Deformable Surface Tracking by Graph Matching

Tao Wang[1,2] Haibin Ling[3] Congyan Lang[1] Songhe Feng[1] Xiaohui Hou[2]
[1]Beijing Jiaotong University, Beijing 100044, China
[2]HiScene Information Technologies, Shanghai 201210, China
[3]Stony Brook University, Stony Brook, NY 11794, USA.
{twang,cylang,shfeng}@bjtu.edu.cn, hling@cs.stonybrook.edu, houxh@hiscene.com

Abstract

This paper addresses the problem of deformable surface tracking from monocular images. Specifically, we propose a graph-based approach that effectively explores the structure information of the surface to enhance tracking performance. Our approach solves simultaneously for feature correspondence, outlier rejection and shape reconstruction by optimizing a single objective function, which is defined by means of pairwise projection errors between graph structures instead of unary projection errors between matched points. Furthermore, an efficient matching algorithm is developed based on soft matching relaxation. For evaluation, our approach is extensively compared to state-of-the-art algorithms on a standard dataset of occluded surfaces, as well as a newly compiled dataset of different surfaces with rich, weak or repetitive texture. Experimental results reveal that our approach achieves robust tracking results for surfaces with different types of texture, and outperforms other algorithms in both accuracy and efficiency.

1. Introduction

This paper addresses the problem of tracking a generic deformable surface with a known initial 3D shape, namely *template*, and recovering its 3D shape in a video sequence under monocular perspective projection. The template could be provided manually in advance or computed from a few video frames using *shape-from-motion* [1, 27].

Popular approaches to deformable surface tracking can be roughly classified as dense approaches (*e.g.* [12, 25, 31, 48]) or feature-based ones (*e.g.* [4,9,24,34]). Dense approaches directly use pixel appearance without extracting features, and optimize a similarity measure between a template and a captured image. This type of approaches is usually guided by the brightness consistency assumption, and thus suffers from illumination change, partial occlusion and motion blur. Besides, most of them cost much computational time due to the large parameter space. Feature-based approaches perform shape reconstruction based on point correspondences between the template and an input image. Once point correspondences can be established, many *shape-from-template* approaches [4, 7, 9, 22, 24, 34] can reconstruct the 3D shape in the input image. These methods rely on the quality of the correspondences, and most of them establish correspondences based solely on local appearance without sufficient consideration of the spatial relationships among the feature points and the constraints imposed. Therefore, they often fail if the texture quality is too poor to guarantee reliable correspondences, as happens in the presence of repetitive patterns, dramatic deformation between the template and the input image, and environmental perturbations such as illumination change.

To obtain high quality correspondences between two images, graph-based methods [13,42,49] are widely used by constructing graphs that encode the geometric relationships between feature points and then accomplishing correspondences by means of graph matching. However, these graph-based methods suffer from several shortages when applied to deformable surface tracking. Firstly, graph matching amounts to an NP-hard binary programming problem, and many graph matching algorithms [47, 49] may take several minutes to process a few hundred points despite some approximation strategies are employed. It is therefore difficult to use them directly in 3D shape reconstruction where thousands of reliable correspondences are usually required to compute an accurate 3D shape. Secondly, these graph-based methods are independent of subsequent steps of outlier rejection and shape reconstruction, which may hurt the accuracy of the reconstructed shape due to lack of information about the deformation model.

Addressing the issues discussed above, we propose a novel graph-based method to deformable surface reconstruction and tracking. Different from traditional methods that usually treat separately feature correspondence, outlier rejection and shape reconstruction, we integrate these procedures into a unified graph-based framework, and propose to solve optimizations of correspondence and defor-

mation iteratively. Considering computational efficiency, we relax the *hard matching* constraint in the conventional graph matching problem to *soft matching* constraint. Such soft relaxation allows us to maintain more matching details that result in more accurate shape, and also benefits greatly to the computational efficiency through a novel matching algorithm developed under the soft matching constraint. Accompanied with a well-designed strategy for candidate match filtering, our graph-based approach is able to process thousands of points in a few seconds, which is much faster than conventional graph-based algorithms.

For a thorough evaluation, we compare our approach with several recently proposed approaches [8,24,25] on two datasets: the *tracking surface with occlusion* (TSO) benchmark [25] containing two video sequences in presence of occlusions, and a newly collected dataset[1] containing 11 sequences involving different surfaces with rich, weak or repetitive texture under significant deformation. On all sequences our approach produces the best or nearly the best tracking results. Regarding the computational efficiency, it also outperforms the compared algorithms in general.

In summary, our contribution lies in the new graph-based approach for deformable surface tracking in three aspects: (1) we introduce graph model and graph matching into deformable surface tracking by a soft matching relaxation and a well-designed strategy of candidate match filtering; (2) we design a unified optimization framework to explore full information about local appearance, spatial relationship and deformation model to acquire accurate shape reconstruction; and (3) we construct a new real world dataset with annotation for evaluating deformable surface tracking algorithms in the context of different types of textures.

2. Related Work

Recovering the nonrigid shape of a surface from input images usually includes three steps: *keypoint correspondence*, *outlier rejection*, and *shape reconstruction*. In the following we sample some classical studies or related ones that inspire our study on *deformable surface tracking*.

Keypoint correspondence refers to extracting keypoints from given images and subsequently relating them by some distance metric to keypoints in a nearest-neighbour fashion. Some popular keypoint detectors and descriptors (*e.g.* SIFT [21] and SURF [5]) are designed to be robust against changes in scale and rotation.Aiming at real-time systems, several extremely fast keypoint detectors and binary descriptors [20, 30] were developed as well. A simple way to establish correspondence between extracted keypoints is to assign each point to the point with the closest descriptor. This naive approach suffers from both intrinsic disturbance (*e.g.* repetitive patterns) and extrinsic variation (*e.g.* lighting change). In order to improve the robustness against such perturbations, some approaches [42, 46, 49] construct graphs to encode the geometric relationship between keypoints, and reformulate keypoint correspondence as a graph matching problem. Solving graph matching amounts to an NP-hard binary programming problem, and approximate solutions [19, 47] are commonly applied to finding efficient and tight relaxations.

Outlier rejection works by fitting a deformable model using the established correspondences and eliminating incorrect ones that conflict with the fitted model. Existing methods of outlier detection can be roughly categorized into 3D methods and 2D methods, which use 3D and 2D deformable models respectively. The main advantage of 3D methods [11, 26, 32, 38] lies in that they can use physically meaningful properties, *e.g.* isometry, which are invariant to changes of the camera viewpoint or camera parameters. On the contrary, 2D methods [28, 29, 43] cannot exploit surface isometry without involving 3D constraints, and are thus necessary to impose some general assumptions on the 2D-2D flowfield. Usually, these methods assume the flowfield is globally or piecewise smooth.

