Pattern Recognition 43 (2010) 244–254

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Monocular 3D tracking of deformable surfaces using sequential second order cone programming☆

Shuhan Shen[a,*], Yuncai Liu[a], Wu-Sheng Lu[b]

[a]Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China
[b]Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada

ARTICLE INFO

Article history:
Received 7 July 2008
Received in revised form 19 March 2009
Accepted 29 June 2009

Keywords:
3D tracking
Deformable tracking
Second order cone programming
Convex optimization

ABSTRACT

Reconstructing structures of deformable objects from monocular image sequences is important for applications like visual servoing and augmented reality. In this paper, we propose a method to recover 3D shapes of deformable surfaces using sequential second order cone programming (SOCP). The key of our approach is to represent the surface as a triangulated mesh and introduce two sets of constraints, one for model-to-image keypoint correspondences which are SOCP constraints, another for retaining the original lengths of the mesh edges which are non-convex constraints. In the process of tracking, the surface structure is iteratively updated by solving sequential SOCP feasibility problems in which the non-convex constraints are replaced by a set of convex constraints over a local convex region. The shape constraints used in our approach is more generic than previous methods, that enables us to reliably recover surface shapes with smooth, sharp and other complex deformations. The capability and efficiency of our approach are evaluated quantitatively with synthetic image sequences and qualitatively with real image sequences.

1. Introduction

Recovery 3D shapes of objects from 2D monocular image sequences is of central importance in computer vision. Many years of work in the field have led to several reliable approaches for reconstruction of rigid [7,11], multiple rigid [6,23] and articulated rigid objects [5]. However, many objects in the real world vary their shapes over time, such as faces, papers, clothes, etc. The problem of reconstructing structures of these deformable objects remains challenging and has been a subject of current research.

For rigid objects, 3D to 2D correspondences between keypoints on objects and their image locations can be used for reliable recovery of object structures [12]. For deformable surfaces, however, estimating time-varying 3D shapes from monocular 2D point tracks is a severely under-constrained problem. To remove the ambiguities, many approaches have been proposed. Structure-from-motion based methods [24,22] formulate the object deformation as a linear combination of a set of shape bases. Machine learning based methods [4,14,18] learn a motion model from a set of training data. Physics based methods [15,16] formulate the physical properties of the surface as penalty functions. However, these approaches typically introduce constraints that prevent the surface from folding sharply, which makes them not adapted for reconstructing surfaces undergoing complex or sharp deformations such as those of Fig. 1.

Recent advances in solving geometric vision problems using convex optimization inspire a method to formulate the deformable surface reconstruction problem as a second order cone programming (SOCP) problem with a unique minimum which can be efficiently calculated using an SOCP solver [17]. This method models the surface as a triangulated mesh and introduces a set of constraints to stop the orientation of the mesh edges from changing irrationally between consecutive frames. These constraints are more generic than smoothness constraints and makes this method applicable for various kinds of deformations. Theoretically, however, the most generic shape constraints for an inextensible surface should be designed to prevent the mesh from expanding or shrinking which are typically non-convex constraints. In some sense, the shape constraints in [17] are only approximations of the generic non-convex constraints in order for them to be incorporated into an SOCP framework.

In this paper, we describe a method that deals with the generic non-convex constraints in a sequential manner that allows us to use the efficient SOCP solver. More specifically, the central idea of our approach is to iteratively update the 3D structure of the surface by solving sequential SOCP feasibility problems in which the

☆This work was supported by National Natural Science Foundation of China under Grant 60833009, National 973 Key Basic Research Program of China under Grant 2006CB303103, National 863 Program of China under Grant 2009AA01Z330 and Graduate Innovation Foundation of SJTU.

* Corresponding author.
E-mail address: shenshuhan@gmail.com (S. Shen).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.06.016

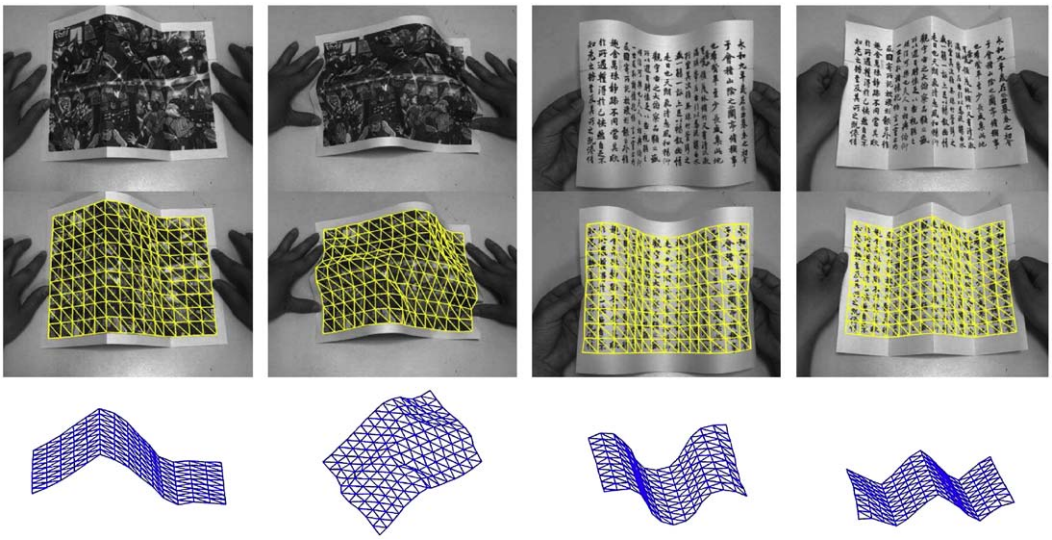

Fig. 1. Reconstructing the structure of a deformable surface from monocular image sequences using our approach. The top row are the original images. The middle row are the images with reprojected meshes. The bottom row are the reconstructed triangulated meshes seen from a different view.

non-convex constraints are replaced by a set of convex constraints over a local convex region. Therefore, our method on one hand uses the most generic shape constraints which enables us to remove the ambiguities without loss of generality, and on the other hand takes full advantage of SOCP as a reliable and efficient convex programming framework. Our experimental studies have demonstrated that the proposed method can reliably recover 3D structures of surfaces with smooth, sharp and other complex deformations such as those of Fig. 1.

The rest of the paper is organized as follows. In Section 2 we review previous work. In Section 3 we give the problem statement of deformable surface 3D tracking. Then the surface tracking approach using sequential SOCP is detailed in Section 4. Finally, the proposed approach is evaluated with both synthetic and real image sequences in Section 5, followed by conclusions in Section 6.

2. Previous work

Recovering 3D shapes of deformable objects from monocular 2D image sequences using keypoint correspondences is a severely under-constrained problem. Various approaches have been proposed to remove the ambiguities. Generally, these approaches can be divided into four categories called structure-from-motion based methods, machine learning based methods, physics based methods and volumetric methods. In this section we give a brief survey of these approaches.

Structure-from-motion methods for rigid object are well understood [7]. For deformable environment, some approaches have been proposed to introduce a prior knowledge of deformations. Xiao et al. [24] represent the deformable structure as a linear combination of shape bases, and present a two-step factorization approach for perspective reconstruction. This method is useful for scenes that contain independently moving rigid objects, but it is not suitable for general deformable structures. Torresani et al. [22] 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 process of reconstruction. This method, again, makes very strong assumptions about the deformations which makes it not suitable for objects undergoing large deformations.

Machine learning based methods 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 non-rigid objects and have been successfully applied for 3D face reconstruction [4,14]. 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 non-linear 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 non-linear learning methods and a large number of training samples with all possible 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 [18]. However, the training samples are still not easy to obtain even though local patches have fewer degrees of freedom.

