An Effective Approach to 3D Deformable Surface Tracking

Jianke Zhu[1], Steven C.H. Hoi[2], Zenglin Xu[1], and Michael R. Lyu[1]

[1] Department of Computer Science and Engineering,
Chinese University of Hong Kong, Shatin, Hong Kong
{jkzhu,zlxu,lyu}@cse.cuhk.edu.hk
[2] School of Computer Engineering,
Nanyang Technological University, Singapore
chhoi@ntu.edu.sg

Abstract. The key challenge with 3D deformable surface tracking arises from the difficulty in estimating a large number of 3D shape parameters from noisy observations. A recent state-of-the-art approach attacks this problem by formulating it as a Second Order Cone Programming (SOCP) feasibility problem. The main drawback of this solution is the high computational cost. In this paper, we first reformulate the problem into an unconstrained quadratic optimization problem. Instead of handling a large set of complicated SOCP constraints, our new formulation can be solved very efficiently by resolving a set of sparse linear equations. Based on the new framework, a robust iterative method is employed to handle large outliers. We have conducted an extensive set of experiments to evaluate the performance on both synthetic and real-world testbeds, from which the promising results show that the proposed algorithm not only achieves better tracking accuracy, but also executes significantly faster than the previous solution.

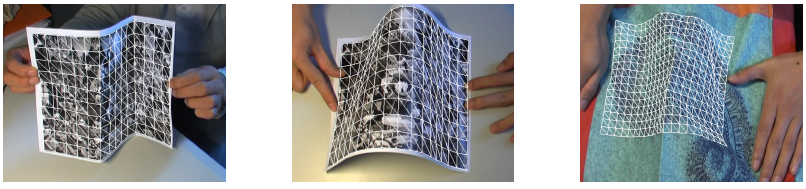
1 Introduction

Deformable surface modeling and tracking has attracted extensive research interest due to its significant role in many computer vision applications [1,2,3,4,5,6].

Since the deformable surface is usually highly dynamic and represented by many deformation parameters, the prior models are often engaged in dealing with the ill-posed optimization problem of deformable surface recovery. A variety of methods have been proposed to create these models, such as the interpolation method [1,7], the data embedding method [2,5,8] and physical models [9,10,11]. The major problem of these models is that their smoothness constraints usually limit their capability of recovering sharply folded and creased surfaces accurately.

Instead of using the strong prior models, M. Salzmann et al. recently formulated the problem generally as a Second Order Cone Programming (SOCP) problem without engaging the unwanted smoothness constraints [12]. Although they have demonstrated some promising results on tracking deformable surfaces from 3D to 2D correspondences, their approach is computationally expensive

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part III, LNCS 5304, pp. 766–779, 2008.



(a) Sharply folded (b) Bending (c) Cloth

Fig. 1. Recovering highly deformable surfaces from video sequences (a-c). (a) A piece of paper with well-marked creases. (b) Severely bending. (c) A piece of cloth.

while handling a large number of SOCP constraints for a large set of free variables. In this paper, we apply the principles they have described, and investigate new techniques to address the shortcomings.

Specifically, we propose a novel unconstrained quadratic optimization formulation for 3D deformable surface tracking, which requires only the solution of a set of sparse linear equations. In our approach, we first show that the SOCP formulation can be viewed as a special case of a general convex optimization feasibility problem. Then, we introduce a slack variable to rewrite the SOCP formulation into a series of Quadratic Programming (QP). Furthermore, we convert the SOCP constraints into a quadratic regularization term, which leads to a novel unconstrained optimization formulation. Finally, we show that the resulting unconstrained optimization problem can be solved efficiently by a robust progressive finite Newton optimization scheme [13], which can handle large outliers. Hence, not only is the proposed solution highly efficient, but also it can directly handle noisy data in an effective way. To evaluate the performance of our proposed algorithm, we have conducted extensive experiments on both synthetic and real-world data, as shown in Fig. 1.

The rest of this paper is organized as follows. Section 2 reviews the previous approaches to deformable surface recovery. Section 3 presents the proposed 3D deformable surface tracking solution using a novel unconstrained quadratic optimization method. Section 4 shows the details of our experimental implementation and evaluates the experimental results. Section 5 discusses some limitations and sets out our conclusion.

2 Related Work

We are motivated from the SOCP method [12], the convex optimization [14] and quasiconvex optimization [15] to the triangulation problem. Moreover, it is important to note that our work is closely related to previous work on structure from motion [16] as well as nonrigid surface detection and tracking [4,6,11,13].

Factorization methods are widely used in 3D deformable surface recovery. Bregler et al. [16] proposed a solution for recovering 3D nonrigid shapes from video sequences, which factorizes the tracked 2D feature points to build the 3D

shape model. In this approach, the 3D shape in each frame is represented by a linear combination of a set of basis shapes. A similar method was applied to the Active Appearance Models fitting results in order to retrieve the 3D facial shapes from video sequences [8]. Based on the factorization method, a weak constraint [17] can be introduced to handle the ambiguities problem by constraining the frame-to-frame depth variations. In addition, machine learning techniques have also been applied to building the linear subspace from either the collected data or the synthetic data. Although some promising results have been achieved in 3D face fitting [18] and deformable surface tracking [2], these methods usually require a large number of training samples to obtain sufficient generalization capability.