Shape reconstruction estimates the nonrigid shape of the surface based on the known template and the established keypoint correspondence. Such correspondence between the template and the input image allows one to compute a 2D warp and then infer a 3D shape in closed form [3]. However, the 2D warp does not take the 3D constraints into account, and thus may hurt the accuracy of the recovered shape. Alternatively, recent methods tend to compute directly from correspondences to 3D shape, which result in solving degenerate linear systems [33]. To handle this ill-conditioned problem, a large number of methods employ dimensionality reduction techniques, such as *principal component analysis* (PCA) [6, 14, 16], *free form deformations* (FFD) [7], model analysis [22, 23] and Laplacian formalism [24, 41], to reduce the degree of freedom. In addition to dimensionality reduction, another popular way is to impose some additional constraints to make the problem well-posed. Isometry constraints [2, 8, 9, 22] that involve preserving geodesic distances as the surface deforms or inextensibility constraints [7, 15, 24, 34, 39, 45] that prevent Euclidean distances between neighboring points from growing beyond a bound are commonly enforced in recent approaches. In particular, conformal deformation (angle-preserving) [4] relaxed from isometric deformation makes it applicable to some types of extensible surfaces.

Different from the above mentioned algorithms that treat keypoint correspondence, outlier rejection and shape reconstruction as separate steps, a few investigations have been devoted to solving simultaneously these problems by optimizing a single objective function. Examples include [38] that formulates these problems jointly in a mixed integer

[1]There is a serious lack of deformable tracking benchmarks due to difficulty in groundtruth annotation, the new dataset is collected for this reason.

quadratic form, and [37] that reduces the complexity of the joint optimization problem by using weak pose and shape priors, and [40] that encodes 3D shape reconstruction into an end-to-end deep neural network.

Our approach shares similarity with above algorithms in the use of geometric context to assist surface tracking, but differs in the context model (*i.e*. with graph matching). In particular, our work falls into the group of structure-aware tracking, with improvement in two-folds: (1) modeling keypoint correspondence and shape reconstruction in line with pairwise projection errors between graph structures, instead of conventional unary projection errors between keypoint sets, and (2) developing an efficient graph matching algorithm under soft matching relaxation. Our approach aims to provide accurate and efficient tracking for deformable surfaces, as validated in the experiments.

3. The proposed method

We represent a known template shape $\mathcal{T}$ as a triangulated mesh of N_v vertices $\{\mathbf{v}_i^r = [x_i, y_i, z_i]^T, 1 \leq i \leq N_v\}$ connected by a set E_{mesh} of N_e edges. We stack the vertices into a vector $\mathbf{x}^r \in \mathbb{R}^{3N_v}$, which is described in the camera reference frame. The known template $\mathcal{T}$ is related to the unknown deformed shape $\mathcal{S}$ by an unknown 3D continuously differentiable deformation $\psi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, *i.e*., ψ maps a point in $\mathcal{T}$ to the corresponding point in $\mathcal{S}$. Similarly, we can represent $\mathcal{S}$ using N_v vertices $\mathbf{v}_i$ with unknown 3D coordinates and stack them into a vector $\mathbf{x} \in \mathbb{R}^{3N_v}$, which is to be solved in our algorithm. We assume that the camera is calibrated, with known intrinsic and extrinsic parameters. That is, we have a known projection function $\tau : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ maps each 3D mesh point to a 2D image point.

Let $P^r = \{p_i^r\}_{i=1}^m$ and $P = \{p_i\}_{i=1}^n$ be the two feature sets extracted from the reference and input images, respectively. For each feature point $p_i^r \in P^r$ or $p_j \in P$, we also use the same symbol to indicate its homogeneous coordinates in the 2D image for simplicity. Since the 3D surface for the reference image is known, for each feature point $p_i^r \in P^r$ we can compute its 3D mesh point $\mathbf{p}_i^r \in \mathbb{R}^3$.

The correspondences between points in P^r and P are represented by a matrix $C \in \mathbb{R}^{m \times n}$ in which each element $C_{i,j} \in [0, 1]$ indicates the probability of assigning p_i^r to p_j. Note that we use *soft* correspondences here rather than *hard* ones that are commonly adopted in previous approaches. Soft correspondences allow us to maintain more correspondence details, and thus improve the accuracy of the recovered 3D shape. Another benefit brought by it lies in that the subsequent quadratic programming problem becomes much easier to be solved by dropping the discrete constraints.

The optimal shape $\mathcal{S}$ to be reconstructed can be obtained by solving simultaneously for both C and ψ that minimizing a cost function $\mathcal{E}(C, \psi)$:

$$
\begin{aligned}
&(C^*, \psi^*) = \arg\min_{C,\psi} \mathcal{E}(C, \psi), \\
&\text{s.t.} \quad \begin{cases} C \succcurlyeq \mathbf{0}_{m\times n}, C\mathbf{1}_n \preccurlyeq \mathbf{1}_m, C^T\mathbf{1}_m \preccurlyeq \mathbf{1}_n, \\ \|\psi(\mathbf{p}_i^r) - \psi(\mathbf{p}_j^r)\|_2 \leq l_{i,j}, \forall (i,j) \in E_{\text{mesh}}, \end{cases}
\end{aligned} \tag{1}
$$

where $\mathbf{0}_{m\times n}$ denotes a matrix of $m \times n$ zeros, $\mathbf{1}_n$ denotes a column vector of n ones, $\succcurlyeq$ ($\preccurlyeq$) are element-wise $\geq$ ($\leq$), and $l_{i,j}$ represents the constraint of the geodesic distance between points $\mathbf{p}_i^r$ and $\mathbf{p}_j^r$. The constrains on correspondence C guarantee that each point can be matched at most once, while those on deformation ψ are inextensibility constraints that prevent Euclidean distances between neighboring vertices from growing beyond a bound.

In previous approaches, the cost function $\mathcal{E}(C, \psi)$ is usually defined to accumulate the projection error of each correspondence $\langle \mathbf{p}_i^r, p_j \rangle$ under deformation ψ. In this paper, we propose a graph-based measure that assembles the projection errors between graph structures as

$$
\mathcal{E}(C, \psi) = \sum_{i,j} \sum_{a,b} d(\psi, i, j, a, b) C_{i,a} C_{j,b}, \tag{2}
$$

where $d(\psi, i, j, a, b)$ is cost function measuring the pairwise inconsistence between edges $(\mathbf{p}_i^r, \mathbf{p}_j^r)$ and (p_a, p_j) under deformation ψ. We define d as combination of an appearance inconsistence function d_{app} and a geometric inconsistence function d_{geo}, which are specified as

$$
\begin{aligned}
d_{\text{app}}(i, j, a, b) &= \|f_i^r - f_a\|_2 + \|f_j^r - f_b\|_2, \\
d_{\text{geo}}(\psi, i, j, a, b) &= \|(\tau(\psi(\mathbf{p}_i^r)) - \tau(\psi(\mathbf{p}_j^r))) - (p_a - p_b)\|_2, \\
d(\psi, i, j, a, b) &= (1-\alpha) d_{\text{app}}(i, j, a, b) + \alpha d_{\text{geo}}(\psi, i, j, a, b),
\end{aligned} \tag{3}
$$

where f_i^r and f_a are photometric descriptors of feature points p_i^r and p_a respectively, and $\alpha \in [0, 1]$ balances between local features and graph structures used for shape reconstruction.

