij}\right)^2 \le d_{ij}^2 - s_i^2 \sin^2\alpha_{ij}\end{gathered}$$

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

$$\begin{gathered}d_{ij}^2 - s_i^2 \sin^2\alpha_{ij} \ge 0 \implies \\ s_i \le \frac{d_{ij}}{|\sin\alpha_{ij}|}\end{gathered} \tag{18}$$

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:

$$s_i^* \le \min_{\substack{j=1,\cdots,n \\ j\neq i}} \frac{d_{ij}}{|\sin\alpha_{ij}|} \tag{19}$$

As has been pointed out in [23], $s_i^*\mathbf{e}_i$ 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:

$$\mathbf{M}\mathbf{V} = \mathbf{N}, \tag{20}$$

where

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

However, solving this linear system may have ambiguities and the reason is as follows. Each vertex of the mesh belongs to several adjacent facets. An example is shown in Fig. 3, where $\mathbf{v}_p$ is a vertex and the facets it belongs to are shown in blue and gray. If all these facets do not contain keypoints, $\mathbf{M}$ will be rank deficient, in other words, $\mathbf{v}_p$ is an unconstrained vertex that can move freely. To remove this ambiguity, we regularize the

surface at each unconstrained vertex. We choose each unconstrained vertex such that the surface at this vertex is as locally flat as possible, that is, setting $\mathbf{v}_p$ to be the average of its four neighboring vertices, as:

$$\mathbf{v}_p = \frac{1}{4}\left(\mathbf{v}_q + \mathbf{v}_r + \mathbf{v}_s + \mathbf{v}_t\right) \tag{21}$$

where $\mathbf{v}_q, \mathbf{v}_r, \mathbf{v}_s$ and $\mathbf{v}_t$ are four neighboring vertices of $\mathbf{v}_p$ as shown in **Fig. 3**.

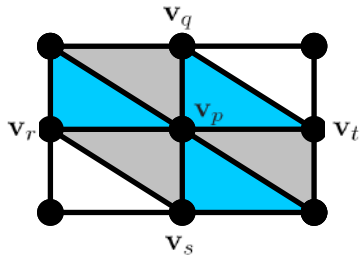


Fig. 3. An vertex $\mathbf{v}_p$ of the mesh belongs to several adjacent facets which are shown in blue and gray.

For each unconstrained vertex, we introduce a linear equality (21) (if $\mathbf{v}_p$ is on the boundary of the mesh, we let it to be the average of its two neighboring vertices). Adding all these equalities to (20) gives an extended linear system:

$$\widetilde{\mathbf{M}}\mathbf{V} = \widetilde{\mathbf{N}}, \tag{22}$$

where $\widetilde{\mathbf{M}}$ is a full-rank matrix. Solving the linear system (22) yields an initial structure. However, this linear system is extremely sensitive to outliers. In practice, after solving this system, we remove the keypoint with maximum reprojection error and re-solve this system until the maximum reprojection error is less than 5 pixels, and then use this result as the initial mesh $\mathbf{V}_0$. **Fig. 4(a)** shows an example of $\mathbf{V}_0$ generated by this linear initialization method. As we can see, the reprojection of $\mathbf{V}_0$ in **Fig. 4(a)** is generally fine, but its overall shape is wrong due to depth ambiguities and outliers.

4.2 Refinement

Given the initial point $\mathbf{V}_0$ and an upper bound of reprojection errorsγ, we can iteratively solve the problem (17) to refine the mesh. In the k-th iteration, once a solution φ_k of (17) is calculated, we (i) obtain δ_k using (16), and set $\mathbf{V}_{k+1} = \mathbf{V}_k + \delta_k$, then let $k = k + 1$; (ii) use the updated point to re-evaluate $\mathbf{F}_k$ and $\mathbf{g}_k$ using (12), compute the SVD of $\mathbf{F}_k$ and then $\mathbf{F}_k^+$ and $\mathbf{S}_k$; and (iii) solve (17) with the updated data to obtain a new φ_k. Theoretically, this iteration should continue until $\delta_k^T \mathbf{E}_r^T \mathbf{E}_r \delta_k \to 0$ as $\eta \to 0$ which means Eq. (10) and (9) are equivalent to each other. In practice, we do not have to wait until $\eta \to 0$, but stop the iteration and accept $\mathbf{V}_k$ as the result if the difference between each edge length, $\|\mathbf{E}_r \mathbf{V}_k\|$, and its original length, l_r, is below $0.001 l_r$. If the LP problem (17) becomes *infeasible* in a certain iteration, it means there is no solution for the current$\gamma$and we say this$\gamma$is *infeasible*, otherwise we say this$\gamma$is *feasible*. The procedure of this refinement process can be summarized as Algorithm 1.

Algorithm 1 Refinement

Require: The initial structure $\mathbf{V}_0$ and an upper bound of reprojection errors γ
$k = 0$
repeat
  Solve the LP problem (17) to obtain a solution
  if the LP problem (17) does not have a solution **then**
    γ is *infeasible* and stop the algorithm.
  end if
  Calculateδ_k using Eq. (16)
  $\mathbf{V}_{k+1} = \mathbf{V}_k + \delta_k$
  $k = k + 1$
  Re-evaluate $\mathbf{F}_k$ and $\mathbf{g}_k$ using Eq. (12)
  Calculate $\mathbf{F}_k^+$ and $\mathbf{S}_k$from the SVD of $\mathbf{V}_k$
until $\max_r |\|\mathbf{E}_r\mathbf{V}_k\| - l_r| < 0.001 l_r$
γ is *feasible* and output $\mathbf{V}_k$

In Algorithm 1, a relatively large γtends to be *feasible* (e.g.$\gamma = 10$). However, γ may still be *infeasible* if large outliers exist. In this situation, we continuously increase γ until a *feasible* γ is found. The procedure of finding a *feasible* γ is summarized as Algorithm 2.