As for 2D nonrigid surface detection, J. Pilet et al. [11] proposed a real-time algorithm which employs a semi-implicit optimization approach to handle noisy feature correspondences. In contrast, several image registration methods [1,7] tend to be computationally expensive and are mainly aimed at object recognition.

3 Fast 3D Deformable Surface Tracking

In this section, we first formally define the 3D deformable surface tracking problem. Then we present an optimization framework for treating the 3D deformable surface tracking problem as a general convex optimization feasibility problem. We then revisit previous SOCP work that can be viewed as a special case of the general convex optimization framework. With a view to improving the efficiency of the optimization, we present techniques to relax the SOCP constraints properly and propose two new optimization formulations. One is a QP formulation and the other is an efficient unconstrained quadratic optimization.

3.1 Problem Definition

The 3D deformable surface is explicitly represented by triangulated meshes. As shown in Fig. 1, we employ a triangulated 3D mesh with n vertices, which are formed into a shape vector $\mathbf{s}$ as below:

$$\mathbf{s} = \begin{bmatrix} x_1 \ldots x_n \; y_1 \ldots y_n \; z_1 \ldots z_n \end{bmatrix}^\top$$

in which we define $\mathbf{v}_i = (x_i, y_i, z_i)^\top$ as the coordinates of the i^{th} mesh vertex. The shape vector $\mathbf{s}$ is the variable to be estimated.

Given a set of 3D to 2D correspondences $\mathcal{M}$ between the surface points and the image locations, a pair of matched points is defined as $\mathbf{m} = (\mathbf{m}_S, \mathbf{m}_I) \in \mathcal{M}$, where $\mathbf{m}_S$ is the 3D point on the surface and $\mathbf{m}_I$ is the corresponding 2D location on the input image.

We assume that the surface point $\mathbf{m}_S$ lies on a facet whose three vertices' coordinates are $\mathbf{v}_i, \mathbf{v}_j$ and $\mathbf{v}_k$ respectively, and $\{i, j, k\} \in [1, n]$ is the index of each vertex. The piecewise affine transformation is used to map the surface points $\mathbf{m}_S$ inside the corresponding triangle into the vertices in the mesh:

$$\mathbf{m}_S = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{bmatrix} \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 \end{bmatrix}^\top$$

where $(\xi_1, \xi_2, \xi_3)^\top$ are the barycentric coordinates for the surface point $\mathbf{m}_S$.

As in [12], we assume that the 3×4 camera projection matrix $\mathbf{P}$ is known and remains constant. This does not mean that the camera is fixed, since the relative motion with respect to the camera can be recovered during the tracking process. Hence, with the projection matrix $\mathbf{P}$, we can compute $\mathbf{m}_I = [\, u \; v \,]^\top$, the 2D projection of the 3D surface point $\mathbf{m}_S$, as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{P_{1,1}x + P_{1,2}y + P_{1,3}z + P_{1,4}}{P_{3,1}x + P_{3,2}y + P_{3,3}z + P_{3,4}} \\ \frac{P_{2,1}x + P_{2,2}y + P_{2,3}z + P_{2,4}}{P_{3,1}x + P_{3,2}y + P_{3,3}z + P_{3,4}} \end{bmatrix} \tag{1}$$

In order to directly represent the projection by the variables $\mathbf{s}$, an augmented vector $\mathbf{a} \in \mathbb{R}^{3n}$ is defined as below:

$$\begin{array}{lll} \mathbf{a}_i = \xi_1 P_{1,1} & \mathbf{a}_{i+n} = \xi_1 P_{1,2} & \mathbf{a}_{i+2n} = \xi_1 P_{1,3} \\ \mathbf{a}_j = \xi_2 P_{1,1} & \mathbf{a}_{j+n} = \xi_2 P_{1,2} & \mathbf{a}_{j+2n} = \xi_2 P_{1,3} \\ \mathbf{a}_k = \xi_3 P_{1,1} & \mathbf{a}_{k+n} = \xi_3 P_{1,2} & \mathbf{a}_{k+2n} = \xi_3 P_{1,3} \end{array}$$

The remaining elements of the vector $\mathbf{a}$ are all set to zero. Similarly, we define other two vectors $\mathbf{b}, \mathbf{c} \in \mathbb{R}^{3n}$ accordingly, and then rewrite Eqn. 1 as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{a}^\top \mathbf{s} + P_{1,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} \\ \frac{\mathbf{b}^\top \mathbf{s} + P_{2,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} \end{bmatrix} \tag{2}$$