For conciseness we can reformulate Eq. (2) in a pairwise compatibility form

$$
\mathcal{E}(C, \psi) = \mathbf{c}^T K(\psi) \mathbf{c}, \tag{4}
$$

where $\mathbf{c} \doteq \text{vec}(C)$ is the vectorized version of matrix C and $K(\psi) \in \mathbb{R}^{mn \times mn}$ is the corresponding affinity matrix:

$$
K_{\text{ind}(i,a),\text{ind}(j,b)}(\psi) = d(\psi, i, j, a, b) - \kappa, \tag{5}
$$

where (i, a) denotes a candidate match from point p_i^r in the reference image to point p_a in the input image, and $\text{ind}(\cdot)$ is a bijection that maps a vertex correspondence to an integer index. Note that κ is chosen to be sufficiently large to ensure that $K(\psi)$ is nonpositive, of which the purpose is to avoid the trivial solution in which no correspondence is activated.

To filter outlier correspondences with large projection errors under deformation ψ, we penalize matched points by means of projection error term which increases as more points are matched

$$
\mathcal{E}(C, \psi) = \mathbf{c}^T K(\psi) \mathbf{c} + \lambda \mathbf{c}^T e(\psi), \tag{6}
$$

where $\lambda > 0$ adaptively controls the degree of outlier rejection, and $e(\psi) \in \mathbb{R}^{mn}$ encodes the unary projection error of each point correspondence as

$$e_{\text{ind}(i,a)}(\psi) = \|\tau(\psi(\mathbf{p}_i^r)) - p_a\|_2. \tag{7}$$

4. Optimization

For each incoming frame, we first predict $\mathbf{c}$ and ψ using the solutions from previous frames, and then refine them alternatively and iteratively.

4.1. Optimization of correspondence

Given a deformation ψ, problem (1) is reduced to solving for an optimal correspondence as

$$\begin{aligned} \mathbf{c}^* = \arg\min_{\mathbf{c}} \mathbf{c}^T K(\psi)\mathbf{c} + \lambda \mathbf{c}^T e(\psi), \\ \text{s.t. } \mathbf{c} \succcurlyeq \mathbf{0}_{mn}, B\mathbf{c} \preccurlyeq \mathbf{1}_{m+n}, \end{aligned} \tag{8}$$

where $B\mathbf{c} \preccurlyeq \mathbf{1}_{m+n}$ encodes the one-to-one matching constraints (refer to [47] for details on constructing matrix B).

Problem (8) can be viewed as a relaxed graph matching problem by dropping the discrete constraints and adding a penalization term. Some power iteration algorithms [10, 19] for classical graph matching can be extended to solve for a soft correspondence $\mathbf{c}$, but it is hard apply them for (8) due to the penalization term. In this section, we propose an approach based on the Frank-Wolfe algorithm [17] for minimizing problem (8) with respect to correspondence $\mathbf{c}$, which is described in Algorithm 1.

4.2. Optimization of deformation

Given a correspondence $\mathbf{c}$ (*i.e.*, matrix C in (1)), problem (1) is reduced to solving for an optimal deformation as

$$\begin{aligned} \psi^* = \arg\min_{\psi} \Big\{ \sum_{i,j} \sum_{a,b} d_{\text{geo}}(\psi, i, j, a, b) C_{i,a} C_{j,b} \\ + \lambda \sum_{i,a} e_{\text{ind}(i,a)}(\psi) C_{i,a} \Big\}, \\ \text{s.t. } \|\psi(\mathbf{p}_i^r) - \psi(\mathbf{p}_j^r)\|_2 \le l_{i,j}, \forall (i,j) \in E_{\text{mesh}}. \end{aligned} \tag{9}$$

We relax the first term of problem (9) by

$$\begin{aligned} d_{\text{geo}}(\psi, i, j, a, b) &= \|(\tau(\psi(\mathbf{p}_i^r)) - \tau(\psi(\mathbf{p}_j^r))) - (p_a - p_b)\|_2 \\ &\le \|(\tau(\psi(\mathbf{p}_i^r)) - p_a\|_2 + \|\tau(\psi(\mathbf{p}_j^r)) - p_b\|_2. \end{aligned}$$

Problem (9) is thus relaxed as a linear fitting

$$\begin{aligned} \psi^* = \arg\min_{\psi} \sum_{i,a} \omega_{i,a} \|(\tau(\psi(\mathbf{p}_i^r)) - p_a\|_2, \\ \text{s.t. } \|\psi(\mathbf{p}_i^r) - \psi(\mathbf{p}_j^r)\|_2 \le l_{i,j}, \forall (i,j) \in E_{\text{mesh}}, \end{aligned} \tag{10}$$

Algorithm 1 Frank-Wolfe for correspondence $\mathbf{c}$

% ψ_0: given a deformation.
% Ω: solution space of feasible $\mathbf{c}$.
Initialization: compute matrix $K(\psi_0)$ and vector $e(\psi_0)$.
Initialization: initialize correspondence $\mathbf{c}$ as trivial.
while $\mathbf{c}$ not converged **do**
  $\mathbf{g} = 2K(\psi_0)\mathbf{c} + e(\psi_0)$ % gradient
  $\mathbf{y} = \arg\min_{\mathbf{y}} \mathbf{g}^T\mathbf{y}$, s.t. $\mathbf{y} \in \Omega$
  $\beta = \arg\min_{\beta} \mathcal{E}_\lambda(\mathbf{c} + \beta(\mathbf{y} - \mathbf{c}))$, s.t. $0 \le \beta \le 1$
  $\mathbf{c} \leftarrow \mathbf{c} + \beta(\mathbf{y} - \mathbf{c})$
end while
return $\mathbf{c}$

where $\omega_{i,a} = C_{i,a}(\sum_j C_{j,a} + \sum_b C_{i,b}) + \lambda$ is the weight for each sample.

As described in [24], this problem can be further reformulated to a well-conditioned linear system with respect to the coordinates of the mesh vertices as

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|M\mathbf{x}\|_2^2 + r\|A\mathbf{x}\|_2^2, \text{ s.t. } \|\mathbf{x}\|_2 = 1, \tag{11}$$

where M is a coefficient matrix, A a regularization matrix, and r a scalar coefficient defining how much we regularize the solution. More details about this conditioned linear system can be found in [24].