Physics based methods 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, McInerney et al. [15] present a physics based approach for recovering the 3D shape and tracking the motion of non-rigid 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 a smoothness constraint into the objective function limits its applicabilities. Pilet et al. [16] present a real-time method for detecting deformable surfaces using keypoint matches between the 2D triangulated mesh and the image. This method is currently accepted as the most effective algorithm for 2D deformable surface detection, but it is hard to be generalized to 3D. Salzmann et al. [17] represent surfaces as triangulated meshes and disallow large changes of edge orientation between two consecutive frames, and formulate the tracking problem as an SOCP feasibility problem. This method yields a convex formulation with a unique minimum and enables us to handle highly deformable surfaces without adding unwarranted smoothness constraints. However, as has already been pointed out in Section 1, the shape constraints in [17] is only approximations to the most generic shape constraints in order for them to be incorporated into an SOCP framework.

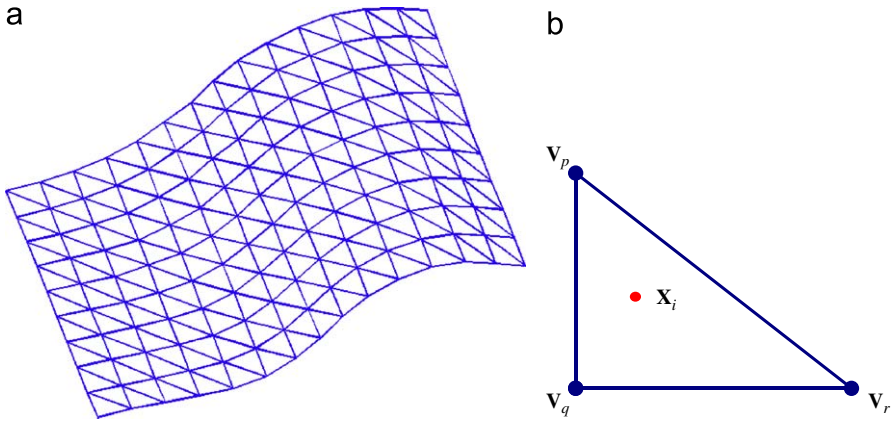

Fig. 2. 3D triangulated mesh. (a) A deformed rectangular mesh with 150 vertices. (b) A facet of the mesh with a keypoint.

Volumetric methods of space carving are popular approaches for object 3D reconstruction, mainly for smooth objects [1,2]. These methods are silhouette-based reconstruction methods: intersecting the visual cones generated by the silhouettes and the projection centers of each image, a 3D model can be determined. This 3D model is denominated as *visual hull* [10], a locally convex over-approximation of the volume occupied by an object. Volumetric methods represent the 3D space model by using voxels (volumetric pixels). The space of interest is divided into discrete voxels which are then classified into two categories: inside and outside. The union of all the inside voxels is an approximation of the visual hull. Volumetric methods are fully automatic approaches and suitable for many real applications. However, these methods are only applicable for static objects and need the whole image sequence to compute the solution which makes them not suited for tracking applications.

3. Problem statement

Modeling the surface as a 3D triangulated mesh is a popular way to represent the behavior of general deformations. In this paper, we follow the same way and model the surface as a N-vertex triangulated mesh. Fig. 2(a) shows an example of a rectangular mesh with 150 vertices. The 3D coordinates of N vertices are $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N$ respectively, and they can form a long vector as

$$\mathbf{v} = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_N \end{bmatrix}$$

Apparently, the overall shape of the mesh is controlled by $\mathbf{v}$ which should be calculated at each time instance in the process of tracking.

For tracking a deformable surface, we rely on establishing keypoint correspondences between the model and the image. Each keypoint on the model lies on a facet of the triangulated mesh, and it can be expressed in terms of its barycentric coordinates of the facet, as

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

where $\mathbf{x}_i$ is the 3D coordinate of the i-th keypoint. $\boldsymbol{v}_p$, $\boldsymbol{v}_q$ and $\boldsymbol{v}_r$ are the vertices of the facet that $\mathbf{x}_i$ lies on, as shown in Fig. 2(b). a_i, b_i and c_i are the barycentric coordinates of $\mathbf{x}_i$. Obviously, $\mathbf{x}_i$ is a linear transformation of $\mathbf{v}$, which can be written as

$$\mathbf{x}_i = \mathbf{T}_i \mathbf{v} \tag{2}$$

where $\mathbf{T}_i$ is a $3 \times 3N$ matrix in the form:

$$\mathbf{T}_i = \begin{bmatrix} \cdots & a_i & 0 & 0 & \cdots & b_i & 0 & 0 & \cdots & c_i & 0 & 0 & \cdots \\ \cdots & 0 & a_i & 0 & \cdots & 0 & b_i & 0 & \cdots & 0 & c_i & 0 & \cdots \\ \cdots & 0 & 0 & a_i & \cdots & 0 & 0 & b_i & \cdots & 0 & 0 & c_i & \cdots \end{bmatrix}$$

The triangulated mesh model can therefore be parameterized as $\mathbf{M} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_S\}$, where S is the number of keypoints on the model.

Suppose the model $\mathbf{M}$ for the surface and the projection matrix $\mathbf{P}$ of the camera are known. The problem of deformable surface 3D tracking is formulated as: given model $\mathbf{M}$ and camera $\mathbf{P}$, how can we reconstruct the structure $\mathbf{v}$ for each frame?

4. Deformable surface 3D tracking using sequential second order cone programming

Recently, there has been interest in solving geometric vision problems such as triangulation and camera resectioning using L_∞ minimization [8,9]. 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. In our approach, we follow the same idea and introduce two sets of constraints. The first set of constraints are model-to-image keypoint correspondences which can be formulated as SOCP constraints. The second set of constraints are designed to stop the edges of the mesh from expanding or shrinking which are typically non-convex constraints. In order to take advantage of the efficient SOCP solver, we introduce an approach to deal with these non-convex constraints in a sequential manner. In this section, we first briefly describe the convex optimization and SOCP. Then we formulate the deformable surface 3D tracking problem. Finally we introduce how to solve this problem using sequential SOCP.

4.1. Convex optimization

There are great advantages to recognizing or formulating a problem as a convex optimization problem. The most basic advantage is that the problem can then be solved very reliably and efficiently [3]. A convex optimization problem is one of the form:

$$\begin{aligned} &\text{minimize} && f_0(\mathbf{x}) \\ &\text{subject to} && f_i(\mathbf{x}) \le 0, \quad i = 1, \ldots, m \\ &&& \mathbf{a}_j^T \mathbf{x} = \mathbf{b}_j, \quad j = 1, \ldots, n \end{aligned} \tag{3}$$

Here $\mathbf{x} \in \mathbf{R}^n$ is the optimization variable and both the objective function $f_0 : \mathbf{R}^n \to \mathbf{R}$ and the constraint functions $f_i : \mathbf{R}^n \to \mathbf{R}$ are convex functions.