3.2 Convex Optimization Formulations

General Convex Formulation. Since it is impossible to find a perfect projection that can ideally match all the 3D to 2D correspondences in practice, we let γ denote the upper bound for the reprojection error of each correspondence pair $\mathbf{m} \in \mathcal{M}$. As a result, for each 2D image observation $\mathbf{m}_I = [\, \hat{u} \; \hat{v} \,]^\top$, the following inequality constraint will be satisfied:

$$\left\| \frac{\mathbf{a}^\top \mathbf{s} + P_{1,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{u}, \frac{\mathbf{b}^\top \mathbf{s} + P_{2,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{v} \right\|_p \leq \gamma \text{ for } \mathbf{m} \in \mathcal{M}, \tag{3}$$

where $p \geq 1$ is a constant integer and the inequality constraint is known as a p-norm cone constraint [19]. As a result, the 3D deformable surface tracking problem can be formulated as a general convex optimization problem:

$$\begin{aligned} &\min_{\gamma \geq 0, \mathbf{s}} \; \gamma \\ &\text{s. t.} \; \left\| \frac{\mathbf{a}^\top \mathbf{s} + P_{1,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{u}, \frac{\mathbf{b}^\top \mathbf{s} + P_{2,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{v} \right\|_p \leq \gamma \quad \text{for each } \mathbf{m} \in \mathcal{M}. \end{aligned}$$

In the above optimization, γ is usually set by the bisection algorithm [12,14]. Hence, the tracking problem can be regarded as a feasibility problem for the above general convex optimization.

When $p = 2$, the p-norm cone constraint above reduces to the well-known SOCP constraint. In the following discussion, we will show that a recently proposed SOCP formulation can be viewed as a special case of the above general convex optimization feasibility problem.

SOCP Formulation. The recent work in [12] formulated the 3D deformable surface tracking problem as an SOCP feasibility problem, which can be viewed as a special case of the above general convex optimization with $p = 2$:

$$\begin{aligned} &\min_{\gamma \geq 0, \mathbf{s}} \ \gamma \\ &\text{s. t.} \ \left\| \frac{\mathbf{a}^\top \mathbf{s} + P_{1,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{u}, \frac{\mathbf{b}^\top \mathbf{s} + P_{2,4}}{\mathbf{c}^\top \mathbf{s} + P_{3,4}} - \hat{v} \right\| \leq \gamma \quad \text{for each } \mathbf{m} \in \mathcal{M}. \end{aligned} \tag{4}$$

where the 2-norm notation $\|\cdot\|_2$ is by default written as $\|\cdot\|$ without ambiguity. To handle the outliers, we employ the method [20] to remove the set of matches whose reprojection errors equal the minimal γ.

In practice, to regularize the deformable surface, an additional constraint is introduced to prevent irrational changes of the edge orientations between two consecutive frames [12]. We assume that the shape $\mathbf{s}^t$ at time t is known, and that the orientation of the edge linking the vertices $\mathbf{v}_i^t$ and $\mathbf{v}_j^t$ will be similar at time $t+1$. For each edge in the triangulated mesh, the corresponding constraint can be formulated below:

$$\left\| \mathbf{v}_i^{t+1} - \mathbf{v}_j^{t+1} - \boldsymbol{\theta}_{ij}^t \right\| \leq \lambda L_{i,j} \tag{5}$$

where $L_{i,j}$ is the original length of the edge. $\boldsymbol{\theta}_{ij}^t$ is the difference of the two vertices $\mathbf{v}_i^t$ and $\mathbf{v}_j^t$ at time t normalized by the original edge length, namely $\boldsymbol{\theta}_{ij}^t = L_{i,j} \frac{\mathbf{v}_i^t - \mathbf{v}_j^t}{\|\mathbf{v}_i^t - \mathbf{v}_j^t\|}$. Also, λ is a coefficient to control the regularity of the deformable surface. Again, the above inequality constraint is also an SOCP constraint. As a result, the tracking problem is formulated as an SOCP feasibility problem [1] with a number of SOCP constraints, which can be solved by some bisection algorithm [12,14].

A major problem of the above formulation is that the number of correspondences $|\mathcal{M}|$ is often much larger than the number of variables for ensuring sufficient correct matches, and thus the SOCP formulation has to engage a large number of SOCP constraints. Specifically, if n_e denotes the number of edges in the mesh model, the above SOCP formulation should have $(|\mathcal{M}| + n_e)$ SOCP constraints in total. Solving the above SOCP optimization directly leads to very high computational cost in practice.

[1] The SOCP optimization problem is solved by Sedumi: http://sedumi.mcmaster.ca.

QP Formulation. The drawback of the SOCP formulation lies in the large number of SOCP constraints. In this part, we present a QP formulation by removing the SOCP constraints. Specifically, for each of the SOCP constraints in Eqn. 4, we can rewrite it equivalently as

$$[(\mathbf{a}-\hat{u}\mathbf{c})^\top\mathbf{s}+d_u]^2+[(\mathbf{b}-\hat{v}\mathbf{c})^\top\mathbf{s}+d_v]^2\leq\gamma(\mathbf{c}^\top\mathbf{s}+d_w)^2$$

where $d_w=P_{3,4}$, $d_u=P_{1,4}-\hat{u}d_w$, and $d_v=P_{2,4}-\hat{v}d_w$. Further, we can introduce a slack variable $\epsilon(\mathbf{m})$ for each $\mathbf{m}\in\mathcal{M}$ and rewrite the inequality constraint as the following equality:

$$[(\mathbf{a}-\hat{u}\mathbf{c})^\top\mathbf{s}+d_u]^2+[(\mathbf{b}-\hat{v}\mathbf{c})^\top\mathbf{s}+d_v]^2+\epsilon(\mathbf{m})^2=\gamma(\mathbf{c}^\top\mathbf{s}+d_w)^2$$