5. Implementation details

5.1. Graph construction

An undirected graph of n vertices can be represented by $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V} = \{v_1, \ldots, v_n\}$ and $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ denote the vertex and edge sets, respectively. Given the initial region $\mathcal{R}$ of the surface of interest in the reference image, we construct a model graph $\mathbb{G}^r$ for the surface as follows.

Vertex generation. It is common to extract keypoints from the image to represent local parts, and then model them as vertices of the graph. Many approaches obtain the keypoints as local extremes of cross-scale DoG images, *e.g.* SIFT. However, the number of the keypoints extracted using such methods may vary drastically depending on detectors and frame content. Moreover, they are often sensitive to environmental variations, such as illumination changes and motion blurs, and thus hurt the tracking accuracy.

We adopt a more robust and flexible way to address these issues. We first divide $\mathcal{R}$ evenly into N grids, and compute the SIFT response for each pixel in each grid. After that, we select the keypoint with maximum response from each grid, and treat such keypoints as graph vertices. Specifically, for vertex generation of a nonrectangular shape, we first divide its minimum bounding rectangle into even grids and extract a candidate vertex from each grid, and then remove invalid vertices lying outside the shape. Finally, the SIFT descriptors of these keypoints are recorded as vertex attributes.

Edge generation. There are several widely-used methods for edge generation, including the fully connected graph, the ε-neighborhood graph and the k-nearest neighbor graph. The fully connected graph suffers from high computational complexity and is thus not suitable for graphs with large size. Meanwhile, the ε-neighborhood graph is sensitive to the selected parameter ε, and suffers from the scale changes of the object. Instead, we adopt Delaunay triangulation [18] for edge generation so as to build stable graph structures invariant to scaling, translation and rotation.

For each incoming frame t, we construct a candidate graph $\mathbb{G}^t$ in the same way, and then formulate the feature correspondence problem by means of graph matching.

5.2. Candidate match filtering

There are N vertices in both model graph $\mathbb{G}^r$ and candidate graph $\mathbb{G}^t$, and thus in total N^2 candidate matches between vertices of $\mathbb{G}^r$ and $\mathbb{G}^t$. The size of the affinity matrix $K(\psi)$ is, therefore, as large as N^4, which results in high costs in not only storage space but also computational time.

In order to improve the computational efficiency, we reduce the size of $K(\psi)$ by filtering candidate matches under a reasonable continuity assumption. In particular, unreliable matches that cause leaps between consecutive frames are eliminated from the candidate match set. For an incoming frame t, we construct a candidate match set for each vertex $v_i^r \in \mathbb{V}^r$ by applying geometric and photometric constraints

$$D_i^t = \{(i,a) | \|p_a^t - \tau(\psi_{t-1}(\mathbf{p}_i^r))\|_2 \leq \epsilon_{\mathrm{g}}, \cos(f_i^r, f_a^t) \geq \epsilon_{\mathrm{a}}\},$$

where ϵ_{g} and ϵ_{a} are tolerances of geometric and appearance changes respectively. We further remove redundant matches from D_i^t and keep at most n_{c} matches with maximum appearance similarity. The final set of candidate matches D^t is constructed by combining candidate match sets over all vertices $D^t = \cup_i D_i^t$.

The constructed D^t is then used to condense the affinity matrix $K(\psi)$ by removing the corresponding row and column for each $(i,a) \notin D^t$. The size of the affinity matrix $K(\psi)$ is thus reduced to $n_{\mathrm{c}}^2 N^2$ at most. We set empirically $\epsilon_{\mathrm{g}} = 20$, $\epsilon_{\mathrm{a}} = 0.6$ and $n_{\mathrm{c}} = 5$ throughout our experiments.

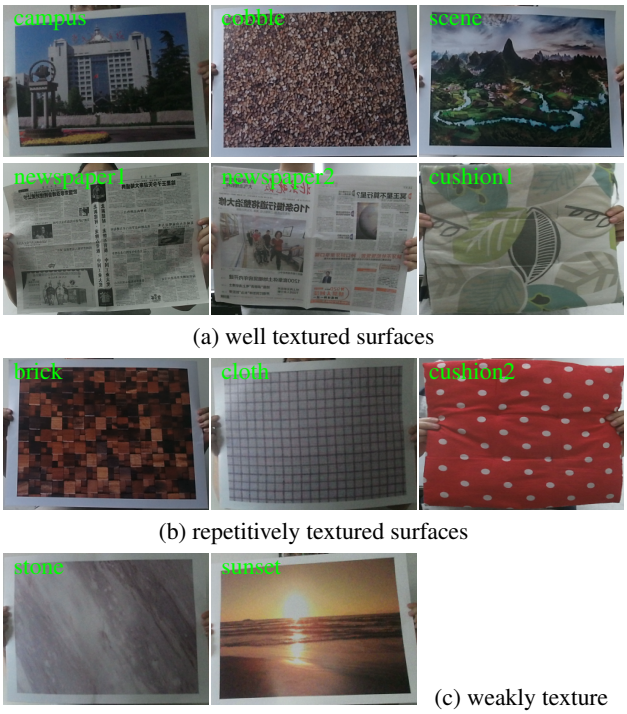
5.3. Self-adaptive outlier rejection

Our approach fuses keypoint correspondence, outlier rejection and shape deformation into a unified optimization framework as Eq. (6) that drives outlier rejection through the penalization item $\lambda \mathbf{c}^T e(\psi)$, where $\lambda > 0$ controls the degree of outlier rejection. It is usually hard to choose a proper λ for outlier rejection in practice. A too small λ cannot get the effect of denoising, while a too large one may reject many correct correspondences as outliers.

To address this issue, we propose to use self-adaptive outlier rejection by adjusting λ in line with affinity matrix $K(\psi)$ and project error $e(\psi)$ as

$$\lambda = \frac{N \sum_{i,j} |K_{i,j}(\psi)|}{|D^t| \sum_i e_i(\psi)}, \tag{12}$$

where $|D^t|$ denotes the size of the candidate match set D^t. The motivation of this self-adaptive strategy is that we choose a proper λ to avoid either of the two items in Eq. (6) dominating the optimization.



(a) well textured surfaces

(b) repetitively textured surfaces

(c) weakly texture

Figure 1. The proposed DeSurT dataset of surfaces with (a) well textured, (b) repetitive textured, and (c) weakly textured.

