Monocular 3D Tracking of Deformable Surfaces*†

Luis Puig[1] and Kostas Daniilidis[2]

Abstract— **The problem of reconstructing deformable 3D surfaces has been studied in the non-rigid structure from motion context, where either tracked points over long sequences or an initial 3D shape are required, and also with piecewise methods, where the deformable surface is modeled as a triangulated mesh, which is fitted to an initial estimation of the 3D surface computed from correspondences in two views.**

In this paper we present a new scheme to reconstruct deformable surfaces by tracking relevant features that parametrize such deformation. Assuming that an initial 3D shape related to a reference frame is available, we initially match the reference and current frames using visual information. Then, these correspondences are clustered in patches with geometric characteristics in the image domain and 3D space. In order to reduce the number of parameters to be estimated, we explain each cluster using thin-plate splines (TPS) with a minimal number of control points. Then the 3D coordinates of these control points in the deformed surface are estimated using a non-linear least squares approach, deriving on the reconstruction of the full deformed patches. We perform experiments in synthetic and real data of monocular video sequences to validate our approach.

I. Introduction

The reconstruction of deformable objects and the camera pose from a monocular video sequence is an intrinsically ill-posed problem. Several approaches have been developed to deal with this problem. These approaches, known as non-rigid structure from motion [1], [2], [3], [4], either require an initial model of the 3D shape or tracked features along the full video sequence. They assume that the shape of a non-rigid object can be expressed in a compact way as a linear combination of an unknown low-rank shape basis. In sequential approaches the initial 3D shape is computed from few frames of the video sequence using rigid structure from motion approaches. After this shape the fixed low-rank basis are computed. The successive deformed shapes are represented as the sum of the shape at rest previously computed and a linear combination of those basis. Moreover, these approaches assume a simple orthographic projection model. When the deformable object is actually a deformable surface, a different type of methods called piecewise methods [5], [6], [7], have been developed. These approaches parametrize the surface as a triangulated mesh and compute the initial reconstruction of the surface from at least two views. This parametrization simplifies the degrees of freedom of the mesh vertices and allows to represent the deformation of the surface using much simpler local deformation models.

In this work we focus on scenarios where the following challenges arise: i) the object of interest can occupy a small portion of the image; ii) several objects are observed in a single frame and they may overlap; iii) the projection model is more complex than the orthographic one. Under these conditions most of the previously approaches cannot be used.

In this paper we present a new approach that tracks the deformation undergone by the surface in subsequent frames. We assume that an initial shape at rest of the object of interest is available. Similarly to piecewise approaches that simplify the deformation expected by parameterizing the surface as a triangulated mesh, we subdivide the deformable surface in smaller patches using visual and depth information. Then, in order to further reduce the number of points to represent the surface, a minimum number of points are selected as control points of thin-plate splines. These splines have been widely used as warps of deformable surfaces [8], [9].

Our approach consists of two main steps: 1) From an initial reconstruction of the object and image correspondences we segment the image based on texture, 2D image geometry, and depth; and 2) Estimation of the deformed surfaces using thin-plate splines.

The main contributions of this paper are the following: 1) Simplification of the 3D surfaces by subdivision in smaller clusters that geometrically relate image correspondences; 2) We do not require to compute a set of basis, the clusters are represented using thin-plate splines; and 3) a full perspective projection model is integrated in the problem formulation.

A. Related work

number of approaches devoted to the reconstruction of a single deformable object observed in a sequence of images is considerable. The underlying principle behind most approaches is to model the time-varying shape as a linear combination of an unknown low-rank shape basis. Such basis can be estimated from an initial reconstruction of the object at rest [1], [2], [3], [4] or learned from previously observed examples [5], [10]. Depending on the number of frames processed at a time these approaches can be classified in either batch or sequential approaches. The batch methods assume that a sequence of tracked points of the object is available and process all this information at once [1], [2], [11], [10], obtaining the 3D representation of the deformable object at each frame. The sequential methods estimate a new

[1]Dept. of Computer Science & Engineering, University of Washington, Seattle, USA lpuig@cs.washington.edu

[2]GRASP Laboratory, University of Pennsylvania, 3330 Walnut Street, L402, Philadelphia, USA kostas@seas.upenn.edu

*This work was developed during Luis Puig's postdoctoral fellowship at GRASP Laboratory, University of Pennsylvania.