In addition, we can replace the SOCP constraints in Eqn. 5 with 1-norm cone constraints. As a result, we can rewrite the original formulation by a min-max optimization formulation:

$$\begin{aligned}&\min_{\gamma\geq 0}\max_{\mathbf{s}}\sum_{\mathbf{m}\in\mathcal{M}}\epsilon(\mathbf{m})^2\\&\quad\text{s. t. }\left\|\mathbf{v}_i^{t+1}-\mathbf{v}_j^{t+1}-\boldsymbol{\theta}_{ij}^t\right\|_1\leq\lambda L_{i,j}\qquad\text{for each edge }(\mathbf{v}_i,\mathbf{v}_j)\text{ in the mesh.}\end{aligned}$$

in which the objective function can be expressed as:

$$\sum_{\mathbf{m}\in\mathcal{M}}\epsilon(\mathbf{m})^2=-(\mathbf{s}^\top\mathbf{H}\mathbf{s}+2\mathbf{g}^\top\mathbf{s}+d)\tag{6}$$

where $\mathbf{H}\in\mathbb{R}^{3n\times 3n}$, $\mathbf{g}\in\mathbb{R}^{3n\times 1}$ and $d\in\mathbb{R}$ are defined as:

$$\begin{aligned}\mathbf{H}&=\sum_{\mathbf{m}\in\mathcal{M}}(\mathbf{a}-\hat{u}\mathbf{c})(\mathbf{a}-\hat{u}\mathbf{c})^\top+(\mathbf{b}-\hat{v}\mathbf{c})(\mathbf{b}-\hat{v}\mathbf{c})^\top-\gamma\mathbf{c}\mathbf{c}^\top\\\mathbf{g}&=\sum_{\mathbf{m}\in\mathcal{M}}d_u(\mathbf{a}-\hat{u}\mathbf{c})+d_v(\mathbf{b}-\hat{v}\mathbf{c})-\gamma\mathbf{c}\\d&=\sum_{\mathbf{m}\in\mathcal{M}}d_u^2+d_v^2-\gamma d_w^2\end{aligned}$$

It is clear that the above objective function is quadratic. For the tracking task to be an optimization feasibility problem, γ is assumed to be known. Hence, the min-max optimization becomes a standard QP problem. To solve it, we also employ the bisection algorithm and engage an interior-point optimizer [2].

3.3 Unconstrained Quadratic Optimization

The QP formulation still has to include a number of 1-norm cone constraints. To address it, we present an unconstrained quadratic optimization formulation that completely relaxes all constraints. Specifically, instead of engaging the SOCP

[2] http://www.mosek.com/

constraints in Eqn. 5, we integrate such constraints into the objective function by treating it as a weighted penalty function, which converts the complex SOCP constraints into a simple quadratic term. This leads to the following unconstrained minimization formulation:

$$\min_{\gamma,\mathbf{s}} -\sum_{\mathbf{m}\in\mathcal{M}} \epsilon^2 + \mu \sum_{k=1}^{n_e} \eta_k^2 \tag{7}$$

where μ is a regularization coefficient, and η_k is a variable to constrain the regularity of the k^{th} edge:

$$\eta_k = \left\| \mathbf{v}_i^{t+1} - \mathbf{v}_j^{t+1} - \boldsymbol{\theta}_{ij}^t \right\|$$

Moreover, the edge regularization term can be expressed as:

$$\sum_{k=1}^{n_e} \eta_k^2 = \mathbf{s}^\top \mathbf{Q}\mathbf{s} - 2\mathbf{f}^\top \mathbf{s} + \varphi \tag{8}$$

where $\mathbf{Q} \in \mathbb{R}^{3n\times 3n}$, $\mathbf{f} \in \mathbb{R}^{3n\times 1}$ and $t \in \mathbb{R}$ are defined as:

$$Q = \sum_{k=1}^{n_e} \tilde{\mathbf{a}}\tilde{\mathbf{a}}^\top + \tilde{\mathbf{b}}\tilde{\mathbf{b}}^\top + \tilde{\mathbf{c}}\tilde{\mathbf{c}}^\top$$

$$\mathbf{f} = \sum_{k=1}^{n_e} \theta_x \tilde{\mathbf{a}} + \theta_y \tilde{\mathbf{b}} + \theta_z \tilde{\mathbf{c}}, \qquad \varphi = \sum_{k=1}^{n_e} \|\boldsymbol{\theta}_k\|$$

where $\boldsymbol{\theta}_k = (\theta_{kx}, \theta_{ky}, \theta_{kz})^\top$ is used to denote $\boldsymbol{\theta}_{ij}^t$. For the k^{th} edge with vertices $\mathbf{v}_i$ and $\mathbf{v}_j$, three augmented vectors $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}$ and $\tilde{\mathbf{c}} \in \mathbb{R}^{3n}$ are defined as follows:

$$\begin{array}{lll} \tilde{\mathbf{a}}_i = 1 & \tilde{\mathbf{b}}_{i+n} = 1 & \tilde{\mathbf{c}}_{i+2n} = 1 \\ \tilde{\mathbf{a}}_j = -1 & \tilde{\mathbf{b}}_{j+n} = -1 & \tilde{\mathbf{c}}_{j+2n} = -1 \end{array}$$

and the remaining elements in $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}$ and $\tilde{\mathbf{c}}$ are all set to zero. By substituting Eqn. 6 and Eqn. 8 into Eqn. 7, we can thus obtain the following unconstrained minimization formulation:

$$\min_{\gamma\geq 0,\mathbf{s}} \mathbf{s}^\top(\mathbf{H} + \mu\mathbf{Q})\mathbf{s} + 2(\mathbf{g} - \mu\mathbf{f})^\top \mathbf{s} + d + \varphi \tag{9}$$

Remark. In the above formulation, $\mathbf{H}$, $\mathbf{g}$ and d are all related to the upper bound variable γ, which seems like a complicated optimization problem. Fortunately, we find that the upper bound γ plays the same role as the support of the robust estimator in [11,13], which is able to handle large outliers. Therefore, the above problem can be perfectly solved by the progressive finite Newton method as proposed in [6,13], which makes the proposed method capable of handling large outliers. Specifically, the upper bound γ starts at a large value, and then is

progressively decreased at a constant rate. For each value of the upper bound γ, we can simply solve the following linear equation:

$$(\mathbf{H} + \mu\mathbf{Q})\mathbf{s} = -\mathbf{g} + \mu\mathbf{f} \tag{10}$$

where $\mathbf{H}$ and $\mathbf{g}$ are computed with the inlier matches only. We employ the results from previous step to compute the inlier set. Obviously, the square matrix $\mathbf{Q}$ is kept constant for the given triangulated mesh, and $\mathbf{f}$ only needs to be computed once for each frame. Since both $\mathbf{H}$ and $\mathbf{Q}$ are sparse matrices, the above linear system can be solved very efficiently by a sparse linear solver. Owing to its high efficiency, the proposed solution enables us to handle very large scale 3D deformable surface tracking problems with high resolution meshes.

4 Experimental Results

In this section, we present the details of our experimental implementation and report the empirical results on 3D deformable surface tracking. First, we perform an evaluation on synthetic data for comparison with the convex optimization method. Then, we show results of our proposed approach in various environments, which demonstrate that our method is both efficient and effective for 3D deformable surface tracking.

4.1 Experimental Setup

All the experiments reported in this paper were carried out on an Intel Core2 Duo 2.0GHz Notebook Computer with 2GB RAM, and a DV camera was engaged to capture the videos. For simplicity, our QP formulation is denoted as "QP", and the proposed unconstrained quadratic optimization method is denoted as "QO". All the methods are implemented in Matlab, in which some routines were written in C code. Instead of relying on the 2D tracking results as in [12], we directly employ the SIFT method [21] to build the 3D to 2D correspondences by matching the model image and the input image. The planar surface with a template image is used due to its simplicity. Moreover, the non-planar surface can be employed by embedding the texture into 2D space.

For the SOCP method, we use the similar parameters settings as given in [12]. Specifically, in our experiments, λ is set to 0.1, and the bisection algorithm stops when the maximal reprojection error is below one pixel. For the proposed QO method, the regularization parameter μ is found by grid searching, which is set to 5×10^4 for all experiments. The decay rate for the upper bound γ is set to 0.5.

To initialize the 3D tracking, we register the first frame by the 2D nonrigid surface detection method [11], and then estimate the camera projection matrix $\mathbf{P}$ from 3D to 2D correspondences. In fact, the tracking usually starts from a surface that is slightly deformed. This method works well in practice, and it can automatically fit to the correct positions even when the initialization is not very accurate.

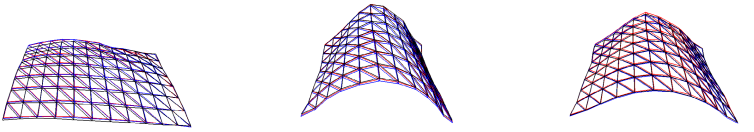
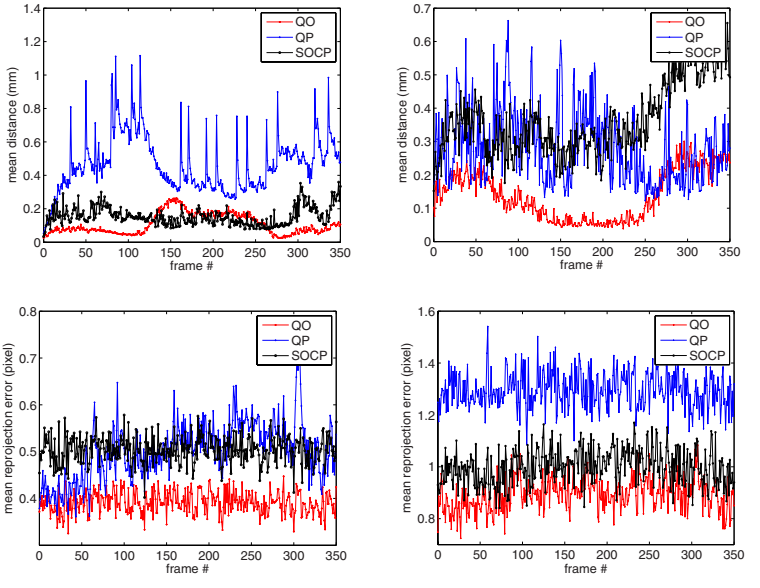