Algorithm 2 Finding a *feasible* γ

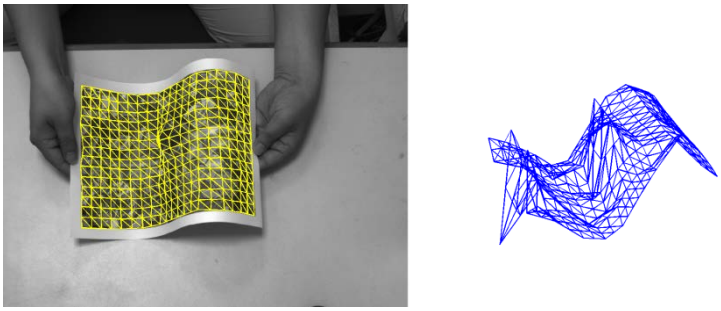
Require: The initial structure $\mathbf{V}_0$ and an upper bound of reprojection errors γ
Do Algorithm 1 with $\mathbf{V}_0$ and γ
if γ is *infeasible***then**
  $\gamma = 2 \times \gamma$
  Go to step 1
end if
Output $\mathbf{V}_k$ and γ

Once finding a *feasible* γ, we decrease γ, use the refined structure $\mathbf{V}_k$ as a new initial point $\mathbf{V}_0$, and redo the refinement process until the minimum *feasible* γ, $\gamma_{\min}$, is found. The procedure of finding $\gamma_{\min}$ is essentially heuristic which can be summarized as Algorithm 3.

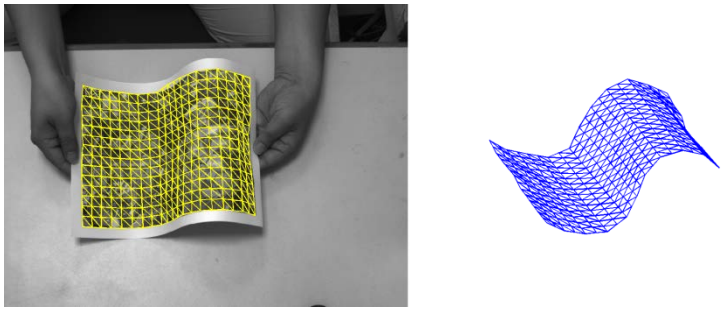
Algorithm 3 Computing$\gamma_{\min}$

Require: A refined mesh $\mathbf{V}_k$ and the *feasible*γ obtained by Algorithm 2
$\mathbf{V}_0 = \mathbf{V}_k,\ \gamma_{\text{feasible}} = \gamma,\ \gamma_{\text{step}} = \gamma/2$
repeat
  $\gamma = \gamma_{\text{feasible}} - \gamma_{\text{step}}$
  Do Algorithm 1 with $\mathbf{V}_0$ and γ
  if γ is *infeasible***then**
    $\gamma_{\text{step}} = \gamma_{\text{step}}/2$
  else
    $\mathbf{V}_0 = \mathbf{V}_k,\ \gamma_{\text{feasible}} = \gamma,\ \gamma_{\text{step}} = \gamma/2$
  end if
until $\gamma_{\text{step}} < 0.05$
$\gamma_{\min} = \gamma_{\text{feasible}}$

We should note that the keypoint correspondences may still contain outliers even though most of them have been removed in the initialization step. As [34], If $\gamma_{\min}$ is more than 2 pixels, we throw out the set of keypoint correspondences whose reprojection error equals $\gamma_{\min}$ and redo Algorithm 2 and 3 until $\gamma_{\min}$ is less than 2 pixels. **Fig. 4(b)** shows an example of a final recovered mesh generated by refining the initial mesh $\mathbf{V}_0$ in **Fig. 4(a)**. As we can see, the overall shape of $\mathbf{V}_0$ has been adjusted to a quite acceptable position after the iterative refinement process.



(a) An initial mesh $\mathbf{V}_0$ generated by the linear initialization method.



(b) The final reconstructed mesh generated by refining the initial mesh $\mathbf{V}_0$

Fig. 4. Mesh initialization and refinement. In both (a) and (b), on the left is the original image with reprojected mesh and on the right is the mesh seen from a different view.

5. Experimental Results

The performance of our algorithm was evaluated with both synthetic and real data. All the experiments are implemented under the Matlab environment and SeDuMi[35] is used as the LP solver.

5.1 Synthetic Data

A 8cm×11cm triangulated mesh is used as the synthetic data. We apply forces to certain vertices of this 88-vertex mesh and keep mesh edges to be their original lengths, which generates two 50-frame synthetic sequences. Synthetic sequence 1 is a smoothly deformed mesh, synthetic sequence 2 is a mesh with sharp folds, and synthetic sequence 3 is a mesh with complex (local) deformations. We randomly choose four 3D points in each facet of the mesh and project these points on an image plane using a perspective projection matrix, which gives us a set of 3D-to-2D point correspondences at each frame.

For comparison, Salzmann et al.'s method [14], Chhatkuli et al.'s method [28], Perriollat et

al.'s method [23] and Matlab's constrained optimization function,*fmincon*, were also used for the synthetic data.

1)Salzmann et al.'s method [14] is a recursive method, that is, the structure recovered in the previous frame would be used in current frame, thus the surface structure for the first frame should be given in advance. In the experiments, the ground-truth of the first frame is used as the initial structure when we use the method in [14].

2)Chhatkuli et al.'s method [28] is able to recover the structure of an isometric surface from a single image as ours. This method uses the two-fold non-holonomic solution todepth-gradient, and is considered as state-of-the-art single view deformable reconstruction method currently.