†The authors thank the following grants: NSF-DGE-0966142, NSF-IIP-1439681, NSF-IIS-1426840, ARL MAST-CTA W911NF-08-2-0004, ARL RCTA W911NF-10-2-0016, and CONACYT.

shape for each new acquired frame. These methods compute an initial 3D representation of the object, using standard structure from motion techniques, from which physical-based basis [4] or simply a low-rank shape model basis [3] are computed. Then the deformed shape is computed for each new acquired frame. Both approaches, batch and iterative assume that a single object is observed on the scene and that enough correspondences are observed in consecutive frames. In contrast to global non-rigid structure from motion (NRSfM) approaches that either require 3D points to be observed over a large number of frames or an initial 3D shape, piecewise approaches are able to reconstruct deformable surfaces from correspondences between pairs of frames [6], [7]. This type of algorithms are more suitable for deformable surfaces than those developed for generic deformable objects. These methods represent the surface as a predefined triangulated mesh, where the surface 3D points, initially reconstructed from correspondences in two views [12], are represented as its vertices. The goal is to compute the deformation of the mesh that best fit the 3D points. This local formulation allows the simplification of the expected deformations, since local patches have fewer degrees of freedom and can only undergo relatively small deformations, making them easier to learn [5]. One key aspect of these approaches is the partitioning of the mesh. Instead of dealing with the deformable surface as a whole, it is modeled as a combination of smaller patches with common shared features to enforce global consistency. Our approach has been inspired by these methods.

II. Our approach

In this section we present the proposed approach to reconstruct patches of deformable surfaces. Given an initial reconstruction of the deformable surface, our approach: 1) segments the scene based on texture, 2D image geometry, and depth; and 2) estimates of the camera motion and the deformed surfaces using thin-plate splines.

A. Initial 3D Reconstruction

As an initial step we need to recover the 3D structure of the environment. NRSfM approaches reconstruct the scene from multiple frames using conventional SfM techniques while piecewise approaches use planar homographies between two views to recover this structure. Both of these approaches compute sparse features and depend on a well texturized environment. Another technique is shape from shading (SfS), which do not require a well texturized environment and provide a dense 3D reconstruction using a single calibrated frame [13]. The main disadvantage of these techniques is the computation time. In this paper we will explore different reconstruction methods. In Fig. 1 we show examples of reconstructions using SfS and stereo techniques

B. Image Segmentation Using Affinities and Depth

Assuming we have an initial reconstruction of the scene, we subdivide the image into smaller patches based on the correspondences between two consecutive frames and depth information. We initially extract and match SIFT

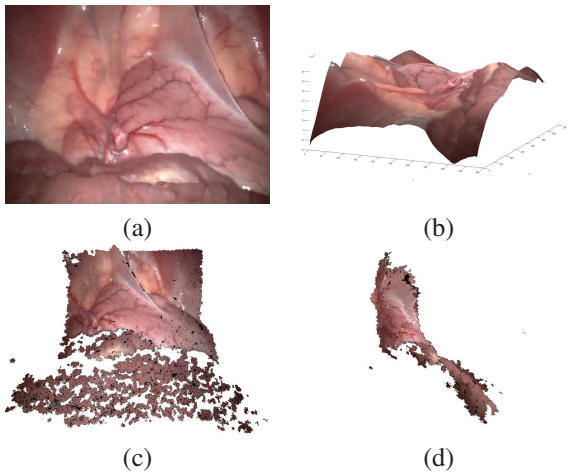


Fig. 1. Examples of 3D reconstruction from laparoscopic images using shape from shading and stereo techniques. (a) Color image of porcine wall. (b) Reconstruction using SfS. (c) Front view of stereo reconstruction. (d) Lateral view of stereo reconstruction.

features [14]. Since no global geometrical constraints can be used due to the deformation of the tissue, an inlier selection strategy is used to eliminate outliers. In particular, we use an approximation to the maximum clique[15], where the distance between matches and feature orientations are used as consistency criteria. These inlier correspondences are the input to our correspondence-based segmentation algorithm, which is inspired by [16], with the difference that our approach combines the local geometric information in the image domain with depth information. Moreover, our approach focuses on the cluster construction instead of the matching, since an already outlier-free set of matches is provided. The main steps of our algorithm are the following. We initially apply the Delaunay triangulation over the matched features in the reference image. In the next step, we randomly select a triangle for which we compute the corresponding affine transformation using its three correspondences in the current image. Then, we map the adjacent vertices of the triangle from the reference to the current image using the computed affine transformation. In order to include a match as a member of a cluster we verify two criteria: i) the distance between the mapped point and the actual match in the current image is smaller that a threshold, and ii) the depth of the adjacent vertex with respect to the cluster's depth is inside a predefined range. Every time a new match is added to the cluster, the affine transformation is updated. This process is repeated until no more adjacent vertices can be added. Then, the whole process is repeated with a new randomly selected triangle until the number of unclustered triangles is smaller than the predefined cluster size. This algorithm is easily implemented using recursion over a tree structure. The output of our algorithm is a set of clusters, which represent consistent elements in the image domain and 3D space. Moreover, each cluster is assigned an affine transformation that accurately (up to a predefined threshold) maps a point contained in the reference cluster to its corresponding point in the current image. The pseudo-code of this procedure is depicted in Algorithm 1. In Fig. 2 we show an example of the