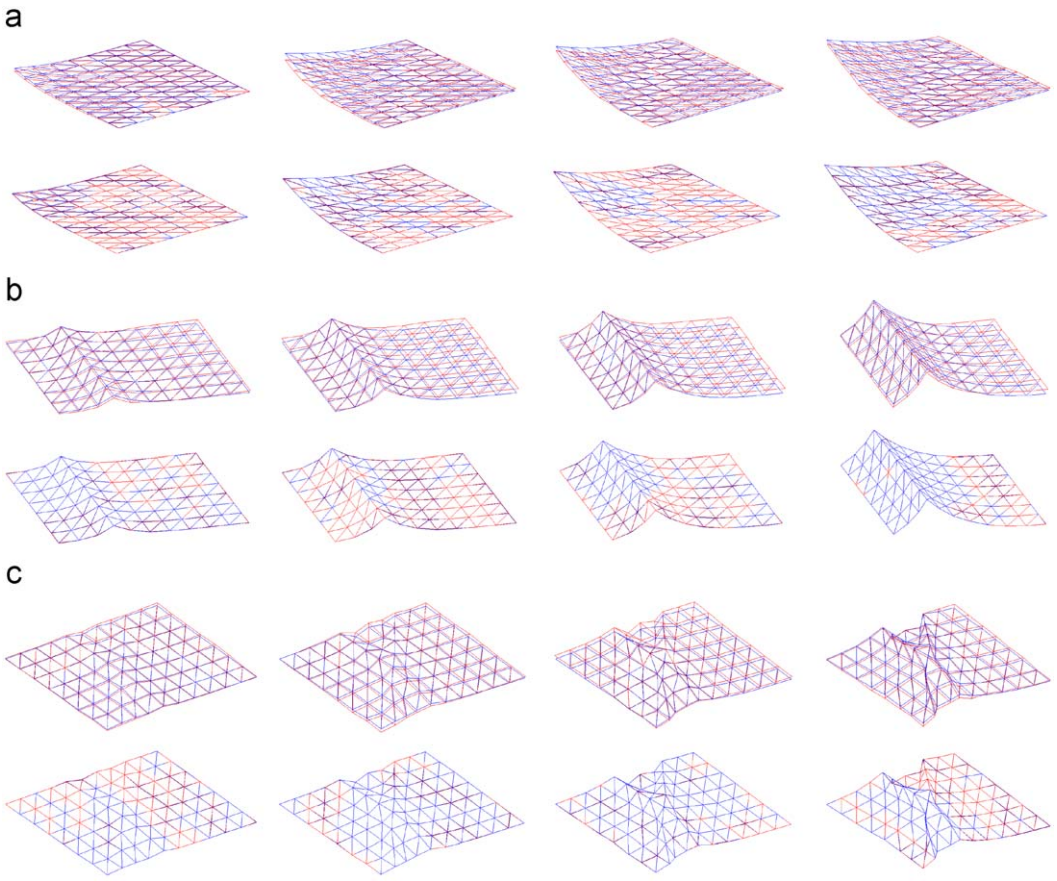

Fig. 3. Some tracking results of three synthetic image sequences including smooth, sharp and complex deformations. In (a), (b) and (c), results in the first row are obtained by the method of [17], and results in the second row are obtained by the sequential SOCP. In all the results, the reconstructed mesh is shown in red, and the ground-truth is shown in blue. (A video of the results is submitted as supplementary material.) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The convex optimization problem is called a second order cone programming (SOCP) if the objective function is affine and the inequality constraints are second order cone constraints, as:

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}^T\mathbf{x} \\ \text{subject to} \quad & \|\mathbf{A}_i\mathbf{x}+\mathbf{b}_i\| \le \mathbf{c}_i^T\mathbf{x}+d_i, \quad i=1,\dots,m \\ & \mathbf{F}\mathbf{x}=\mathbf{g} \end{aligned} \tag{4}$$

If there is no objective function to minimize in Eq. (4), it becomes an SOCP *feasibility problem*, where one seeks for a point within a feasible region, as

$$\begin{aligned} \text{find} \quad & \mathbf{x} \\ \text{subject to} \quad & \|\mathbf{A}_i\mathbf{x}+\mathbf{b}_i\| \le \mathbf{c}_i^T\mathbf{x}+d_i, \quad i=1,\dots,m \\ & \mathbf{F}\mathbf{x}=\mathbf{g} \end{aligned} \tag{5}$$

The SOCP includes linear programming as a special case, but it is less general than SemiDefinite programming. SOCP problems can be effectively solved using interior-point methods [3].

To formulate the tracking problem as an SOCP problem, two sets of constraints are introduced. One set is used to represent model-to-image keypoint correspondences and another set is used to prevent the edges of the mesh from expanding or shrinking, which will be detailed in next sections.

4.2. Constraints for keypoint correspondences

At each time instance in the process of tracking, the model to the image keypoint correspondences are established. As has already been pointed out in [17], these correspondences can be formulated as SOCP constraints. We assume that a keypoint in the model is $\mathbf{x}_i$, and its corresponding image point location is $(\hat{u}_i,\hat{v}_i)^T$. Given the camera projection matrix $\mathbf{P}$, the projection of $\mathbf{x}_i$ is

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}$$

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

$$\begin{aligned} &\left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \end{bmatrix} \right\| \\ &= \frac{\left\| \left(\mathbf{P}_{1:2,1:3} - \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \end{bmatrix} \mathbf{P}_{3,1:3} \right) \mathbf{x}_i + \left(\mathbf{P}_{1:2,4} - \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \end{bmatrix} \mathbf{P}_{3,4} \right) \right\|}{\mathbf{P}_{3,1:3}\mathbf{x}_i + \mathbf{P}_{3,4}} \end{aligned} \tag{6}$$

where $\mathbf{P}_{1:2,1:3}$ is a submatrix of $\mathbf{P}$ formed by rows 1,2 and columns 1,2,3. $\mathbf{P}_{3,1:3}$ is a submatrix of $\mathbf{P}$ formed by row 3 and columns 1,2,3. $\mathbf{P}_{1:2,4}$ is a submatrix of $\mathbf{P}$ formed by rows 1,2 and column 4. And $\mathbf{P}_{3,4}$ is an element of $\mathbf{P}$ in row 3 and column 4.