3) Perriollat et al.'s method [23] has two steps. In the first step, a suboptimal solution is computed by using pairwise constraints. In the second step, an iterative refinement process considers the upper bounds as a whole and tunes all of them to get a fully compatible set of bounds. We should note that our method uses the idea that is similar with the first step of [23] to generate an initial mesh $\mathbf{V}_0$, as described in Section 4.1.

4) Matlab's constrained optimization function, *fmincon*, can handle non-convex constraints. When using *fmincon* for surface reconstruction, we design the objective function in *fmincon* as the sum of square of keypointreprojection errors, and design the constraint functions to retain original lengths of mesh edges. *fmincon* needs an initial structure for each frame, and we set the initialization as the flat position.

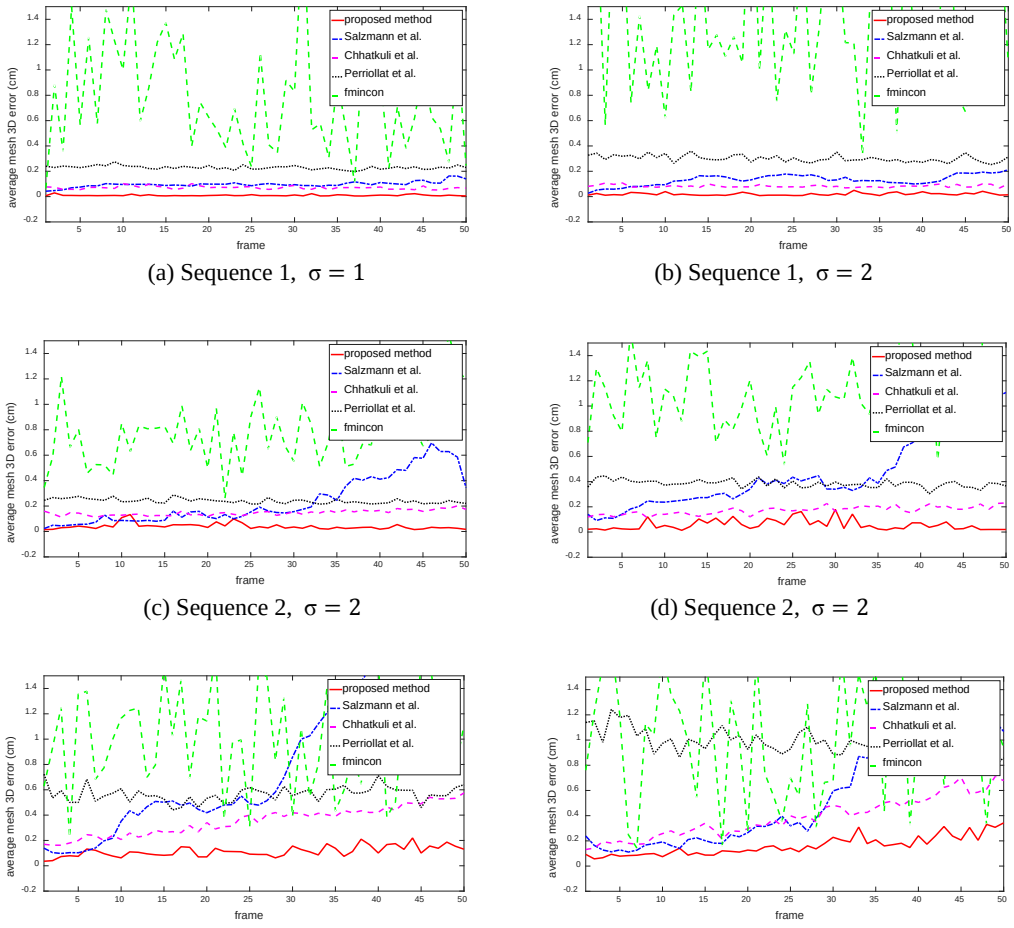
Experiment I. In the first experiment, we evaluate the robustness of five different methods to noise. For this experiment, we add Gaussian noise with mean zero and variance one and two to all the image point locations at each frame. **Fig. 5** shows the average 3D distance between reconstructed mesh vertices and the ground-truth of each frame for different noise levels. **Fig. 6** shows some reconstruction results using five different methods when adding Gaussian noise with variance two.

The results show that our approach gives more stable and accurate results compared with the other three methods. Though less accurate than ours, Chhatkuli et al.'s method [28] also generates quite acceptable results. Salzmann et al.'s method [14] becomes worse in the second half of the sequence 2 and 3. This is because the recursive nature of [14] makes it only applicable for the tracking context, which means it cannot reconstruct the surface from a single image. Perriollat et al.'s method [23] performs less accurate than our method, Chhatkuli et al.'s method and Salzmann et al.'s method for sequence 1, but outperforms the method in [14] at the end of sequence 2 and 3. The *fmincon* function always get trapped in local minimum and gives unacceptable structures due to the non-convexity of the corresponding optimization problem. Furthermore, *fmincon* function takes about 30 minutes to process one frame as opposed to 10 seconds of our approach, 4 seconds of Salzmann et al.'s method [14], 5 seconds of Chhatkuli et al.'s method [28], and 3 seconds of Perriollat et al.'s method [23] on a 3GHz standard PC.

Experiment II. In the second experiment, we evaluate the robustness of five different methods to outliers. For this experiment, we first add Gaussian noise with mean zero and variance two to all the image point locations, and then add Gaussian noise with mean zero and variance ten to 30% and 60% of the image point locations respectively which can be regarded as outliers. **Fig. 7** shows the average 3D distance between reconstructed mesh vertices and the ground-truth of each frame using five different methods.

The results show that the *fmincon* function is extremely sensitive to outliers and its 3D

error quickly exceeds the scope of the graph. As outliers increase, Perriollat et al.'s method [23] biases toward the ground-truth quickly. We think the reason is that the effectiveness of the method in [23] is based on perfect keypoint correspondences, thus it becomes worse when adding large outliers. Salzmann et al.'s method [14], Chhatkuli et al.'s method [28] and ours are all robust in dealing with large outliers, and our methods again performs much more stable and accurate than others, especially on synthetic sequence 3.