Fig. 2. Synthetic meshes with 96 vertices for evaluation. The 2D observations corrupted by noise having a normal distribution with $\sigma = 2$. Results for SOCP (black) and QO (blue) are shown with ground truth (red), at frame 94, 170 and 220.



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

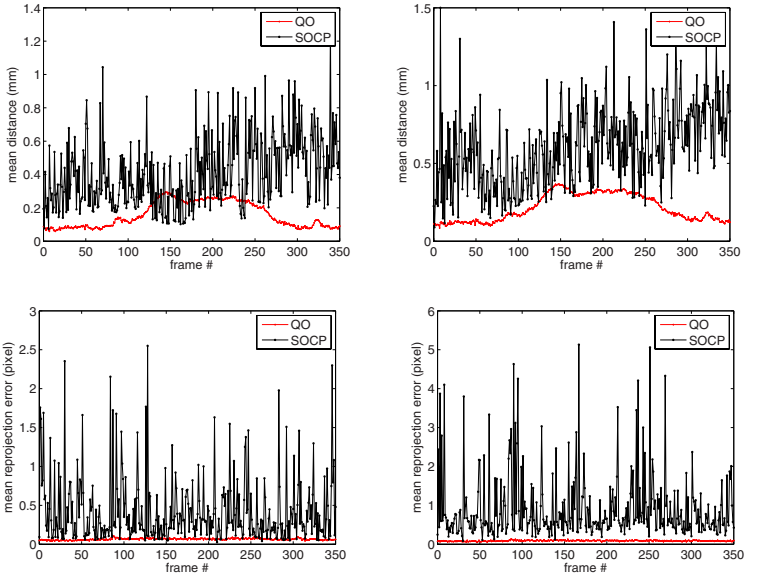
Fig. 3. The performance comparison of the QO, QP and SOCP methods on the 350 synthetic meshes with little added noise. The first row shows the average distance between ground truth and recovery results. The second row is the mean reprojection errors.

4.2 Synthetic Data Comparison

We generate a sequence of 350 synthetic meshes by simulating a surface bending process as shown in Fig. 2. The total size of the mesh is $280mm \times 200mm$. Given a perspective projection matrix $\mathbf{P}$, the 2D correspondences are obtained by projecting the 3D points defined by piecewise affine mapping, where the barycentric coordinates are randomly selected. We conduct two sets of experiments on the synthetic data. Firstly, we conduct the experiments on 2D observations with a small amount of added noises. Secondly, we evaluate the performance of SOCP and QO methods on data with large outliers. We set the number of correspondences in each facet to 5 for the first experiment, and 10 for the second one.

Experiment I. In the first experiment, we consider two cases of noisy data, for which the noise is added to all the 2D observations based on a normal distribution with different standard deviations $\sigma = 1, 2$. Fig. 3 shows the results of the comparison between the QO, QP and SOCP methods. We can see that the proposed QO method achieves the lowest reprojection errors for both cases. When $\sigma = 1$, both QO and SOCP are more effective than the QP formulation in 3D reconstruction performance. Indeed, there is some large jittering for the QP method in 3D reconstruction. This may be due to the L_1 norm relaxation of the constraints that may cause ambiguities in depth. Also, the SOCP method slightly outperforms the QO method when the surface is highly deformed, as observed around frame 170 in Fig. 2. When the standard deviation of the noise increases, we found that the proposed QO method achieves better and more steady results than the other two methods. This shows that the QO method is more resilient to noises.

Experiment II. In the second experiment, we conduct experiments on the synthetic data partially corrupted by noises (40% and 60% respectively) with standard deviation $\sigma = 10$. The experimental results shown in Fig. 4 demonstrate that the proposed QO approach is very robust, and more effective than the SOCP method in dealing with large outliers. Furthermore, we observe that the results achieved by the QO approach are rather smooth. In contrast, large



(a) 40% (b) 60%

Fig. 4. Comparison of the performance of the QO and SOCP on the synthetic data with large outliers. The first row shows the average distance between ground truth and recovery results. The second row is the mean reprojection errors.

jittering is observed in the results from the SOCP method. In our experiments, the number of inliers for the QO method is larger than that for the SOCP method. Specifically, when the percentage of outliers is 60%, the average inlier rate is around 39% for QO, and below 30% for the SOCP method.