Algorithm 1: Correspondences clustering based on affine transforms and depth information.

Input : $\Omega = \{(\mathbf{p}_1, \mathbf{q}_1), \ldots, (\mathbf{p}_n, \mathbf{q}_n)\}$ (correspondences)
$\bar{\mathsf{S}}$ (Initial shape)
Output: $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$ (Set of affine clusters)
*/*Compute Delaunay Triangulation*/*
triangles ← `Delaunay(`Ω`)`
triangle ← `random(`triangles`)` */*Select random triangle*/*
*/*Define maximum cluster depth*/*
thrDepth ← $max\_cluster\_depth$
thrDist ← max_affine_error
while triangles.size() $> min\_cluster\_size$ **do**
  $\mathcal{T}^i_{affine}$ ← `ComputeAffinity(`triangle, Ω`)`
  $\mathcal{A}_i$ ← `InitializeCluster(` $\mathcal{T}_{affine}$, triangle`)`
  nodeID ← 1 */*New tree has only root node*/*
  $\mathcal{A}_i$ ← `ExpandCluster(`$\mathcal{A}_i$, nodeID, Ω, $\bar{\mathsf{S}}$`)`
  triangles ← `RemoveUsedTriangles(`triangles`)`
  $i \leftarrow i + 1$
  triangle ← `random(`triangles`)`
  return $\mathcal{A}$
end

Function `ExpandCluster(`$\mathcal{A}_i$, nodeID, Ω ,$\bar{\mathsf{S}}$`)`

```
leafNode ← A_i(nodeID).isLeaf
if leafNode == false then
    [A_i, adjNodes ] ←GetAdjNodes(A_i, nodeID, S̄)
    if isEmpty(adjNodes) then
        A_i(nodeID).isLeaf ← true
    else
        numAdjNodes ← adjNodes.size()
        for j ← 1 to numAdjNodes do
            nodeID ← adjNodes(j).ID
            valDepth←VerifyDepth(nodeID)
            affDist ← CalcDistance(T^i_aff,
            nodeID)
            if valDepth and affDist < thrDist then
                A_i ← AddNode(nodeID)
                T^i_aff ← UpdateAff( T^i_aff,nodeID)
            end
            A_i ← ExpandCluster(A_i, nodeID, S̄)
        end
    end
end
return A_i
```

end

segmentation using our approach. The scene presents drastic illumination changes and considerable camera motion. We observe that the clusters represent consistent structures in the image and 3D domain.

C. Modeling Patches Deformation

In this work we model the patches deformation using thin-plate spline mappings. These mappings can be seen as a combination of a linear part (affine transformation) and the superimposition of *principal warps* [17], which are basis for

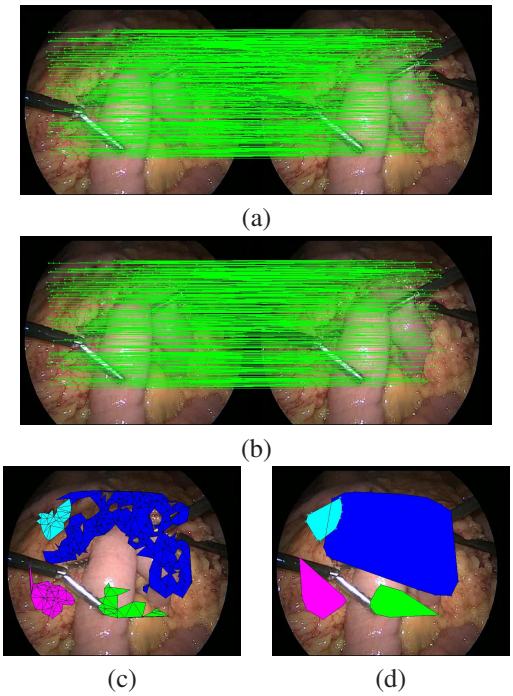


Fig. 2. Image segmentation based on correspondences. (a) SIFT features are matched in two consecutive frames. (b) Inlier selection using maximum clique approximation. (c) Matches are classified into clusters based on geometric and depth information. (d) Convex hull of clusters in current image.