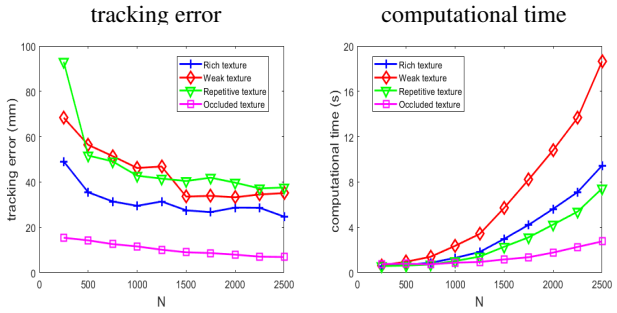
6. Experiments

Our experiments consist of two parts. The first one (Sec. 6.2) studies how the graph size affects tracking accuracy and computational time of the proposed algorithm. The second one (Sec. 6.3) compares the proposed algorithm with state-of-the-arts on two benchmarks.

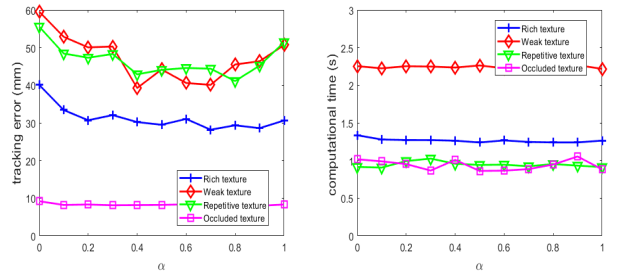
6.1. The proposed benchmark

Recently, several datasets [25, 32, 35, 36, 44] have been provided for evaluating deformable surface tracking. However, most of them lack of annotated ground-truth mesh vertices. Furthermore, these datasets are collected with limited types of surfaces and may be insufficient to evaluate the effectiveness of deformable surface tracking algorithms.

For a thorough evaluation of the proposed algorithm in comparison with the baseline algorithms, we collect a new dataset and name it *Deformable Surface Tracking* (DeSurT).



(a) the curves with respect to N



(b) the curves with respect to α

Figure 2. Tracking error and computational time of our algorithm with respect to (a) keypoint number N, and (b) balance factor α.

DeSurt is collected using a Kinect camera to evaluate tracking performance under various deformations and lighting conditions. It consists of 11 video streams and 3,361 frames showing various deformations of different types of surfaces, including seven printed pictures with different contents, two newspapers and two cushions. As illustrated in Fig. 1, these surfaces are roughly categorized into three classes: *well-textured* (campus, cobble, scene, newspaper1, newspaper2 and cushion1), *repetitively textured* (brick, cloth and cushion2), and *weakly textured* (stone and sunset).

To evaluate the reconstruction accuracy, we use the Kinect point cloud to build ground truth meshes, and compute the average vertex-to-vertex distance (as that in [25]) from the reconstructed mesh to the ground truth mesh. Therefore, in addition to the depth information of each frame, all videos come with manually annotated ground-truth mesh vertices (130 vertices for printed pictures and newspapers, and 121 vertices for cushions) across frames.

To test the robustness of the proposed algorithm against occlusion, we also report results on a public dataset, *tracking surface with occlusion* (TSO) [25], which comprises two video streams displaying well and poorly textured deformable surfaces respectively with a total of 394 frames, in presence of both artificial and realistic occlusions.

6.2. Parameter analysis

The parameter N described in Sec. 5.1 decides the number of extracted keypoints and hence the size of the graphs. It is the most crucial parameter in the proposed algorithm

Table 1. Comparison in the average tracking error (mm). The best result for each row is in bold.

	video	FSD [29]	LM [24]	LLS [8]	DIR [25]	TDA [48]	JRR [31]	ours (#N) 1000	ours (#N) 2000
DeSurT	campus	27.36	35.51	38.41	35.27	48.68	48.59	28.05	**22.02**
	brick	31.05	31.24	37.28	37.33	54.75	55.07	33.82	**27.61**
	scene	29.19	30.35	27.65	32.81	69.93	73.56	24.08	**22.19**
	cloth	298.5	247.2	361.7	175.9	92.58	98.74	71.29	**47.17**
	cobble	23.82	25.97	33.06	266.5	74.57	76.80	24.45	**22.39**
	stone	254.8	361.8	310.9	77.18	129.9	118.3	42.20	**36.09**
	sunset	85.8	117.1	94.69	44.08	76.17	74.30	51.60	**30.93**
	news.1	26.04	33.55	65.35	27.15	58.42	60.94	26.94	**23.05**
	news.2	32.99	34.15	45.39	76.34	55.73	55.78	28.84	**25.93**
	cush.1	46.16	49.45	56.38	92.49	98.93	99.68	71.08	**45.05**
	cush.2	172.0	136.2	203.2	99.18	77.20	75.91	43.73	**40.26**
	AVG	93.43	100.2	115.8	87.65	76.07	76.15	40.55	**31.15**
TSO	classrm	5.26	2.62	12.40	**2.52**	39.60	39.48	3.48	2.75
	white	38.63	49.27	58.61	7.17	60.60	60.49	8.75	**6.12**
	AVG	21.95	25.95	35.51	4.85	50.10	49.99	5.62	**4.44**

Table 2. Comparison in the average computational time (s).

	video	FDS [29]	LM [24]	LLS [8]	DIR [25]	TDA [48]	JRR [31]	ours (#N) 1000	ours (#N) 2000
DeSurT	campus	13.81	2.46	12.96	13.67	8.12	44.73	**1.02**	4.20
	brick	13.07	2.08	12.58	13.43	26.24	24.69	**0.68**	1.95
	scene	13.77	2.45	12.87	11.69	14.77	18.30	**0.80**	3.12
	cloth	9.34	2.88	13.24	14.28	8.99	8.61	**1.27**	6.46
	cobble	13.75	2.77	14.90	18.47	23.01	21.60	**2.37**	10.39
	stone	14.52	2.12	11.99	17.16	9.64	8.36	**1.92**	8.18
	sunset	12.60	**2.21**	12.38	16.27	13.80	13.25	2.41	9.55
	news.1	12.84	2.59	13.24	20.31	21.03	20.81	**1.37**	5.13
	news.2	13.24	2.54	14.89	21.78	12.94	12.74	**0.84**	3.18
	cush.1	12.68	2.42	12.59	16.07	7.31	7.90	**1.18**	3.08
	cush.2	14.19	2.32	12.10	20.80	9.58	9.51	**0.68**	2.43
	AVG	12.98	2.44	13.07	16.72	14.13	15.32	**1.32**	5.24
TSO	classrm	18.06	3.69	12.38	22.06	6.53	6.23	**1.11**	4.02
	white	12.24	1.68	12.20	52.59	24.83	24.75	**1.39**	4.32
	AVG	15.16	2.69	12.29	37.33	15.68	15.49	**1.25**	4.17

and directly affects tracking accuracy and computational time. In addition, α defined in Eq. 3 controls the degree of structure information integrated into our algorithm. In this section, we report the average tracking error and computational time with respect to N and α respectively.

As shown in Fig. 2(a), the tracking error is reduced significantly for all types of surfaces with increasing N when N is smaller than 1500, and saturate afterwards. The computational time is roughly quadratic in N because the size of the affinity matrix K is quadratic in N.