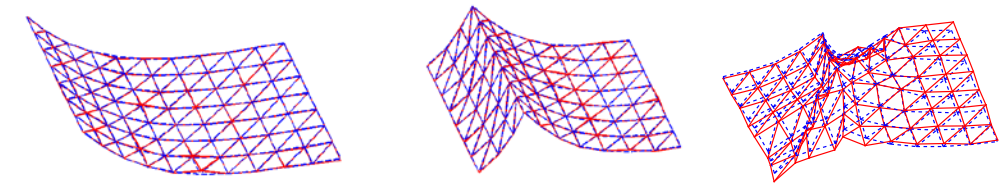


(a) Sequence 1, $\sigma = 1$ (b) Sequence 1, $\sigma = 2$

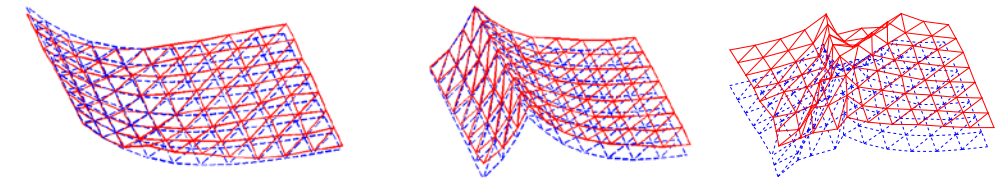
(c) Sequence 2, $\sigma = 2$ (d) Sequence 2, $\sigma = 2$

(e) Sequence 3, $\sigma = 1$ (f) Sequence 3, $\sigma = 2$

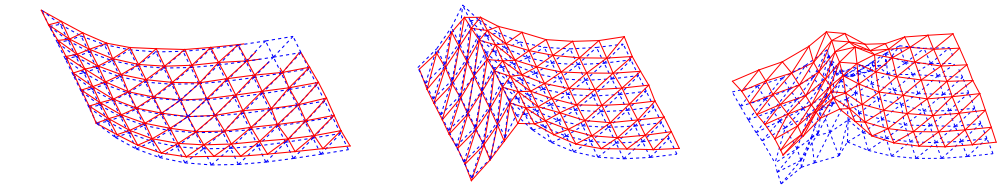
Fig. 5. Average 3D distance between reconstructed mesh vertices and the ground-truth using five different methods for each frame of the synthetic data. (a)and (b) are the results of synthetic sequence 1 when adding Gaussian noise with variances one and two respectively. (c) and (d) are the results of synthetic sequence 2.(e) and (f) are the results of synthetic sequence 3. In all the graphs, results obtained by the proposed method are shown in solid red, results obtained by Salzmann et al.'s recursive method [14] are shown in dash-dotted blue, results obtained by Chhatkuli et al.'s method [28] are shown in dashed purple,results obtained by Perriollat et al.'s method [23] are shown in dotted black, and results obtained by Matlab's *fmincon* function are shown in dashed green.



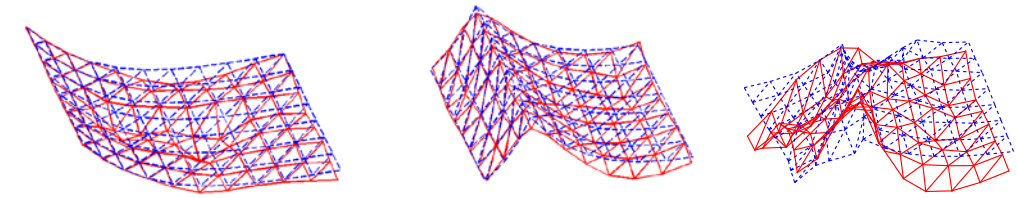
(a) Reconstruction by the proposed method



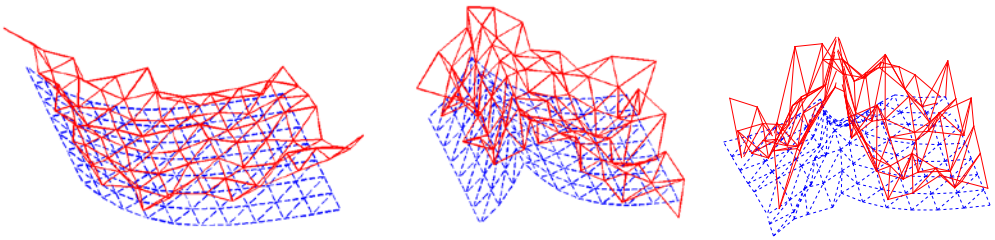
(b) Reconstruction by Salzman et al.'s method