Computational Efficiency. The complexity of the proposed QO method is mainly dominated by the order of Eqn. 10, which is equal to $3n$. Another important factor is the number of inlier matches, which affects the sparseness of the system matrix. This number usually differs from one frame to another. For the synthetic data with 96 vertices, as shown in Fig. 2, the proposed method runs at about 29 frames per second on the synthetic data. Furthermore, the proposed QO method takes 0.034 seconds per frame. On the other hand, the QP and SOCP method require 10 seconds and 5 seconds per frame respectively. On average, the proposed QO method is over 140 times faster than the SOCP method.

4.3 Performance on Real Data

Next, we investigate the 3D deformable surface tracking performance on some real deformable surfaces based on a piece of paper, a bag and a piece of cloth. Since only the QO method is efficient enough in practice, we evaluate only the QO method on the real data. To ensure that a sufficient number of correct correspondences are found, all the objects are well-textured.

Paper. As shown in Fig. 5, the proposed method is robust in handling large bending deformations. In practice, the whole process runs at around one frame per second on the DV size video sequence with a 187-vertex mesh model. The SIFT feature extraction and matching takes most of the time, whereas the optimization procedure only requires 0.1 seconds for each frame. Fig. 6 shows that a sharply folded surface is retrieved, and the well-marked creases can be accurately recovered.

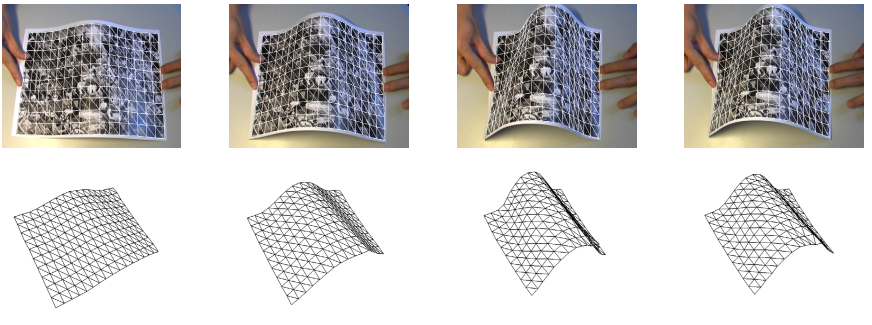


Fig. 5. We use a piece of paper as the deformable object. The deformable surface is recovered from a 300 frame video. The first row shows the images captured by a DV camera size of 720×576 overlaid by the reprojection of the recovered mesh. The second row is a different projective view of the recovered 3D deformable surface.

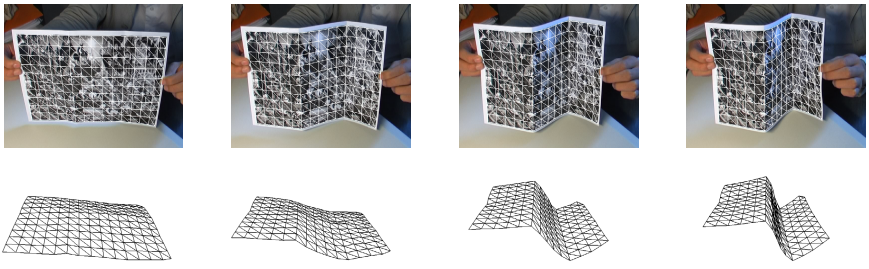


Fig. 6. Tracking the deformable surface with two sharp folds in it. The creases are correctly recovered.

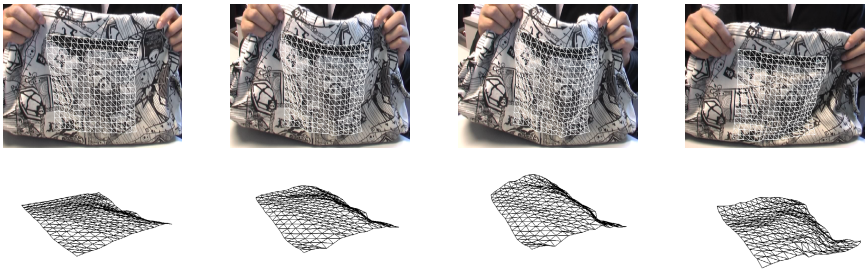


Fig. 7. Recovering the deformation of a bag

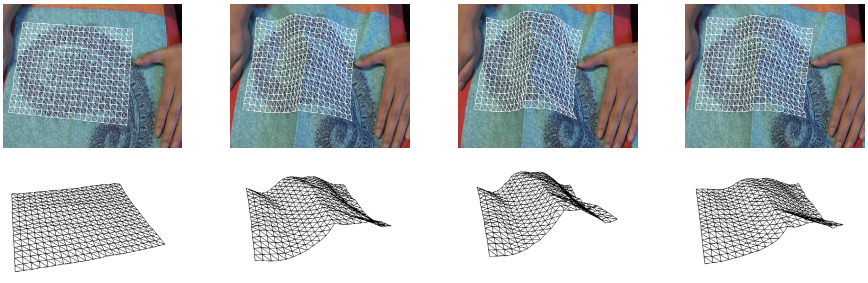


Fig. 8. Recovering the deformation of a piece of cloth

Bag and Cloth. To evaluate the performance on materials less rigid than a piece of paper, we reconstruct the surfaces of a bag and a piece of cloth with the proposed method. For the high efficiency of the proposed solution, we can handle real-world objects with high resolution mesh very fast. Fig. 7 shows the tracking results of the bag surface. The optimization procedure only takes about 0.2 seconds to process a mesh with 289 vertices. Similarly, Fig. 8 shows the tracking results of a piece of cloth. From these results, we can observe that the proposed method is able to recover the deformable surfaces accurately with the high resolution mesh.