the representation of shape change. More formally, a thin-plate mapping $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ for a point $\mathbf{x} = (x, y)$, is defined by a radial base function (RBF) $U(\mathbf{s}) = \mathbf{s}^2 \log(\mathbf{s}^2)$, a 3-vector $\mathbf{r} = (r_1, r_2, r_3)$ (affine transformation) and a n-vector $\mathbf{t} = (w_1, \ldots, w_n)$ defining non-affine transformations, which are associated to a set of n control points $\mathbf{c} = (\bar{x}, \bar{y})$, such that $f(\mathbf{x}) = r_1 + r_2 x + r_3 y + \sum_{i=1}^{n} w_i U(\|\mathbf{c}_i - \mathbf{x}\|)$. In order to consider a three dimensional mapping $\mathcal{W} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, we stack three RBFs f^x, f^y and f^z sharing their control points

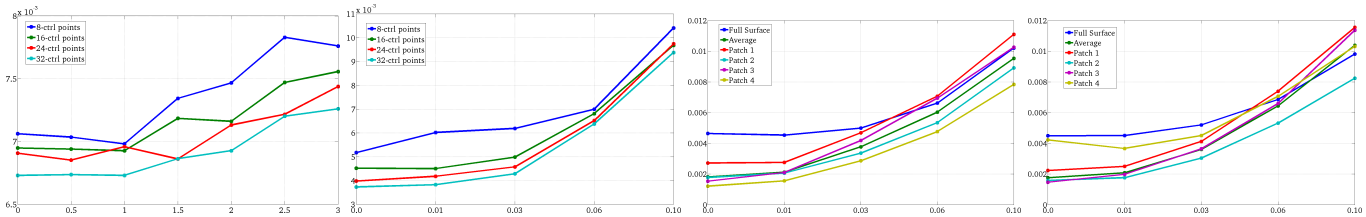
$$\mathcal{W}(\mathbf{x}) = \begin{bmatrix} r_2^x & r_3^x & r_1^x \\ r_2^y & r_3^y & r_1^y \\ r_2^z & r_3^z & r_1^z \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \sum_{i=1}^{n} \begin{bmatrix} w_i^x \\ w_i^y \\ w_i^z \end{bmatrix} U(\|\mathbf{c}_i - \mathbf{x}\|), \tag{1}$$

with $\mathbf{c}_i = (s\bar{x}, s\bar{y}, s)$ and $\mathbf{x} = (sx, sy, s)$ in homogeneous coordinates. The parameters that define f^x, f^y and f^z can be estimated by solving a linear system that relates the control points $\mathbf{c}$ and their correspondences $\mathbf{c}' = \mathcal{W}(\mathbf{c})$. Let P_c and P'_c be the stacked correspondences of the control points in normalized coordinates in the reference and current image, respectively. We compute the thin-plate parameters using

$$\left[\begin{array}{c|ccc} \mathbf{t}_x & r_3^x & r_2^x & r_1^x \\ \mathbf{t}_y & r_3^y & r_2^y & r_1^y \\ \mathbf{t}_z & r_3^z & r_2^z & r_1^z \end{array}\right] = \mathsf{L}^{-1} \begin{bmatrix} \mathsf{P}'_c \\ 0 \end{bmatrix}; \mathsf{L} = \begin{bmatrix} \mathsf{U} & \mathsf{P}_c \\ \mathsf{P}_c^{\mathsf{T}} & 0 \end{bmatrix} \tag{2}$$

where $\mathsf{U}_{ij} = U(\|\mathbf{c}_j - \mathbf{c}_i\|)$ and 0 is a 3×3 zero matrix. The *principal wraps* of this mapping are given by the eigenvectors of the *bending energy matrix* $\mathsf{L}_n^{-1}\mathsf{U}\mathsf{L}_n^{-1}$, with L_n^{-1} the upper left $n \times n$ sub-block of L^{-1}.