(c) Reconstruction by Chhatkuli et al.'s method

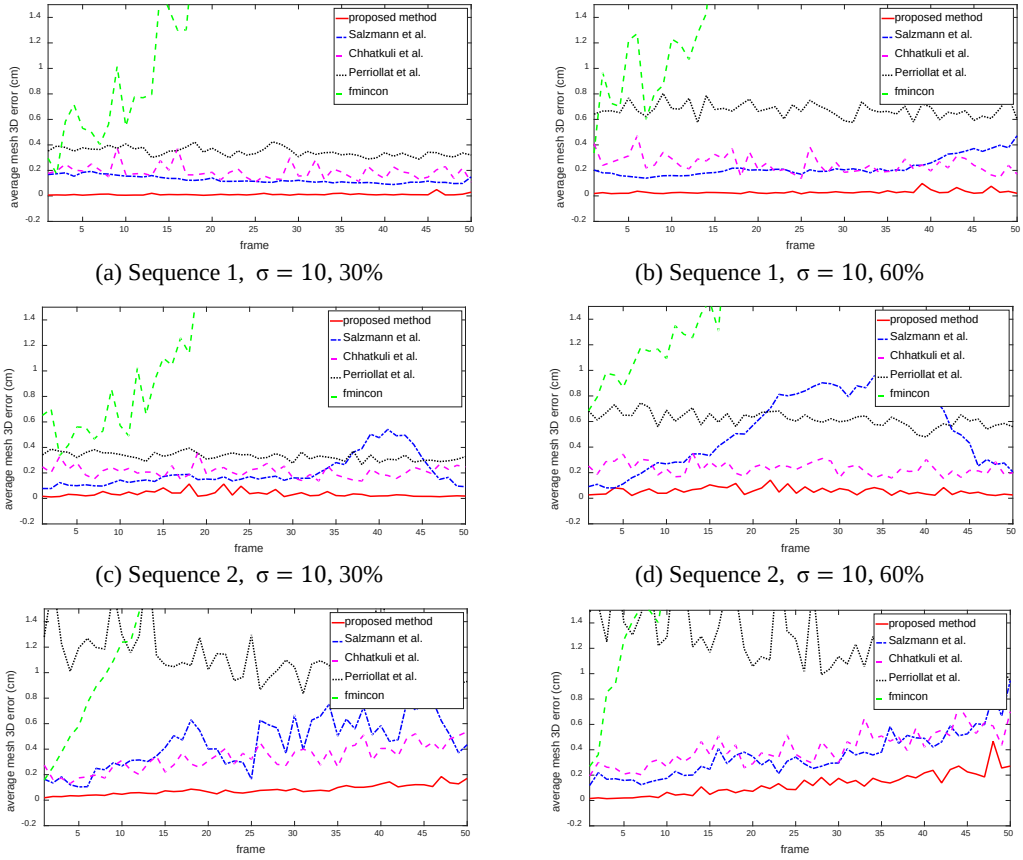


(d) Reconstruction by Perriollat et al.'s method



(e) Reconstruction by Matlab's*fmincon* function

Fig. 6. Some reconstruction results of the synthetic data using five different methods, when adding Gaussian noise with mean zero and variance two. In all the graphs, the reconstructed mesh is shown in solid red, and the ground-truth is shown in dashed blue.



(a) Sequence 1, $\sigma = 10$, 30%

(b) Sequence 1, $\sigma = 10$, 60%

(c) Sequence 2, $\sigma = 10$, 30%

(d) Sequence 2, $\sigma = 10$, 60%

(e) Sequence 3, $\sigma = 10$, 30%

(f) Sequence 3, $\sigma = 10$, 60%

Fig. 7. Average 3D distance between reconstructed mesh vertices and the ground-truth using five different methods for each frame of the synthetic data. (a) and (b) are the results of synthetic sequence 1 when first adding Gaussian noise with mean zero and variance two to all the image point locations, and then adding Gaussian noise with mean zero and variance ten to 30% and 60% of the image point locations respectively. (c) and (d) are the results of synthetic sequence 2.(e) and (f) are the results of synthetic sequence 3.In all the graphs, results obtained by the proposed method are shown in solid red, results obtained by Salzmann et al.'s recursive method [14] are shown in dash-dotted blue, results obtained by Chhatkuli et al.'s method [28] are shown in dashed purple,results obtained by Perriollat et al.'s method [23] are shown in dotted black, and results obtained by Matlab's *fmincon* function are shown in dashed green.

5.2 Real Data

Our approach is qualitatively evaluated on real images. We use three objects for experiments: a paper sheet, a piece of cloth and a bed-sheet. 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 [36]. We should note that, even though our approach is applied for image sequences, there is no linking between two consecutive frames.

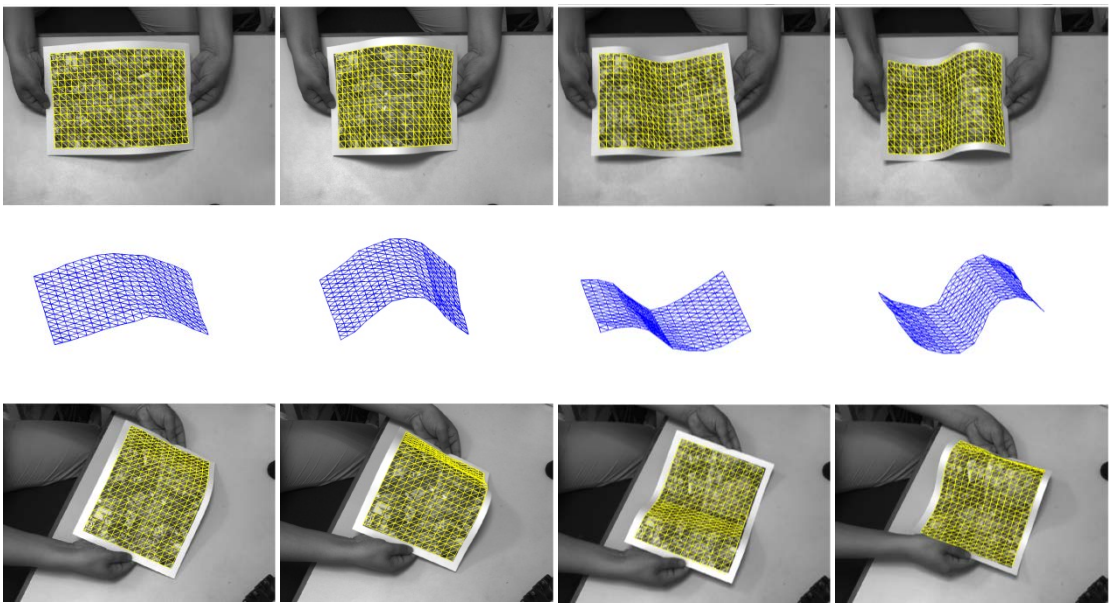


Fig. 8. Reconstruction of a paper sheet with smooth deformations. The top row are the original images with reprojected mesh. The middle row are the reconstructed meshes seen from a different view. The bottom roware images of another camera with reprojected mesh.