Fig. 2(b) illustrates how our algorithm is influenced by α, where $\alpha = 0$ indicates sole local appearances being used and $\alpha = 1$ means integrating fully structure information. It is shown that the tracking error is reduced remarkably for surfaces with rich, weak or repetitive texture when we fuse certain degrees of structure information (*e.g.* $0.3 \leq \alpha \leq$

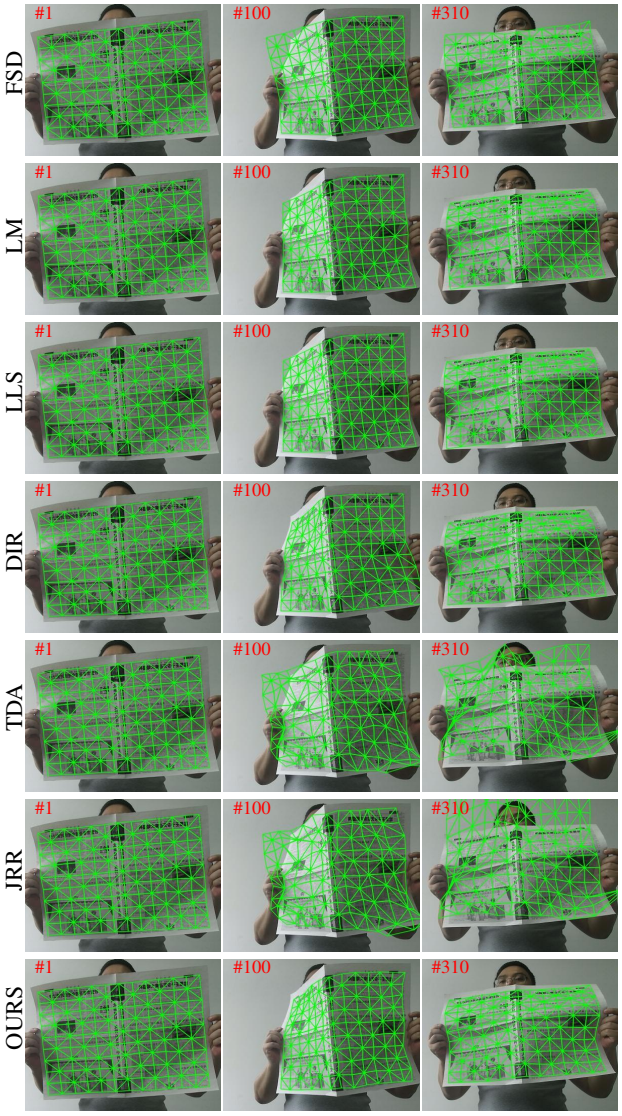


Figure 3. Examples of surface with rich texture.

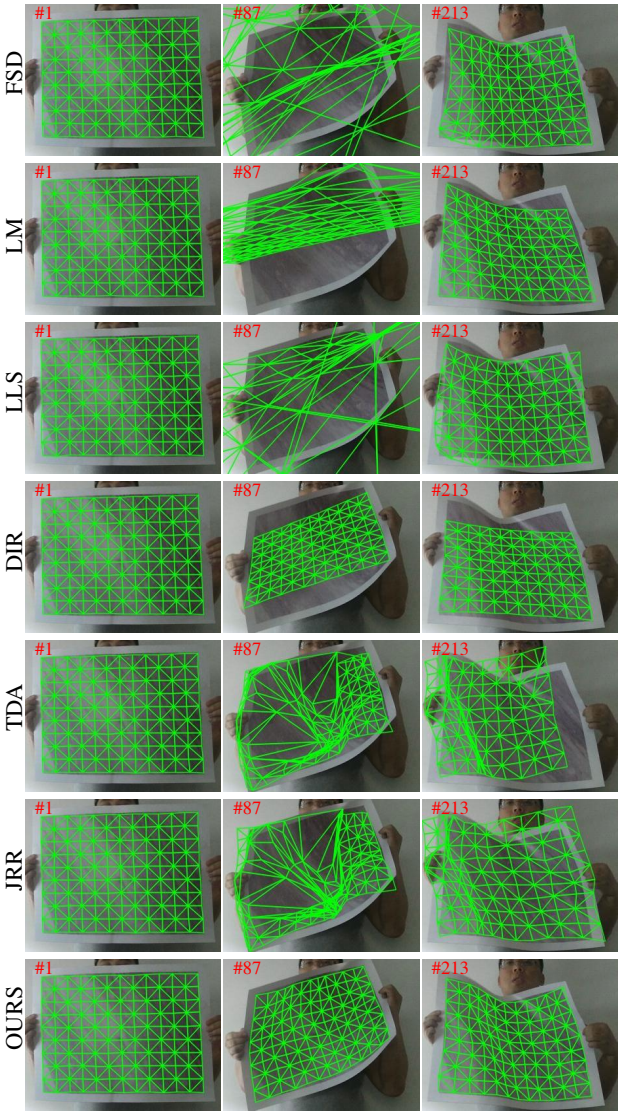


Figure 4. Examples of surface with weak texture.

0.8). Meanwhile, occluded surfaces are less benefited from the integrated structure information.

6.3. Comparison with state-of-the-art algorithms

In this section, we report experimental results of the proposed algorithm in comparison with several state-of-the-art baselines, including FSD [29], LM [24], LLS [8], DIR [25], TDA [48] and JRR [31], of which the first three algorithms are feature-based approaches and the last three ones are dense approaches. For our algorithm we fix $\alpha = 0.7$ for all trials, and report two groups of results with $N = 1000$ and 2000 respectively.

As shown in Table 1, our algorithm is robust to different types of surfaces with rich, weak or repetitive texture, and outperforms all baseline algorithms significantly even when relatively less keypoints ($N = 1000$) are extracted from each surface. As for occluded surfaces (the TSO dataset), DIR achieves satisfactory tracking results with the assistance of a well-designed strategy for occlusion detection. Interestingly, without any specified process for occluded surfaces, our algorithm provides comparable results with DIR on the TSO dataset, and outperforms other baseline algorithms in general. When we rise N up to 2000, the tracking accuracy of our algorithm is further improved remarkably on all video sequences of both datasets.

Considering computational time (Table 2), the feature-based methods (FSD, LM, LLS and ours) cost less than the dense methods (DIR, TDA and JRR). In particular, our algorithm beats not only the dense methods but also the compared feature-based ones on both datasets with $N = 1000$. When we increase the number of keypoints to 2000, our algorithm needs more computational time and becomes slower than LM, but it is still more efficient than other baseline algorithms on both datasets.