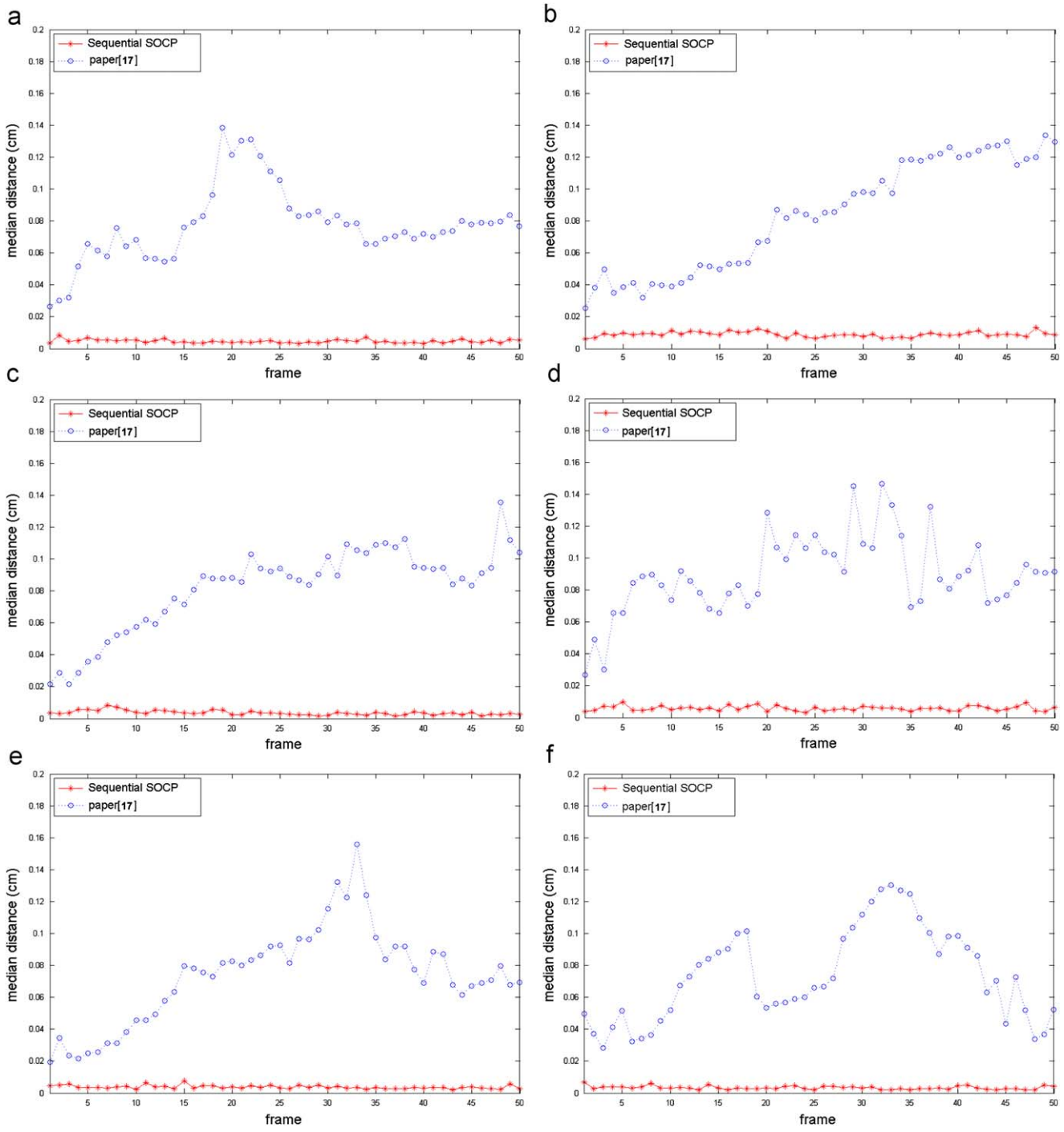

Fig. 4. Median distances between reconstructed mesh vertices and ground-truth of three synthetic image sequences when adding Gaussian noise with mean zero and variance one and two. (a), (c) and (e) are the results of three sequences with variance one noise, $\sigma=1$. (b), (d) and (f) are the results of three sequences with variance two noise, $\sigma=2$. In all the graphs, results obtained by the method of [17] are shown in blue, and results obtained by the sequential SOCP are shown in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Due to image noise, Eq. (6) cannot be zero, and a variable γ is used as its upper bound. If γ is considered to be known, the tracking problem can be formulated as

$$\begin{aligned} &\text{find} \quad \mathbf{v} \\ &\text{subject to} \quad \left\| \left(\mathbf{P}_{1:2,1:3} - \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \end{bmatrix} \mathbf{P}_{3,1:3} \right) \mathbf{x}_i + \left(\mathbf{P}_{1:2,4} - \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \end{bmatrix} \mathbf{P}_{3,4} \right) \right\| \\ &\qquad \leq \gamma \mathbf{P}_{3,1:3} \mathbf{x}_i + \gamma \mathbf{P}_{3,4}, \quad i = 1, \ldots, m \end{aligned} \tag{7}$$

where m is the number of model-to-image keypoint correspondences. Since $\mathbf{x}_i$ is a linear transformation of $\mathbf{v}$, $\mathbf{x}_i = \mathbf{T}_i \mathbf{v}$, as defined in Eq. (2), we denote

$$\begin{aligned} &\mathbf{A}_i = \left(\mathbf{P}_{1:2,1:3} - \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \end{bmatrix} \mathbf{P}_{3,1:3} \right) \mathbf{T}_i, \quad \mathbf{b}_i = \mathbf{P}_{1:2,4} - \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \end{bmatrix} \mathbf{P}_{3,4}, \\ &\mathbf{c}_i^T = \mathbf{P}_{3,1:3} \mathbf{T}_i, \quad d_i = \mathbf{P}_{3,4} \end{aligned} \tag{8}$$

Table 1
Computational speed of different methods.

Image sequence	Paper [17] (s)	Sequential SOCP (s)
Sequence 1	237.2	244.8
Sequence 2	235.9	219.1
Sequence 3	255.1	250.8

Each value in this table is the computation time to process a whole 50 frames sequence when adding mean zero and variance two Gaussian noise.

Then Eq. (7) can be expressed as

$$\begin{aligned} &\text{find} \quad \mathbf{v} \\ &\text{subject to} \quad \|\mathbf{A}_i\mathbf{v}+\mathbf{b}_i\| \le \gamma \mathbf{c}_i^T\mathbf{v}+\gamma d_i, \quad i=1,\dots,m \end{aligned} \tag{9}$$

where m is the number of model-to-image keypoint correspondences. Eq. (9) defines an SOCP feasibility problem. Without other constraints, the minimal γ in Eq. (9) can be found using *bisection* algorithm [8,17]. 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.

4.3. Constraints for the mesh shape

In [17], a set of constraints was introduced to prevent the orientation of the mesh edges from changing irrationally between two consecutive frames. Assume we know the vertices linking an edge at time t are $\boldsymbol{v}_p^t$ and $\boldsymbol{v}_q^t$, the constraints for vertices at time $t+1$, $\boldsymbol{v}_p^{t+1}$ and $\boldsymbol{v}_q^{t+1}$, are stated as

$$\left\| \boldsymbol{v}_p^{t+1}-\boldsymbol{v}_q^{t+1}-L_{p,q}\frac{\boldsymbol{v}_p^t-\boldsymbol{v}_q^t}{\|\boldsymbol{v}_p^t-\boldsymbol{v}_q^t\|} \right\| \le 0.1L_{p,q} \quad \langle p,q\rangle \in C \tag{10}$$

where $L_{p,q}$ is the original length of the edge linking vertices $\boldsymbol{v}_p$ and $\boldsymbol{v}_q$, and $C=\{\langle p,q\rangle | \boldsymbol{v}_p \text{ and } \boldsymbol{v}_q \text{ are neighboring vertices of the mesh}\}$. Since $\boldsymbol{v}_p^{t+1}$ and $\boldsymbol{v}_q^{t+1}$ are all linear transformations of $\mathbf{v}$, the constraints of Eq. (10) are also SOCP constraints that can be involved in the SOCP problem defined in Eq. (9).

In the process of tracking, [17] solves the SOCP feasibility problem at each time instance, and the remaining scale ambiguity is handled by rescaling the area of the mesh to its initial size. The advantage of this method is that all the constraints can be perfectly incorporated into the SOCP framework. However, the constraints introduced in Eq. (10) cannot reflect the true behavior of the surface. Theoretically, for an inextensible surface, the constraints should be designed to stop the edges of the mesh from expanding or shrinking, as:

$$\|\boldsymbol{v}_p-\boldsymbol{v}_q\| = L_{p,q} \quad \langle p,q\rangle \in C \tag{11}$$

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

$$\boldsymbol{v}_p-\boldsymbol{v}_q=\mathbf{E}_j\mathbf{v}, \quad l_j=L_{p,q} \tag{12}$$

where $\mathbf{E}_j$ is a $3\times 3N$ matrix in the form:

$$\mathbf{E}_j=\begin{bmatrix} \cdots & 1 & 0 & 0 & \cdots & -1 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & 0 & \cdots & 0 & -1 & 0 & \cdots \\ \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & -1 & \cdots \end{bmatrix}$$

Then the constraints in Eq. (11) can be expressed as

$$\|\mathbf{E}_j\mathbf{v}\| = l_j, \quad j=1,\dots,n \tag{13}$$

where n is the number of edges of the mesh. The constraints in Eq. (13) are more generic than Eq. (10), but they are typically non-convex terms and cannot be involved in the SOCP problem defined in Eq. (9) directly. To this end, we introduce an approach to deal with the non-convex terms in Eq. (13) using SOCP solver in next section.