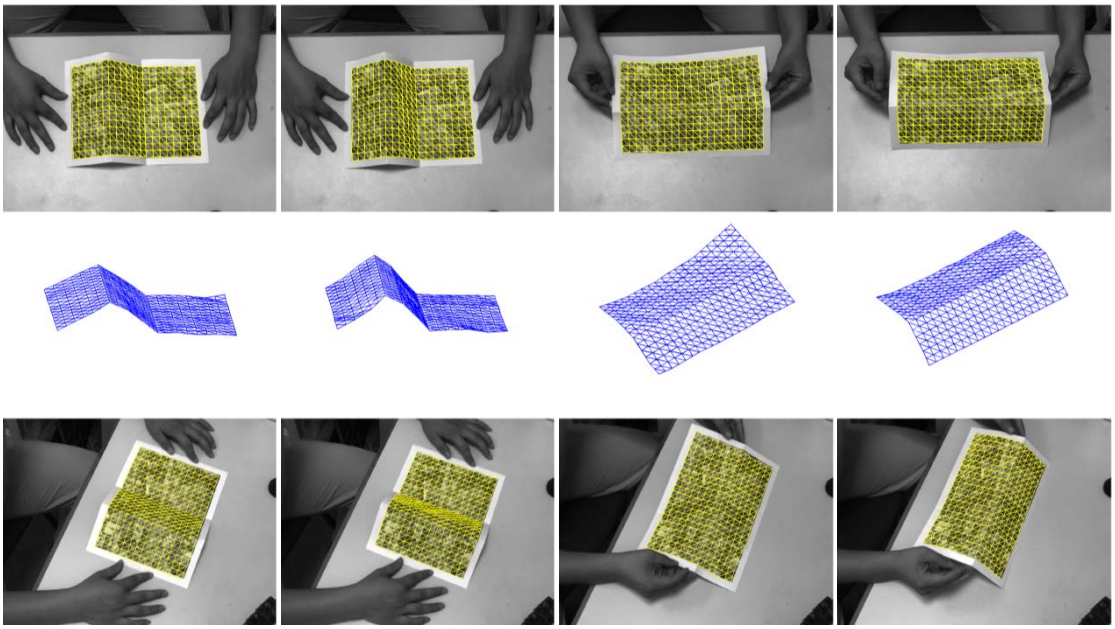


Fig. 9. Reconstruction of a paper sheet with sharp folds. The top row are the original images with reprojected mesh. The middle row are the reconstructed meshes seen from a different view. The bottom row are images of another camera with reprojected mesh.

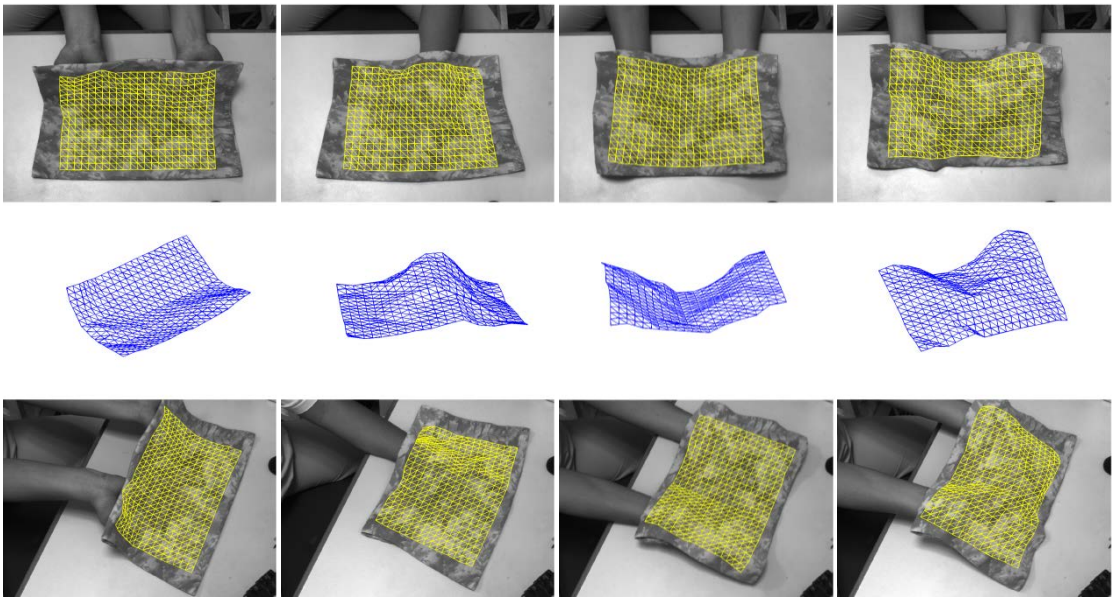


Fig. 10. Reconstruction of a piece of cloth. The top row are the original images with reprojected mesh. The middle row are the reconstructed meshes seen from a different view. The bottom row are images of another camera with reprojected mesh.