5 Discussions and Conclusion

We have proposed a novel solution for the 3D deformable surface tracking by formulating the problem into an unconstrained quadratic optimization. Compared with previous convex optimization approaches, the proposed method enjoys several major advantages. Firstly, our method is very efficient without involving complicated SOCP constraints. Secondly, the proposed approach can handle large outliers and is more resilient to noises. Compared with the previous SOCP method, we have improved both the efficiency and robustness performance significantly. Furthermore, different from the previous SOCP approach that usually requires a sophisticated SOCP solver, our proposed method can be implemented easily in practice, requiring the solution of only a set of linear equations. Also, the optimization method used in this paper might be applicable to other similar problems solved by SOCP. We have conducted experimental evaluations on objects made of different materials. The experimental results show that the proposed method is significantly more efficient than the previous approach, and is also rather robust to noises. Promising tracking results show that the proposed solution is able to handle large deformations that often occur in real-world applications.

Although promising experimental results have validated the efficiency and effectiveness of our methodology, some limitations should be addressed as future work. First of all, self-occlusion problem has not yet been studied. Also, in some situations we found that some jitter may occur due to a lack of texture information. To address the problem, we may consider employing the visible surface detection algorithm. Furthermore, global bundle-adjustment can be fitted into our optimization framework, which will help handle the jittering problem. Finally, we will consider an efficient GPU-based point-matching algorithm to facilitate real-time 3D deformable surface tracking applications.

Acknowledgments

The work was fully supported by the Research Grants Council Earmarked Grant (CUHK4150/07E), and the Singapore MOE AcRF Tier-1 research grant (RG67/07).

References

1. Bartoli, A., Zisserman, A.: Direct estimation of non-rigid registration. In: Proc. British Machine Vision Conference, Kingston (September 2004)
2. Salzmann, M., Pilet, J., Ilic, S., Fua, P.: Surface deformation models for nonrigid 3d shape recovery. IEEE Trans. Pattern Anal. Mach. Intell. 29(8), 1481–1487 (2007)
3. Tsap, L.V., Goldgof, D.B., Sarkar, S.: Nonrigid motion analysis based on dynamic refinement of finite element models. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(5), 526–543 (2000)
4. White, R., Forsyth, D.A.: Combining cues: Shape from shading and texture. In: Proc. Conf. Computer Vision and Pattern Recognition, pp. 1809–1816 (2006)

5. Zhu, J., Hoi, S.C., Lyu, M.R.: Real-time non-rigid shape recovery via active appearance models for augmented reality. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 186–197. Springer, Heidelberg (2006)
6. Zhu, J., Lyu, M.R., Huang, T.S.: A fast 2d shape recovery approach by fusing features and appearance. IEEE Trans. Pattern Anal. Mach. Intell (to appear, 2008)
7. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding 89(2-3), 114–141 (2003)
8. Xiao, J., Baker, S., Matthews, I., Kanade, T.: Real-time combined 2d+3d active appearance models. In: Proc. Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 535–542 (2004)
9. Fua, P., Leclerc, Y.: Object-centered surface reconstruction: Combining multi-image stereo and shading. Int'l J. Computer Vision 16(1), 35–56 (1995)
10. Ilic, S., Fua, P.: Implicit meshes for surface reconstruction. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(2), 328–333 (2006)
11. Pilet, J., Lepetit, V., Fua, P.: Fast non-rigid surface detection, registration, and realistic augmentation. Int'l J. Computer Vision (2007)
12. Salzmann, M., Hartley, R., Fua, P.: Convex optimization for deformable surface 3-d tracking. In: Proc. Int'l Conf. Computer Vision (October 2007)
13. Zhu, J., Lyu, M.R.: Progressive finite newton approach to real-time nonrigid surface detection. In: Proc. Conf. Computer Vision and Pattern Recognition, pp. 1–8 (2007)
14. Kahl, F.: Multiple view geometry and the l_∞-norm. In: ICCV, pp. 1002–1009 (2005)
15. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. IEEE Trans. Pattern Anal. Mach. Intell. 29(10), 1834–1847 (2007)
16. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: Conf. Computer Vision and Pattern Recognition, pp. 690–696 (2000)
17. Salzmann, M., Lepetit, V., Fua, P.: Deformable surface tracking ambiguities. In: Proc. Conf. Computer Vision and Pattern Recognition (2007)
18. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. IEEE Trans. on Pattern Analysis and Machine Intelligence 25(9), 1063–1074 (2003)
19. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
20. Sim, K., Hartley, R.: Removing outliers using the $L\infty$ norm. In: Proc. Conf. Computer Vision and Pattern Recognition, pp. 485–494 (2006)
21. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int'l J. Computer Vision 60(2), 91–110 (2004)