4.4. Solving the tracking problem using sequential SOCP

Now we have two sets of constraints, one for model-to-image keypoint correspondences as defined in Eq. (9), another for retaining the original lengths of mesh edges as defined in Eq. (13). The deformable surface 3D tracking problem can be written as

$$\begin{aligned} &\text{find} \quad \mathbf{v} \\ &\text{subject to} \quad \|\mathbf{A}_i\mathbf{v}+\mathbf{b}_i\| \le \gamma \mathbf{c}_i^T\mathbf{v}+\gamma d_i, \quad i=1,\dots,m \qquad (14a) \\ &\qquad\qquad \|\mathbf{E}_j\mathbf{v}\| = l_j, \quad j=1,\dots,n \qquad (14b) \end{aligned}$$

where m is the number of model-to-image keypoint correspondences, and n is the number of edges of the mesh. $\mathbf{A}_i$, $\mathbf{b}_i$, $\mathbf{c}_i$ and d_i are defined in Eq. (8), $\mathbf{E}_j$ and l_j are defined in Eq. (12). Since constraints in Eq. (14b) are non-convex constraints, this problem is a non-convex feasibility problem. Obviously, the equality constraints in Eq. (14b) can reasonably be replaced by two sets of inequality constraints, as

$$\|\mathbf{E}_j\mathbf{v}\| \le (1+\varepsilon)l_j, \quad j=1,\dots,n \tag{15a}$$

$$\|\mathbf{E}_j\mathbf{v}\| \ge (1-\varepsilon)l_j, \quad j=1,\dots,n \tag{15b}$$

where ε can be set to a very small value (throughout the paper we set $\varepsilon=0.001$). Now the deformable surface 3D tracking problem become

$$\begin{aligned} &\text{find} \quad \mathbf{v} \\ &\text{subject to} \quad \|\mathbf{A}_i\mathbf{v}+\mathbf{b}_i\| \le \gamma \mathbf{c}_i^T\mathbf{v}+\gamma d_i, \quad i=1,\dots,m \qquad (16a) \\ &\qquad\qquad \|\mathbf{E}_j\mathbf{v}\| \le (1+\varepsilon)l_j, \quad j=1,\dots,n \qquad (16b) \\ &\qquad\qquad \|\mathbf{E}_j\mathbf{v}\| \ge (1-\varepsilon)l_j, \quad j=1,\dots,n \qquad (16c) \end{aligned}$$

In fact, replacing the equality constraints with these inequality constraints will not change the nature of the problem. That is, the problem after such modifications remains to be non-convex, because the inequalities in Eq. (16c) are non-convex constraints.

Because Eq. (16) is a non-convex feasibility problem, convex programming methods are not directly applicable. Below we will describe a solution method for the problem that allows one to use efficient SOCP solver to solve Eq. (16) in a sequential manner.

Suppose we start with an initial point $\mathbf{v}_0$ and seeks 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 three sets of constraints in Eq. (16) in this case become

$$\|\mathbf{A}_i(\mathbf{v}_k+\delta_k)+\mathbf{b}_i\| \le \gamma \mathbf{c}_i^T(\mathbf{v}_k+\delta_k)+\gamma d_i, \quad i=1,\dots,m \tag{17a}$$

$$\|\mathbf{E}_j(\mathbf{v}_k+\delta_k)\| \le (1+\varepsilon)l_j, \quad j=1,\dots,n \tag{17b}$$

$$\|\mathbf{E}_j(\mathbf{v}_k+\delta_k)\| \ge (1-\varepsilon)l_j, \quad j=1,\dots,n \tag{17c}$$

Constraints in Eq. (17c) can be expressed as

$$2\mathbf{v}_k^T\mathbf{E}_j^T\mathbf{E}_j\delta_k+\delta_k^T\mathbf{E}_j^T\mathbf{E}_j\delta_k \ge (1-\varepsilon)^2l_j^2-\mathbf{v}_k^T\mathbf{E}_j^T\mathbf{E}_j\mathbf{v}_k, \quad j=1,\dots,n \tag{18}$$

Since $\mathbf{E}_j^T\mathbf{E}_j$ is a positive definite matrix, $\delta_k^T\mathbf{E}_j^T\mathbf{E}_j\delta_k > 0$. Now if we remove the second term on the left-hand side of Eq. (18), we have

$$2\mathbf{v}_k^T\mathbf{E}_j^T\mathbf{E}_j\delta_k \geq (1-\varepsilon)^2 l_j^2 - \mathbf{v}_k^T\mathbf{E}_j^T\mathbf{E}_j\mathbf{v}_k, \quad j=1,\ldots,n \tag{19}$$

Obviously, Eq. (18) remains valid as long as Eq. (19) holds. The n linear inequality constraints in Eq. (19) can be put together as

$$\mathbf{F}_k\delta_k \geq \mathbf{g}_k \tag{20}$$

where

$$\mathbf{F}_k = \begin{bmatrix} 2\mathbf{v}_k^T\mathbf{E}_1^T\mathbf{E}_1 \\ 2\mathbf{v}_k^T\mathbf{E}_2^T\mathbf{E}_2 \\ \vdots \\ 2\mathbf{v}_k^T\mathbf{E}_n^T\mathbf{E}_n \end{bmatrix}, \quad \mathbf{g}_k = \begin{bmatrix} (1-\varepsilon)^2 l_1^2 - \mathbf{v}_k^T\mathbf{E}_1^T\mathbf{E}_1\mathbf{v}_k \\ (1-\varepsilon)^2 l_2^2 - \mathbf{v}_k^T\mathbf{E}_2^T\mathbf{E}_2\mathbf{v}_k \\ \vdots \\ (1-\varepsilon)^2 l_n^2 - \mathbf{v}_k^T\mathbf{E}_n^T\mathbf{E}_n\mathbf{v}_k \end{bmatrix}$$

In summary, the above analysis suggests that, in the k-th iteration, the point $\mathbf{v}_k$ can be updated to point $\mathbf{v}_{k+1} = \mathbf{v}_k + \delta_k$ where δ_k is the solution of the following SOCP feasibility problem:

$$\begin{aligned} \text{find} \quad & \delta_k \\ \text{subject to} \quad & \|\mathbf{A}_i(\mathbf{v}_k+\delta_k)+\mathbf{b}_i\| \leq \gamma\mathbf{c}_i^T(\mathbf{v}_k+\delta_k)+\gamma d_i, \quad i=1,\ldots,m \\ & \|\mathbf{E}_j(\mathbf{v}_k+\delta_k)\| \leq (1+\varepsilon)l_j, \quad j=1,\ldots,n \\ & \mathbf{F}_k\delta_k \geq \mathbf{g}_k \end{aligned} \tag{21}$$

Since Eq. (21) is an SOCP problem, now we can take advantage of the efficient SOCP solver. How to identify a good initial point $\mathbf{v}_0$ is very important when we solving Eq. (21). In this paper, at each time instance t, the tracking result for previous time instance $\mathbf{v}^{t-1}$ is a good choice for the initial point $\mathbf{v}_0$, because the change between two consecutive frames is always not violent.