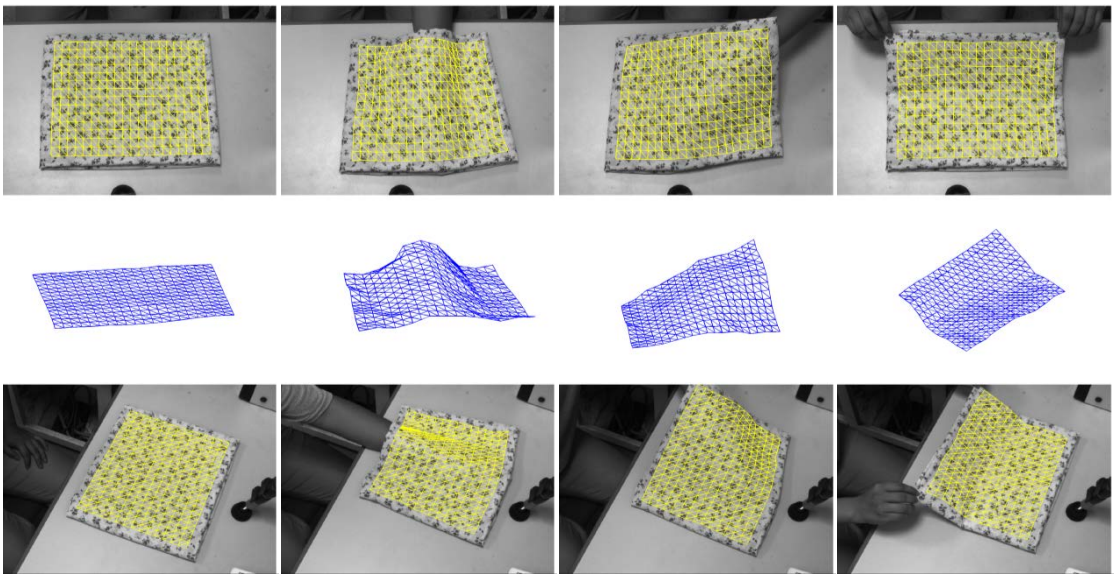


Fig. 11 Reconstruction of a bed-sheet. The top row are the original images with reprojected mesh. The middle row are the reconstructed meshes seen from a different view. The bottom row are images of another camera with reprojected mesh.

Some reconstruction results are shown in **Fig. 8** to **Fig. 11**. The results show that our approach can correctly recover 3D structures of surfaces with smooth, sharp and other complex deformations. The reprojection of the recovered mesh fits the original image very well (the first rows in the results), and the 3D shape of the mesh is quite reasonable since all the mesh edges retain their original lengths (the second rows in the results). Even projecting

the mesh to a image taken from another view, they are also closely matched (the third rows in the results).

We also qualitatively evaluatedthe proposedmethod on the public available deformable reconstruction dataset used in [14] and [22]. This dataset contains deformable objects with different types of motions and materials.Some reconstruction results of our method are shown in Fig. 12.The results show that the proposed approach could correctlyrecovered the 3D shapes of surfaces with different types ofdeformations and materials.

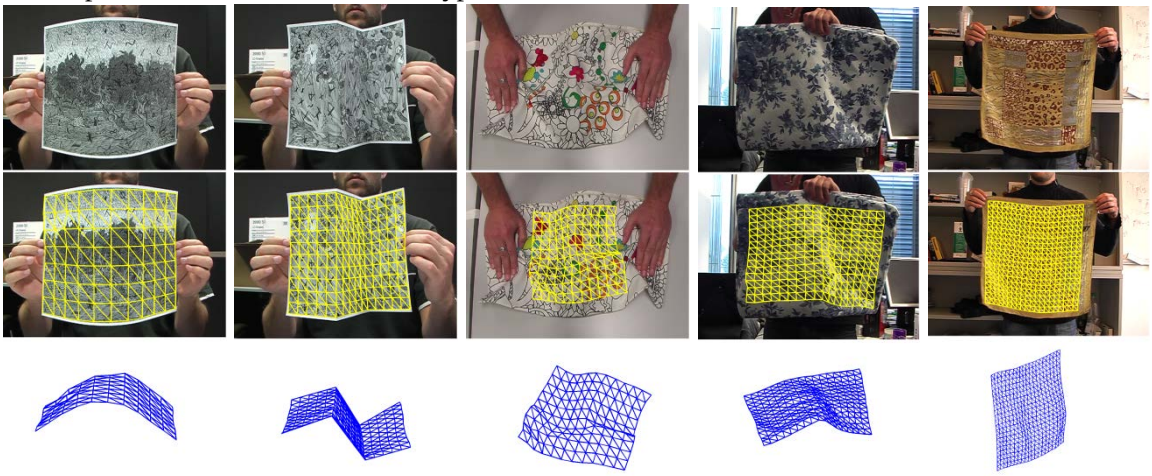


Fig. 12. Some reconstruction results of the public availabledataset used in [14,22]. The top row are the original images.The middle row areimages with reprojected mesh. The bottom row arereconstructed meshes seen from a different view

6. 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.

In future work, we will investigate how to involve the other image cues, such as edges or lighting, into our framework in order to increase the robustness of our method to occlusions. Besides, our future work also includes investigating how to extend our method to handle extensible deformable surfaces.

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.
Article (CrossRef Link).

[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.
Article (CrossRef Link).