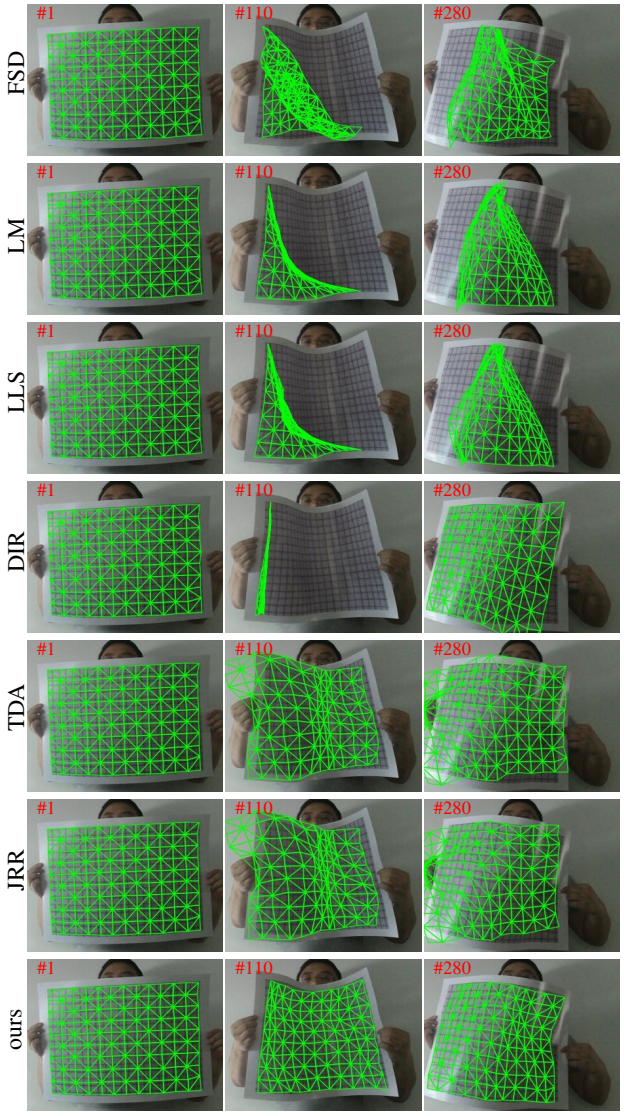


Figure 5. Examples of surface with repetitive texture.

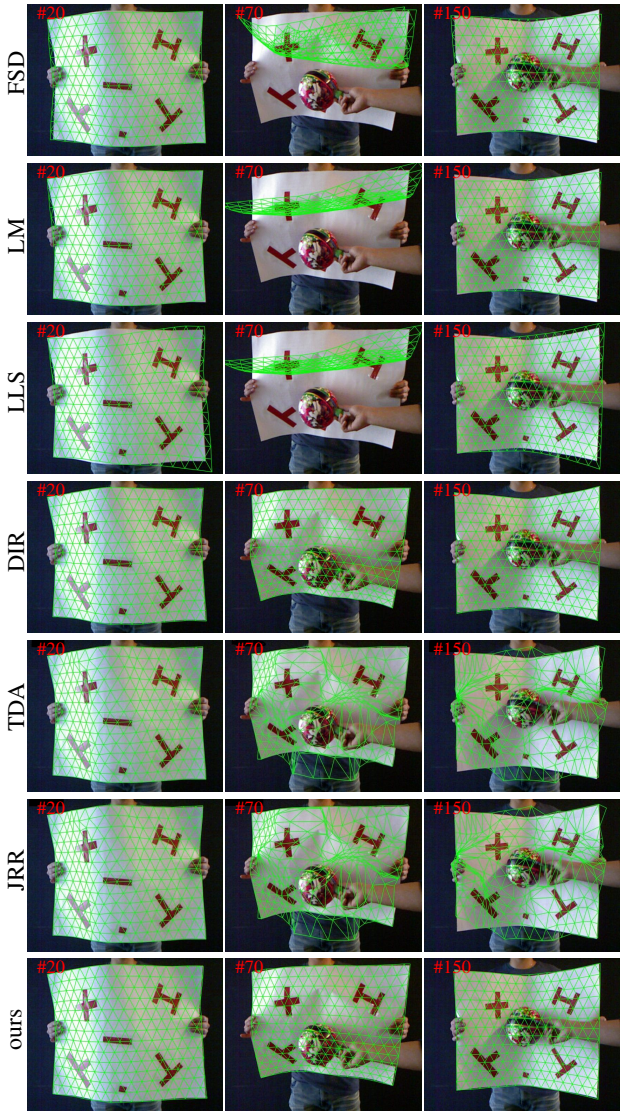


Figure 6. Examples of surface with occluded texture.

Fig. 3 to Fig. 6 illustrate several representative samples of various types of surfaces provided by the compared algorithms. For well-textured surfaces (Fig. 3), TDA and JRR fail to catch the object due to drastic deformation of the surface, while other algorithms achieve reasonable tracking results. As illustrated in Figs. 4 and 5, all the compared baseline algorithms suffer from weakly-textured and repetitively-textured surfaces, but our algorithms is able to provide accurate tracking results across frames. Furthermore, our algorithm, as well as DIR, is robust to partial occlusion (Fig. 6), while other algorithms may fail to catch the object in presence of some degree of occlusion.

7. Conclusion

In this paper, we proposed a novel graph-based approach to deformable surface tracking aiming to improve the tracking performance and efficiency. The proposed approach solves for feature correspondence and shape recovery by means of pairwise projection errors between graph structures, and employs soft matching relaxation to improve the computational efficiency. Experimental results reveal that our algorithm gains accurate and robust tracking performance against various types of surfaces and outperforms recent state-of-the-art algorithms in both accuracy and speed.

Acknowledgment. The authors thank the anonymous reviewers for their constructive suggestions and comments. This work is supported by China National Key Research and Development Plan (No. 2016YFB1001200), the National Nature Science Foundation of China (No. 61673048 and 61872032), and the Fundamental Research Funds for the Central universities (2018JBM015 and 2018JBM017).

References

[1] Antonio Agudo and Francesc Moreno-Noguer. Combining local-physical and global-statistical models for sequential deformable shape from motion. *International Journal of Computer Vision*, 122(2):371–387, 2017.

[2] Adrien Bartoli and Toby Collins. Template-based isometric deformable 3d reconstruction with sampling-based focal length self-calibration. In *CVPR*, pages 1514–1521, 2013.

[3] Adrien Bartoli, Yan Gérard, Franccois Chadebecq, and Toby Collins. On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In *CVPR*, pages 2026–2033, 2012.

[4] Adrien Bartoli, Yan Gérard, Franccois Chadebecq, Toby Collins, and Daniel Pizarro. Shape-from-template. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(10):2099–2118, 2015.

[5] Herbert Bay, Tinne Tuytelaars, and Luc J. Van Gool. SURF: speeded up robust features. In *ECCV*, pages 404–417, 2006.