Given the initial point $\mathbf{v}_0$ and an initial upper bound of reprojection errors γ, we can solve Eq. (21). If it is *feasible*, a solution δ_0 can be calculated, otherwise, we continuously increase γ until Eq. (21) becomes *feasible*. Once a solution δ_0 of Eq. (21) is obtained, we set $\mathbf{v}_1 = \mathbf{v}_0 + \delta_0$, and then $\mathbf{v}_k$ can be updated iteratively. In the k-th iteration, once a solution δ_k of Eq. (21) is calculated, we set $\mathbf{v}_{k+1} = \mathbf{v}_k + \delta_k$, let $k := k+1$, and then use the updated data to obtain a new δ_k. Theoretically, $\mathbf{v}_k$ will converge to a solution $\mathbf{v}^*$ as $\|\delta_k\| \to 0$. In fact, $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k, \ldots, \mathbf{v}^*\}$ are all solutions of the tracking problem defined in Eq. (16), because if δ_k is a solution of Eq. (21), $\mathbf{v}_k + \delta_k$ must satisfies three sets of constraints in Eq. (17), i.e. $\mathbf{v}_{k+1}$ is a solution of Eq. (16). In practice, we justtake $\mathbf{v}_1$ as the solution, that means we do not iteratively update $\mathbf{v}_k$ until it converges, but accept $\mathbf{v}_1$ once we get δ_0. The procedure can be summarized as Algorithm 1.

Using Algorithm 1 we could find a solution of Eq. (16) with a feasible upper bound γ_{feasible}. Now the last question is how can we find the minimal γ?

Once we get $\mathbf{v}_1$ and γ_{feasible}, the minimal γ could be found by iteratively solving the SOCP feasibility problem of Eq. (21) and decreasing γ simultaneously, i.e. set $\gamma = \gamma - \gamma_{\text{step}}$ at each iteration. In practice, at each iteration, we always set $\gamma_{\text{step}} = \gamma/2$ and solve Eq. (21). If the problem is *infeasible*, we bisect γ_{step} and resolve Eq. (21). The iteration continues until γ_{step} is less than a prescribed threshold η (throughout the paper we set $\eta = 0.05$). We should note that, this procedure is different from the famous *bisection* algorithm because the SOCP problem in Eq. (21) depends not only on γ but also on $\mathbf{v}_k$. The procedure of finding the minimal γ can be summarized as Algorithm 2.

Algorithm 1.

A. (`Input`)
I. The initial point $\mathbf{v}_0$ which is the tracking result for previous time instance;
II. The initial upper bound γ_{initial} (throughout the paper we set $\gamma_{\text{initial}} = 2$ pixels);

B. (`Solve the SOCP feasibility problem`)
1: $\gamma \leftarrow \gamma_{\text{initial}}$;
2: Solve the SOCP feasibility problem in Eq. (21) to obtain δ_0;
3: **if** *feasible* **then**
4: Set $\mathbf{v}_1 = \mathbf{v}_0 + \delta_0$, and set $\gamma_{\text{feasible}} = \gamma$;
5: Go to step 10;
6: **else**
7: $\gamma \leftarrow \gamma \times 2$;
8: Go to step 2;
9: **end if**
10: Output the solution $\mathbf{v}_1$ and the upper bound γ_{feasible}.

Algorithm 2.

A. (`Input`)
A solution $\mathbf{v}_1$ and an upper bound γ_{feasible} obtained by Algorithm 1;

B. (`Solve the sequential SOCP feasibility problem`)
1: $k \leftarrow 1$;
2: $\gamma_k \leftarrow \gamma_{\text{feasible}}$, $\gamma_{\text{step}} \leftarrow \gamma_k/2$;
3: Set $\gamma = \gamma_k - \gamma_{\text{step}}$;
4: Solve the SOCP feasibility problem in Eq. (21) to obtain δ_k;
5: **if** *feasible* **then**
6: Set $\mathbf{v}_{k+1} = \mathbf{v}_k + \delta_k$, and set $\gamma_{k+1} = \gamma$, $\gamma_{\text{step}} = \gamma_{k+1}/2$;
7: $k \leftarrow k+1$;
8: Go to step 3;
9: **else**
10: set $\gamma_{\text{step}} = \gamma_{\text{step}}/2$;
11: If $\gamma_{\text{step}} < \eta$, go to step 14;
12: Go to step 3;
13: **end if**
14: Output the solution $\mathbf{v}_k$ and the minimal upper bound of reprojection errors γ.

One drawback of using SOCP is that it is not robust to outliers, and wrong correspondences happen occasionally. It has been proved that the set of keypoint correspondences whose reprojection error equals the minimal γ contain at least one outlier [20]. Thus, if keep throwing out these keypoint correspondences, we will eventually remove all outliers in the data. In our implementation, we keep removing outliers and redoing Algorithms 1 and 2 until the minimal γ is less than 2 pixels.

5. Performance evaluation

The performance of our approach was evaluated with both synthetic and real image sequences. All the experiments are implemented under the Matlab environment using SeDuMi which is a toolbox for optimizing over convex cones [21].

5.1. Quantitative evaluation on synthetic image sequences

Our approach was quantitatively evaluated on synthetic image sequences. We synthetically deformed a 8×11 vertex mesh and generate three synthetic image sequences with smooth, sharp and complex deformations respectively. The size of the mesh is 8 cm × 11 cm. We apply forces to certain vertices of the mesh and keep mesh edges to be their original lengths. Each sequence contains 50 frames. We