[3] David Forsyth, OkanArikan, Leslie Ikemoto, James Brien, and Deva Ramanan, "Computational studies of human motion: Part 1, tracking and motion synthesis," *Foundations and Trends in ComputerGraphics and Vision*, 1(2):77–254, 2006. Article (CrossRef Link).
[4] Carlo Tomasi and Takeo Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, 9(2):137–154, Nov. 1992. Article (CrossRef Link).
[5] ChristophBregler, Aaron Hertzmann, and Henning Biermann, "Recovering non-rigid 3d shape from image streams," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2000. Article (CrossRef Link).
[6] Matthew Brand, "A direct method for 3d factorization of nonrigid motion observed in 2d," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005. Article (CrossRef Link).
[7] Lorenzo Torresani, Aaron Hertzmann, and ChristophBregler, "Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors," *IEEE Transactions on Pattern Analysis and MachineIntelligence*, 30(5):878–892, 2008. Article (CrossRef Link).
[8] Jing Xiao and Takeo Kanade, "Uncalibrated perspective reconstruction of deformable structures," in *Proc. of International Conference on Computer Vision*, 2005.Article (CrossRef Link).
[9] Alessio Del Bue and Lourdes Agapito, "Non-rigid stereo factorization," *International Journal of Computer Vision*, 66(3):193–207, 2006. Article (CrossRef Link).
[10] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd, "Coarse-to-fine low-rank structure-from-motion, in *Proc. of IEEE Computer Society Conference on Computer Visionand Pattern Recognition*, 2008. Article (CrossRef Link).
[11] A. Agudo, L. Agapito, B. Calvo, J. M. M. Montiel, "Good vibrations: A modal analysis approach for sequential non-rigid structure from motion," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2014. Article (CrossRef Link).
[12] A. Agudo and F. Moreno-Noguer, "Simultaneous pose and non-rigid shape with particle dynamics," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2015. Article (CrossRef Link).
[13] A. Agudo, F. Moreno-Noguer, B. Calvo and J.M.M. Montiel, "Sequential Non-Rigid Structure from Motion using Physical Priors," *IEEE Transactions on Pattern Analysis and MachineIntelligence*, 38(5):979-994, 2016. Article (CrossRef Link).
[14] Mathieu Salzmann, Richard Hartley, and Pascal Fua, "Convex optimization for deformable surface 3-d tracking," in *Proc. of International Conference on Computer Vision*, 2007. Article (CrossRef Link).
[15] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, Jun. 2001. Article (CrossRef Link).
[16] Iain Matthews and Simon Baker, "Active appearance models revisited," *International Journal of Computer Vision*, 60(2):135–164, 2004. Article (CrossRef Link).
[17] Mathieu Salzmann, Raquel Urtasun, and Pascal Fua, "Local deformation models for monocular 3d shape recovery," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. Article (CrossRef Link).
[18] T. D. Ngo, J. Östlund and P. Fua, "Template-based Monocular 3D Shape Recovery using Laplaci an Meshes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1): 172-187, 2 016. Article (CrossRef Link).
[19] AllaSheffer, Emil Praun, and Kenneth Rose, "Mesh parameterization methods and their applications," *Foundations and Trends in Computer Graphics and Vision*, 2(2):105–171, 2006. Article (CrossRef Link).
[20] V. Gay-Bellile, A. Bartoli, and P. Sayd, "Direct estimation of non-rigid registrations with image-based self-occlusion reasoning," in *Proc. of IEEE International Conference on Computer Vision*, 2007. Article (CrossRef Link).

[21] Tim McInerney and DemetriTerzopoulos, "A finite element model for 3d shape reconstruction and nonrigid motion tracking," in *Proc. of the International Conference on Computer Vision*, Berlin, Germany, 1993. Article (CrossRef Link).
[22] Mathieu Salzmann, Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua, "Closed-form solution to non-rigid 3d surface registration," in *Proc. of the 10th European Conference on ComputerVision*, 2008. Article (CrossRef Link).
[23] Mathieu Perriollat, Richard Hartley, and AdrienBartoli, "Monocular template-based reconstruction of inextensible surfaces," in *Proc. of the 19th British Machine Vision Conference*, 2008. Article (CrossRef Link).
[24] Florent Brunet, Adrien Bartoli, Richard I. Hartley, "Monocular template-based 3D surface recon struction: Convex inextensible and nonconvex isometric methods," *Computer Vision and Image Understanding* 125: 138-154, 2014. Article (CrossRef Link).
[25] E. Serradell, M. Pinheiro, R. Sznitman, J. Kybic and F. Moreno-Noguer et al., "Non-Rigid Graph Registration using Active Testing Search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3): 625-638, 2015. Article (CrossRef Link).
[26] A. Bartoli Y. Gerard, F. Chadebecq, and T. Collins, "On template-based reconstruction from a single view: analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2012. Article (CrossRef Link).
[27] A. Bartoli, D. Pizarro, and T. Collins, "A robust analytical solution to isometric shape-from-template with focal length calibration," in *Proc. of International Conference on Computer Vision*, 2013. Article (CrossRef Link).
[28] A. Chhatkuuli, D. Pizarro and A. Bartoli, Stable "Template-based isometric 3D reconstruction in all imaging conditions by linear least-squares," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2014. Article (CrossRef Link).
[29] S. Vicente and L. Agapito, "Soft inextensibility constraints for template-free non-rigid reconstruction," in *Proc. of the 12th European Conference on ComputerVision*, 2012. Article (CrossRef Link).
[30] Fredrik Kahl, "Multiple view geometry and the L_∞ norm," in *Proc. of International Conference on Computer Vision*, 2005. Article (CrossRef Link).
[31] QifaKe and Takeo Kanade, "Quasiconvex optimization for robust geometric reconstruction," in *Proc. of International Conference on Computer Vision*, 2005. Article (CrossRef Link).
[32] Stephen Boyd and LievenVandenberghe, *Convex optimization*, Cambridge University Press, 2004. Article (CrossRef Link).
[33] YongduekSeo and Richard Hartley, "A fast method to minimize L_∞ error norm for geometric vision problems," in *Proc. of International Conference on Computer Vision*, 2007. Article (CrossRef Link).
[34] Kristy Sim and Richard Hartley, "Removing outliers using the linfty norm," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2006. Article (CrossRef Link).
[35] Jos Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, 11-12:625–653, 1999. Article (CrossRef Link).
[36] David Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 60(2):91–110, 2004. Article (CrossRef Link).

Wenjuan Ma received the Ph.D degree in pattern recognition and intelligence system from Shanghai Jiao Tong University, China. Currently, she is a teacher in the department of Digital Media Technology, Zhejiang Sci-Tech University, China. Her current research interests are computer vision and image processing.

Shusen Sun received his Ph.D degree in computer science from Zhejiang University, China. Currently, he is a teacher in the department of Digital Media Technology, Zhejiang Sci-Tech University, China. His current research interests are image processing and virtual reality.