In order to map features from the reference to the current frame we use the feature driven parametrization of TPS. The



(a) (b) (c) (d)

Fig. 3. (a) RMSE of the reconstructed point cloud as a function of noise in the pixel coordinates in the reference and current images. (b) RMSE of the reconstructed point cloud as a function of noise in the initial point cloud. (c) RMSE of the reconstructed point cloud as a function of noise on the initial 3D coordinates using patches. (d) RMSE of the reconstructed point cloud as a function of noise on the initial point cloud including 10% of outliers.

mapped points $\mathbf{x}' = (x', y')$ of the reference image can be calculated as a function of control point correspondences $\mathbf{c}' = (\bar{x}', \bar{y}')$ on the current image. The m transformed pixel coordinates $\mathbf{x}'_j$ can be stacked in a matrix P'_x such that $\mathsf{P}'_x = [\mathbf{x}'_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]^\mathsf{T}$

$$\mathsf{P}'_x = [\mathsf{V}\ \mathsf{W}]\mathsf{L}_* \mathsf{P}'_c, \tag{3}$$

where $\mathsf{V}_{ij} = \mathsf{U}(\|\mathbf{x}_j - \mathbf{c}_i\|)$, $\mathsf{W}_j = (1, x_j, y_j)$ and L_* is the $(n+3) \times n$ sub-matrix of L. Notice that the matrices V, W and L_* only depend on pixels coordinates in the reference image.

This feature driven mapping allows us to represent the control point correspondences as projections of the 3D points X as $\mathsf{P}'_c = \mathsf{K}\mathsf{X}$ using the 3×4 perspective projection matrix K. We exploit this representation to define the warping

$$\omega(\mathbf{x}_i, \mathbf{X}_i, \mathsf{K}) = [sx_i\ sy_i\ s] = \mathsf{M}_i \mathsf{L}_* \mathsf{K} \mathbf{X}_i, \tag{4}$$

where M_i is the $i-$row of the matrix M corresponding to $\mathbf{x}_i$.

1) Mapping Patches: A patch $\mathcal{A}_i$ is defined by a set of correspondences $\mathbf{q} \leftrightarrow \mathbf{q}'$, which represent points in the reference and current image, respectively. These correspondences are the projections of 3D points $\mathbf{X}_r$ and $\mathbf{X}_c$, respectively. Using the previously defined warp, the correspondences in the current frame can be defined as

$$\mathbf{q}' = \omega(\mathbf{q}, \mathbf{X}_c, \mathsf{K}). \tag{5}$$

In order to compute the deformed point cloud that projects the image correspondences $\mathbf{q}'$ we minimize the following cost function

$$\underset{\mathsf{X}}{\text{minimize}} \sum_{\mathbf{q} \in \mathcal{A}_i} \|\omega(\mathbf{q}, \mathsf{X}, \mathsf{K}) - \mathbf{q}'\|^2 + \lambda \cdot \|\mathsf{X}_p - \mathsf{X}\|^2, \tag{6}$$

where λ is a regularizer that penalizes strong 3D point variations. Since all patches $\mathcal{A}_1, \ldots, \mathcal{A}_N$, share the same projection matrix we define the overall cost function as

$$\underset{\mathsf{X}_1, \ldots, \mathsf{X}_{n_p}}{\text{minimize}} = \sum_{i=1}^{N} \sum_{\mathbf{q} \in \mathcal{A}_i} \|\omega(\mathbf{q}, \mathsf{X}_i, \mathsf{K}) - \mathbf{q}'\|^2 + \lambda \cdot \|\mathsf{X}_{pi} - \mathsf{X}_i\|^2 \tag{7}$$

To solve the minimization problem above we use the Levenberg-Marquardt algorithm. Notice that we minimize the reprojection error of the control points X, which are a subset of the points contained in all clusters. The reprojection error reported on the experiments is computed using all the points contained in the clusters.

D. Selection of Control Points

The control points of thin-plate splines define the accuracy of the surface reconstruction. We found that $\sim 30\%$ of the points contained in each cluster are enough to have a fair reconstruction of the surface's patches. If the patches share points, the common points are selected as control points automatically in order to keep the patches together.

III. Experiments

We apply the previous approach in simulated and real environments to analyze the performance of our approach. In particular we are interested in the advantages that the cluster segmentation provides to the estimation of the surface deformation.

A. Synthetic Data

We analyze the sensitivity of our approach to noise and the impact of the number of control points on the accuracy of the deformed surface. We create a mesh of $20cm \times 20cm$ with 121 vertices, which depth is defined by a third order polynomial on their (X, Y) coordinates. We compute the deformed surface X_{def} as $\mathsf{X}_{def} = \Delta \cdot \mathsf{X} + \Gamma$,where Δ contains scale factors and Γ contains translation vectors for each element in the point cloud X. We apply a scale factor up to ten percent to the X and Y coordinates (Δ_x, Δ_y) and up to two percent to the Z coordinate (Δ_z). We also add a random translation $(\Gamma_x, \Gamma_y, \Gamma_z)$ also up to ten percent. We generate the reference and current images after these two surfaces using a perspective projection. We add Gaussian noise, characterized by its standard deviation σ, to the pixel coordinates of the control points in both, reference and current images. We repeat the experiments 30 times in order to avoid particular cases due to random noise. In Fig. 3(a) we see the root mean square error (RMSE) of the 3D points as a function of noise.