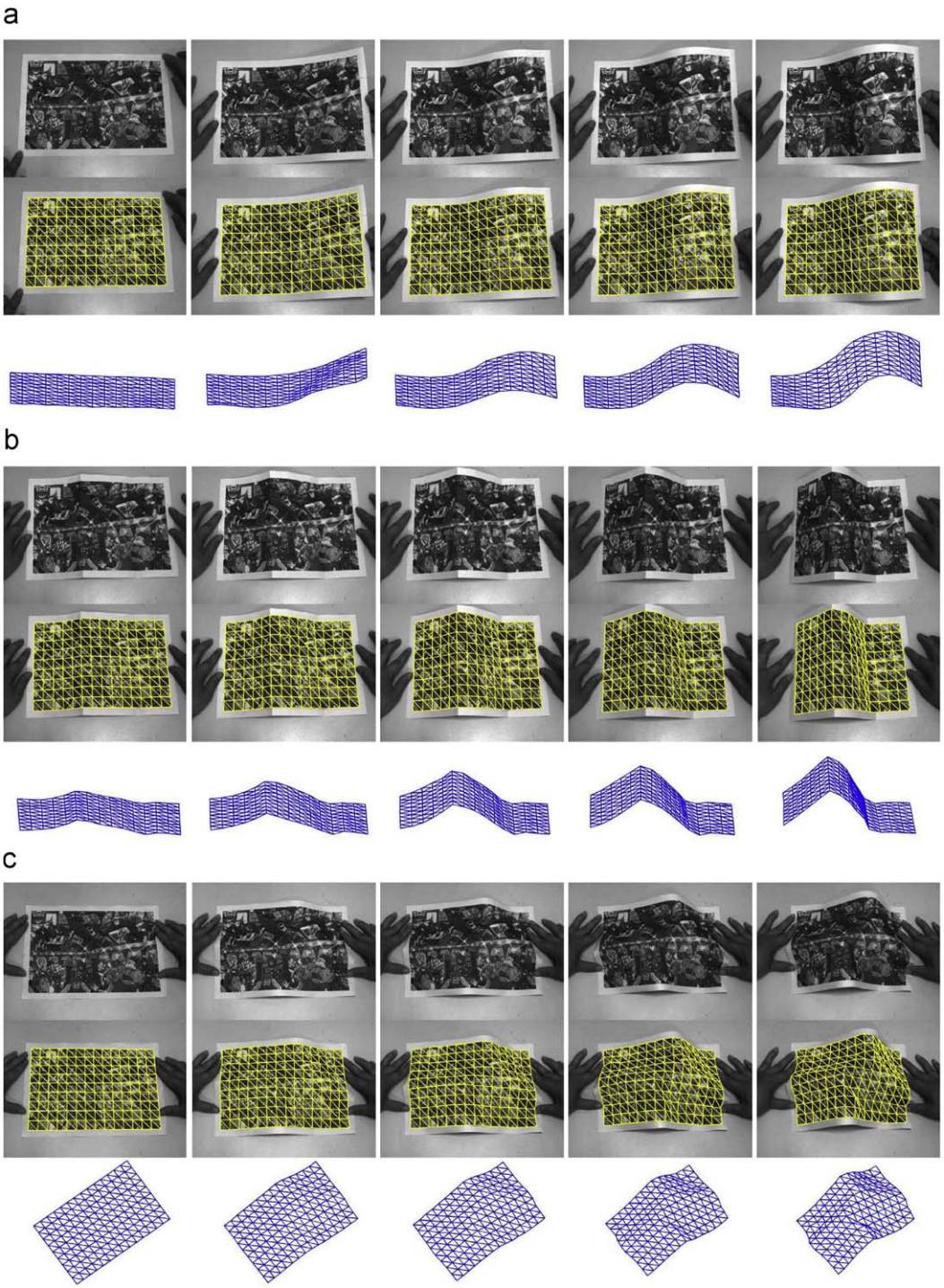

Fig. 5. Tracking a piece of paper with cartoon texture in three real image sequences. In (a), (b) and (c), the top row are original images, the middle row are images with reprojected meshes and the bottom row are reconstructed triangulated meshes seen from a different view. (A video of the results is submitted as supplementary material.) (a) smooth deformation; (b) sharp folds and (c) complex deformation.

randomly chose four 3D points in each facet and projected these points to the image plane using a perspective projection matrix, which gave us a set of point correspondences. To evaluate the robustness of our approach, we add Gaussian noise with mean zero and variance one and two to the image point locations.

We compared our approach with the method in paper [17] for these three synthetic image sequences. Fig. 3 shows some tracking results of these image sequences when adding variance two Gaussian noise. The results show that although the method in [17] generated very good results in which the differences between the reconstructed mesh and the ground-truth are very small, our approach obtained more accurate results in which the differences are almost indistinguishable.

Fig. 4 shows the median distances between reconstructed mesh vertices and ground-truth when adding Gaussian noise with variances one and two. The results show that the median distances are

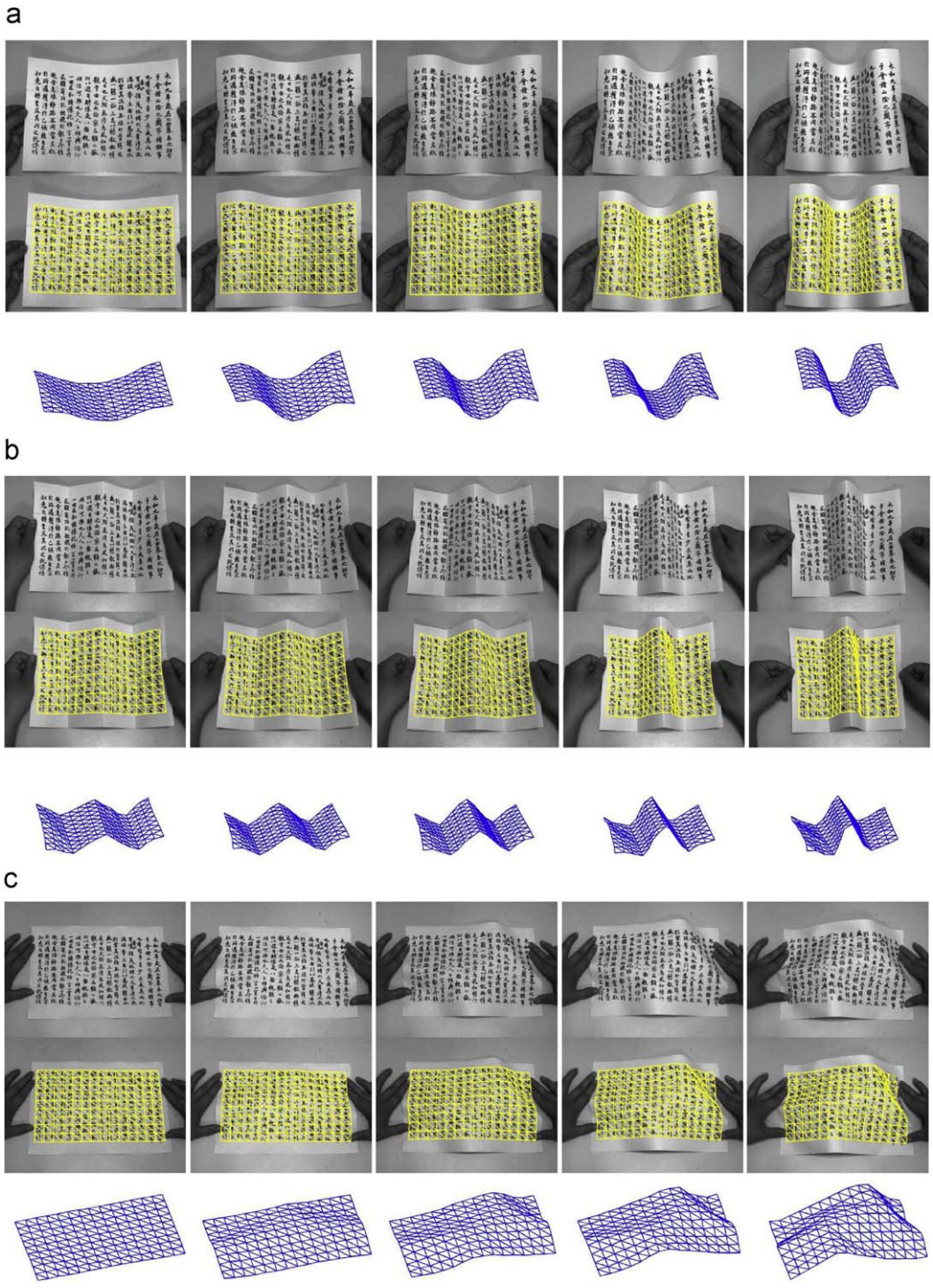

Fig. 6. Tracking a piece of paper with Chinese calligraphy texture in three real image sequences. In (a), (b) and (c), the top row are original images, the middle row are images with reprojected meshes and the bottom row are reconstructed triangulated meshes seen from a different view. (A video of the results is submitted as supplementary material.) (a) smooth deformation; (b) sharp folds and (c) complex deformation.

of the order of 0.15 cm using the method in [17], and are of the order of 0.01 cm using our sequential SOCP approach. The significant improvement of the reconstruction accuracy demonstrates that the shape constraints introduced in our approach truly reflect the physical properties of the surface.

The computational speed of different methods is presented in Table 1. Compare with [17], our approach has n more linear inequality constraints as shown in Eq. (20), but still has a high computational speed owing to the efficient SOCP solver. In sequences 2 and 3 our method runs even a little faster than [17], that is because our approach needs fewer iterations to find the minimal γ in some frames.

5.2. Qualitative evaluation on real image sequences

Finally we evaluated our approach qualitatively on real image sequences. Two pieces of paper, one with cartoon texture and another with Chinese calligraphy texture, were used. We generated two sets of image sequences for these two pieces of paper. Each set has three

image sequences including the paper with smooth, sharp and complex deformations, as shown in Figs. 5 and 6. These image sequences were captured by a digital cameral with a resolution of 1024×768. The camera was calibrated beforehand and the camera projection matrix, **P**, remained constant during the process. The size of the triangulated mesh is set to 10×15.