[6] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187–194, 1999.

[7] Florent Brunet, Richard I. Hartley, Adrien Bartoli, Nassir Navab, and Rémy Malgouyres. Monocular template-based reconstruction of smooth and inextensible surfaces. In *ACCV*, pages 52–66, 2010.

[8] Ajad Chhatkuli, Daniel Pizarro, and Adrien Bartoli. Stable template-based isometric 3d reconstruction in all imaging conditions by linear least-squares. In *CVPR*, pages 708–715, 2014.

[9] Ajad Chhatkuli, Daniel Pizarro, Adrien Bartoli, and Toby Collins. A stable analytical framework for isometric shape-from-template by surface integration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(5):833–850, 2017.

[10] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted random walks for graph matching. In *ECCV*, pages 492–505, 2010.

[11] Toby Collins and Adrien Bartoli. Using isometry to classify correct/incorrect 3d-2d correspondences. In *ECCV*, pages 325–340, 2014.

[12] Toby Collins and Adrien Bartoli. Realtime shape-from-template: System and applications. In *ISMAR*, pages 116–119, 2015.

[13] Toby Collins, Pablo Mesejo, and Adrien Bartoli. An analysis of errors in graph-based keypoint matching and proposed solutions. In *ECCV*, pages 138–153, 2014.

[14] Miodrag Dimitrijevic, Slobodan Ilic, and Pascal Fua. Accurate face models from uncalibrated and ill-lit video sequences. In *CVPR*, pages 1034–1041, 2004.

[15] Ady Ecker, Allan D. Jepson, and Kiriakos N. Kutulakos. Semidefinite programming heuristics for surface reconstruction ambiguities. In *ECCV*, pages 127–140, 2008.

[16] Gareth J. Edwards, Timothy F. Cootes, and Christopher J. Taylor. Face recognition using active appearance models. In *ECCV*, pages 581–595, 1998.

[17] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–100, 1956.

[18] Der-Tsai Lee and Bruce J. Schachter. Two algorithms for constructing a delaunay triangulation. *Int. J. Computer Information Sci*, 9:219–242, 1980.

[19] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, pages 1482–1489, 2005.

[20] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. BRISK: binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, 2011.

[21] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[22] Francesc Moreno-Noguer, Josep M. Porta, and Pascal Fua. Exploring ambiguities for monocular non-rigid shape estimation. In *ECCV*, pages 370–383, 2010.

[23] Francesc Moreno-Noguer, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Capturing 3d stretchable surfaces from single images in closed form. In *CVPR*, pages 1842–1849, 2009.

[24] Dat Tien Ngo, Jonas Östlund, and Pascal Fua. Template-based monocular 3d shape recovery using laplacian meshes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1):172–187, 2016.

[25] Dat Tien Ngo, Sanghyuk Park, Anne Jorstad, Alberto Crivellaro, Chang Dong Yoo, and Pascal Fua. Dense image registration and deformable surface reconstruction in presence of occlusions and minimal texture. In *ICCV*, pages 2273–2281, 2015.

[26] Jonas Östlund, Aydin Varol, Dat Tien Ngo, and Pascal Fua. Laplacian meshes for monocular 3d shape recovery. In *ECCV*, pages 412–425, 2012.

[27] Shaifali Parashar, Daniel Pizarro, and Adrien Bartoli. Isometric non-rigid shape-from-motion with riemannian geometry solved in linear time. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(10):2442–2454, 2018.

[28] Julien Pilet, Vincent Lepetit, and Pascal Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, 2008.

[29] Daniel Pizarro and Adrien Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 97(1):54–70, 2012.

[30] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):105–119, 2010.

[31] Chris Russell, Lourdes Agapito, Rui Yu, Andrew W. Fitzgibbon, and Liu-Yin Qi. Better together: Joint reasoning for non-rigid 3d reconstruction with specularities and shading. In *BMVC*, 2016.

[32] Mathieu Salzmann and Pascal Fua. Reconstructing sharply folding surfaces: A convex formulation. In *CVPR*, pages 1054–1061, 2009.

[33] Mathieu Salzmann and Pascal Fua. *Deformable Surface 3D Reconstruction from Monocular Images*. Synthesis Lectures on Computer Vision. Morgan & Claypool Publishers, 2010.

[34] Mathieu Salzmann and Pascal Fua. Linear local models for monocular reconstruction of deformable surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):931–944, 2011.

[35] Mathieu Salzmann, Richard I. Hartley, and Pascal Fua. Convex optimization for deformable surface 3-d tracking. In *ICCV*, pages 1–8, 2007.

[36] Mathieu Salzmann, Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Closed-form solution to non-rigid 3d surface registration. In *ECCV*, pages 581–594, 2008.

[37] Jordi Sanchez-Riera, Jonas Östlund, Pascal Fua, and Francesc Moreno-Noguer. Simultaneous pose, correspondence and non-rigid shape. In *CVPR*, pages 1189–1196, 2010.

[38] Appu Shaji, Aydin Varol, Lorenzo Torresani, and Pascal Fua. Simultaneous point matching and 3d deformable surface reconstruction. In *CVPR*, pages 1221–1228, 2010.

[39] Shuhan Shen, Wenhuan Shi, and Yuncai Liu. Monocular template-based tracking of inextensible deformable surfaces under L_2-norm. In *ACCV*, pages 214–223, 2009.

[40] Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Didier Stricker. Ismo-gan: Adversarial learning for monocular non-rigid 3d reconstruction. In *CVPRW*, pages 138–153, 2019.

[41] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and Hans-Peter Seidel. Laplacian surface editing. In *Second Eurographics Symposium on Geometry Processing*, pages 175–184, 2004.

[42] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, pages 596–609, 2008.

[43] Quoc-Huy Tran, Tat-Jun Chin, Gustavo Carneiro, Michael S. Brown, and David Suter. In defence of RANSAC for outlier rejection in deformable registration. In *ECCV*, pages 274–287, 2012.

[44] Aydin Varol, Mathieu Salzmann, Pascal Fua, and Raquel Urtasun. A constrained latent variable model. In *CVPR*, pages 2248–2255, 2012.

[45] Sara Vicente and Lourdes Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. In *ECCV*, pages 426–440, 2012.

[46] Tao Wang and Haibin Ling. Gracker: A graph-based planar object tracker. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1494–1501, 2018.

[47] Tao Wang, Haibin Ling, Congyan Lang, and Songhe Feng. Graph matching with adaptive and branching path following. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2853–2867, 2018.

[48] Rui Yu, Chris Russell, Neill D. F. Campbell, and Lourdes Agapito. Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from RGB video. In *ICCV*, pages 918–926, 2015.

[49] Feng Zhou and Fernando De la Torre. Deformable graph matching. In *CVPR*, pages 2922–2929, 2013.