In the next experiment we analyze the sensitivity of our approach to a bad initial estimation of the point cloud. We randomly perturbed the initial estimation of the point cloud by scaling and translating the 3D coordinates. The percentages considered are $\mathcal{P} = (0.0, 0.01, 0.03, 0.05, 0.10)$ for Δ_x, Δ_y and Γ, and $\mathcal{P}/5$ for Δ_z. Pixel coordinate noise of $\sigma = 1$ is added to the feature correspondences. In Fig. 3(b) we observe that the our approach is robust up to ten percent of noise with less than 1cm error.

As commented previously our approach reconstructs small patches. This characteristic makes our algorithm less sensitive to noise and robust to outliers. In the following experiment we divide the surface into four patches with similar area. We reconstruct the full surface and each individual patch using sixteen control points with feature correspondence error of $\sigma = 1$. In Fig. 3(c) we observe

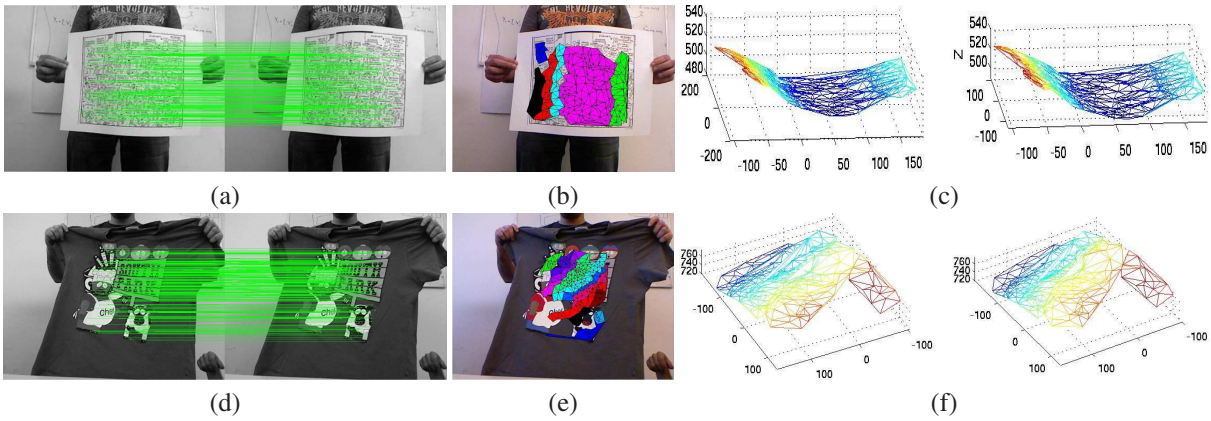


Fig. 4. Reconstruction of deformable surface by tracking relevant features. First row, paper sequence. Second row, t-shirt sequence. Third row, laparoscopic sequence (a,d) Matches after inlier selection. (b,e) Computed clusters based on geometric information and affine transformations. (c,f) Reconstruction by tracking, first plot shows ground truth and second plot shows our reconstruction.

the reconstruction error for different amounts of noise in the initial point cloud. In Fig. 3(d) we observe the impact of outliers on the surface reconstruction. We used the same configuration as the previous experiment and we add ten percent of outliers to the control points used for the full reconstruction of the surface and to the control points of a single patch (Patch 4).

The following observations arise from the experiments:

- Our approach has a low sensitivity to noise in the pixel coordinates of the detected control points.
- Using only eight control points, the estimation error of the whole deformed surface is very close to the one using four times more control
- Single patch reconstruction errors are lower than the full reconstruction error for small amounts of noise in the image coordinates.
- Outliers have a bigger impact on smaller patch areas.
- The use of patches provides a way to isolate outliers.