We use following steps to create the triangulated mesh model. When the piece of paper is flat, it can be seen as a plane and we can get 2D coordinates of four corner points of this piece of paper in this plane. Then the 10×15 triangulated mesh can be formed from these four corner points. Capture an image of this piece of paper without deformations, we can get four image locations corresponding to four corner points of this piece of paper. Then the 2D homography matrix H between the paper plane and the image plane could be calculated using these four 2D to 2D point correspondences. After that, we use SIFT [13] to extract keypoints from the image, and transform locations of these keypoints to the paper plane using H. Now we have locations of both the mesh and the keypoints in the paper plane, and we can calculate the barycentric coordinates of each keypoint and obtain the model $\mathbf{M} = \{T_1, T_2, \ldots, T_S\}$, where S is the number of keypoints on the model.

Given the model **M**, the camera projection matrix **P** and the structure of the surface in the first frame, sequential SOCP was used to reconstruct the surface structures in successive frames. In the process of tracking, keypoint correspondences between the model and the image were established using SIFT at each time instance. Some tracking results of the surface with cartoon texture are shown in Fig. 5, and results of the surface with Chinese calligraphy texture are shown in Fig. 6. The results show that our approach can correctly recover shapes of the surface with smooth, sharp and other complex deformations.

6. Conclusions

This paper proposes a method for tracking deformable surface in 3D from monocular image sequences. In our approach, two sets of constraints are introduced. The first set of constraints are model-to-image keypoint correspondences which can be formulated as SOCP constraints. The second set of constraints are designed to stop the edges of the mesh from expanding or shrinking which are typically non-convex constraints. In order to take advantage of the efficient SOCP solver, the deformable surface tracking problem is solved by solving sequential SOCP feasibility problems in which the non-convex constraints are replaced by a set of convex constraints over a local convex region. Since the shape constraints used in our approach is more generic than previous methods, the proposed method enables us to handle highly deformable surfaces. Experiments on synthetic and real image sequences demonstrate the capability and efficiency of our approach.

Currently we take the tracking result of previous frame as the initial structure of current frame which is the main limitation of our approach, because it may fail to recover the shape correctly when something goes wrong in previous frame. In future work, we will investigate how to obtain an appropriate initial structure without using information of previous frames. Besides, our future work also includes investigating how to extend our method to handle extensible deformable surfaces.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.patcog.2009.06.016.

References

[1] T.C.S. Azevedo, J.M.R.S. Tavares, M.A.P. Vaz, 3d object reconstruction from uncalibrated images using an off-the-shelf camera, in: Advances in Computational Vision and Medical Image Processing, Springer, Berlin, 2008, pp. 117–136.

[2] T.C.S. Azevedo, J.M.R.S. Tavares, M.A.P. Vaz, External anatomical shapes reconstruction from turntable image sequences using a single off-the-shelf camera, Electronic Letters on Computer Vision and Image Analysis 7 (2) (2008) 22–34.

[3] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, 2004.

[4] T.F. Cootes, G.J. Edwards, C.J. Taylor, Active appearance models, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 681–685.

[5] D. Forsyth, O. Arikan, L. Ikemoto, J. Brien, D. Ramanan, Computational studies of human motion, tracking and motion synthesis: part 1, Foundations and Trends in Computer Graphics and Vision 1 (2) (2006) 77–254.

[6] M. Han, T. Kanade, Multiple motion scene reconstruction from uncalibrated views, in: Proceedings of the IEEE International Conference on Computer Vision, Vancouver, Canada, 2001.

[7] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge University Press, Cambridge, 2004.

[8] F. Kahl, Multiple view geometry and the L_∞ norm, in: Proceedings of the IEEE International Conference on Computer Vision, Beijing, China, 2005.

[9] Q. Ke, T. Kanade, Quasiconvex optimization for robust geometric reconstruction, in: Proceedings of the IEEE International Conference on Computer Vision, Beijing, China, 2005.

[10] A. Laurentini, The visual hull concept for silhouette-based image understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (2) (1994) 150–162.

[11] V. Lepetit, P. Fua, Monocular model-based 3d tracking of rigid objects: a survey, Foundations and Trends in Computer Graphics and Vision 1 (1) (2005) 1–89.

[12] V. Lepetit, P. Fua, Keypoint recognition using randomized trees, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (9) (2006) 1465–1479.

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

[14] I. Matthews, S. Baker, Active appearance models revisited, International Journal of Computer Vision 60 (2) (2004) 135–164.

[15] T. Mclnerney, D. Terzopoulos, A finite element model for 3d shape reconstruction and nonrigid motion tracking, in: Proceedings of the International Conference on Computer Vision, Berlin, Germany, 1993.

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

[17] M. Salzmann, R. Hartley, P. Fua, Convex optimization for deformable surface 3-d tracking, in: Proceedings of IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007.

[18] M. Salzmann, R. Urtasun, P. Fua, Local deformation models for monocular 3d shape recovery, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA, 2008.

[19] A. Sheffer, E. Praun, K. Rose, Mesh parameterization methods and their applications, Foundations and Trends in Computer Graphics and Vision 2 (2) (2006) 105–171.

[20] K. Sim, R. Hartley, Removing outliers using the L_∞ norm, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006.

[21] J. Sturm, Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, Optimization Methods and Software 11–12 (1999) 625–653.

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

[23] R. Vidal, Y. Ma, S. Soatto, S. Sastry, Two-view multibody structure from motion, International Journal of Computer Vision 68 (1) (2006) 7–25.

[24] J. Xiao, T. Kanade, Uncalibrated perspective reconstruction of deformable structures, in: Proceedings of the IEEE International Conference on Computer Vision, Beijing, China, 2005.

About the Author—SHUHAN SHEN received the B.S. and M.S. degrees in Control Science and Engineering from Southwest Jiao Tong University, China, in 2003 and 2006, respectively. Now he is a Ph.D. student in the Institute of Pattern Recognition and Image Processing at Shanghai Jiao Tong University, China. His current research interests include computer vision, pattern recognition and robotics.

About the Author—YUNCAI LIU received the Ph.D. degree from the University of Illinois at Urbana-Champaign, in the Department of Electrical and Computer Science Engineering, in 1990, and worked as an Associate Researcher at the Beckman Institute of Science and Technology from 1990 to 1991. Since 1991, he had been a System Consultant and then a Chief Consultant of research in Sumitomo Electric Industries, Ltd., Japan. In October 2000, he joined the Shanghai Jiao Tong University as a distinguished Professor. His research interests are in image processing and computer vision, especially in motion estimation, feature detection and matching, and image registration. He also made many progresses in the research of intelligent transportation systems.

About the Author—WU-SHENG LU received the B.S. degree in Mathematics from Fudan University, Shanghai, China, in 1964, and the M.S. degree in Electrical Engineering and the Ph.D. degree in Control Science from the University of Minnesota, Minneapolis, USA, in 1983 and 1984, respectively. He was a Postdoctoral Fellow at the University of Victoria, Victoria, BC, Canada, in 1985 and a visiting Assistant Professor with the University of Minnesota in 1986. Since 1987, he has been with the University of Victoria where he is a Professor. His current teaching and research interests are in the general areas of digital signal processing and application of optimization methods. He is the co-author with A. Antoniou of *Two-Dimensional Digital Filters* (Marcel Dekker, 1992) and *Practical Optimization: Algorithms and Engineering Applications* (Springer, 2007). He served as an Associate Editor of the Canadian Journal of Electrical and Computer Engineering in 1989, and Editor of the same journal from 1990 to 1992. He served as an Associate Editor for the IEEE Transactions on Circuits and Systems, Part II, from 1993 to 1995 and for Part I of the same journal from 1999 to 2001 and from 2004 to 2005. Presently he is serving as an Associate Editor for the Journal of Circuits, Systems, and Signal Processing. He is a Fellow of the Engineering Institute of Canada and the IEEE.