B. Real data

In the next experiments we test the performance of our approach using three sequences of real data. Two of these sequences were acquired with an RGB-D sensor providing ground truth [18]. The first sequence corresponds to a paper observing different amounts of bending and contains 193 frames. The second sequence contains 313 frames and corresponds to a T-shirt, which observes a more complex deformation. The third sequence used is a stereo laparoscopic video[1]. The image resolution of all sequences is 640×480. We track the deformation of the surface between two consecutive frames. We assume that the initial reference surface is known. We compute and match SIFT features using VLFeat Matlab implementation [19]. Then we compute an approximation to the maximum clique, using the distance and orientation between matches as consistency criteria, as an inlier selection [15]. These inliers along with the initial reconstruction are the input to our algorithm. The number of control points of the TSP depend on the total number of features contained in each patch.

[1]http://hamlyn.doc.ic.ac.uk/vision/

TABLE I
STATISTIC INFORMATION OF SINGLE FRAME RECONSTRUCTION

Sequence	Matches	Clusters	Points per cluster	Error
Paper	324	6	[41,66,166,15,26,12]	2.5mm
T-shirt	364	13	[67,25,54,20,15,17, 14,36,18,36,12,24,26]	2.8mm
Laparoscopy1	828	8	[84,163,384,53,133,62, 81,32]	1.5%
Laparoscopy2	828	8	[219,94,250,100,132,] 82,40,99]	1.2%

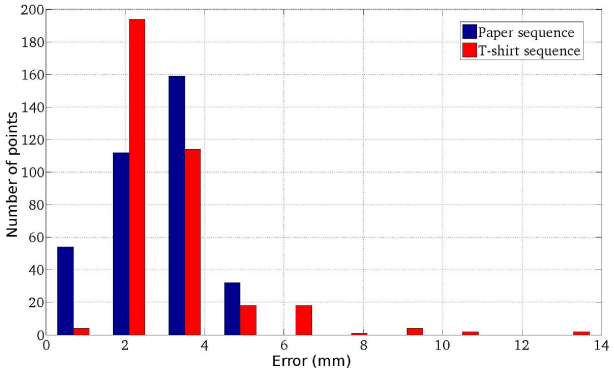


Fig. 5. Error distribution in millimeters for the two reconstructed frames presented in Fig. 4.

1) RGB-D Sequences: In Fig. 4 we present results for single frames of the RGB-D sequences. The initial matches after the inlier selection step are shown in Fig. 4(a,d). The computed clusters using 2D and 3D information are shown in Fig. 4(b,e). We clearly observe how clusters capture areas of the image where local affine transforms relate features in the reference and current images, as well as the impact of the depth information on the bending areas. In Fig. 4(b) the biggest cluster occupies an almost flat area with similar depth. Adjacent clusters are thinner since the change in depth, due to the bending of the paper, is more drastic. In Fig. 4(c,f) we show the reconstruction of the deformed surface computed by our approach. The first surface corresponds to the ground truth and the second one is the reconstructed surface. We observe that the shared points connecting contiguous clusters enforce a smooth transition

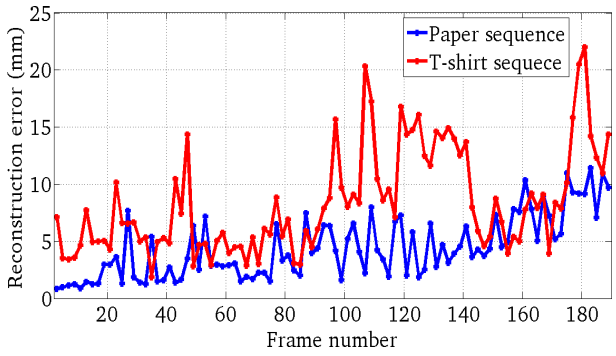


Fig. 6. 3D reconstruction error of the first 190 frames of the RGB-D sequences.

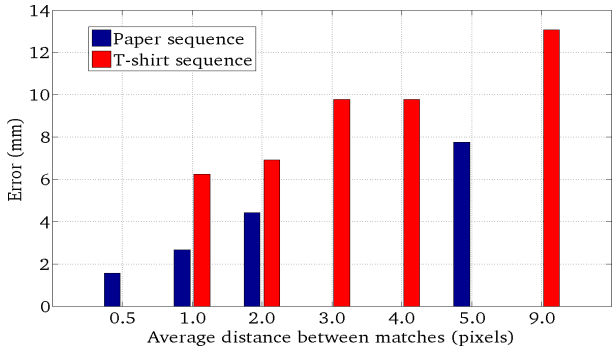


Fig. 8. 3D reconstruction error with respect to the average pixel distance between feature matches using all frames in the sequences.

between clusters. The reconstruction error is $2.5mm$ and $3.5mm$, in the paper and the T-shirt examples, respectively. In Fig. 5 we show the error distribution for both examples. We observe that most reconstructed points have errors of one and two millimeters. Notice that since we do not parametrize the surface as a triangular mesh, the error is computed as the Euclidean distance between the reconstructed 3D points contained in the clusters and the ground truth 3D points. In Table I we observe that the number of points used to track the deformation of the surface is reduced by the use of TPS.

In Fig. 6 we present the reconstruction error for the first 150 frames contained in both sequences. We observe that the errors on the T-shirt sequence are much bigger than those of the paper sequence. This behavior is explained due to the different smoothness of the surfaces (see Fig. 7). Another factor that affects the surface reconstruction is the feature displacement in the image domain. In order to analyze this impact we group the matched images according to their feature displacement. In Fig. 8 we show the 3D reconstruction error as a function of the features displacement. In both sequences we clearly observe that the reconstruction error is strongly related to the feature displacement from frame to frame.

2) Laparoscopic Sequence: In this experiment we use a video sequence acquired with a stereo laparoscope. We reconstruct the scene using two algorithms as shown if Fig. 1. The first algorithm is shape from shading provided by [13] and requires a single calibrated image. This reconstruction is dense, smooth and up to scale. The second reconstruction

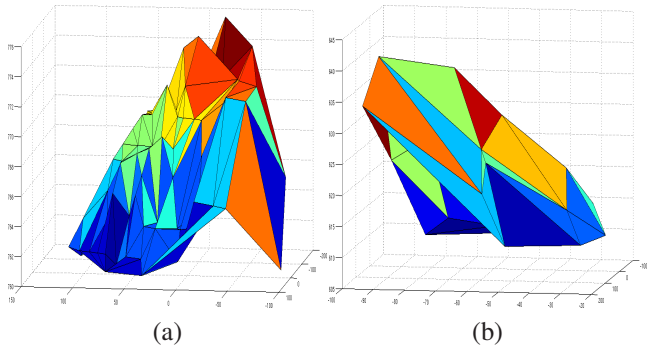


Fig. 7. Surface smoothness. (a) T-shirt surface has abrupt changes between adjacent nodes. (b) Paper surface is smoother than t-shirt surface.

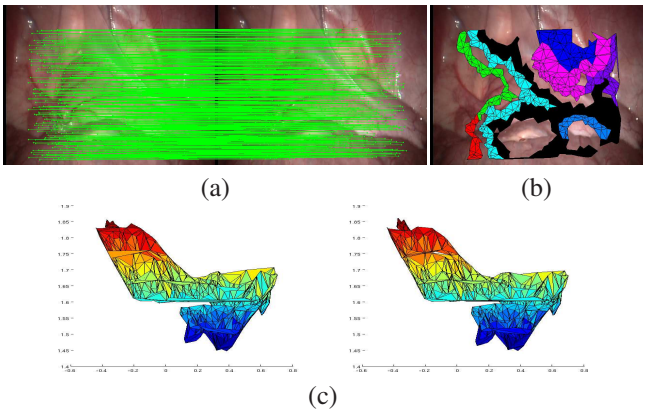


Fig. 9. Surface tracking using the SfS reconstruction. (a) Features matches after inlier selection. (b) Computed clusters. (c) 3D tracked surface.

is performed using the semiglobal block matching (SGBM) algorithm included in OpenCV and provides a less dense reconstruction than the SfS algorithm (see Fig. 1). The baseline of the system constrains the distance to which the scene can be reconstructed using SGBM.

Shape from shading reconstruction. In Fig. 9 we show the reconstruction of a single frame using the SfS reconstruction. Since the reconstruction provided by this approach is up to scale we compute the error relative to the size of the 3D structure. The numerical results of the experiment are presented in the third row of Table I. Notice that the reconstruction error is only 1.5% with a feature match displacement of 0.8 pixels. Similarly to the experiments with the RGB-D images, we observe that the error increases as the distances between matches increases. For instance, a feature match displacement of 2.5 pixels increases the reconstruction error to 7%.

Stereo reconstruction. In Fig. 10 we present the estimation of the deformation of a single frame, using as initial reconstruction the stereo reconstruction given by the SGBM algorithm. We observe that this reconstruction is more realistic that the one provided by SfS, which impacts on the cluster computation (see Fig. 10(b)). The accuracy using this reconstruction is similar to the one using SfS, 1.2% error with the same feature displacement 0.8 pixels. The complete information of this experiment is shown in the last row of Table I. A particular observation of this experiment is related to the reconstruction of the membrane

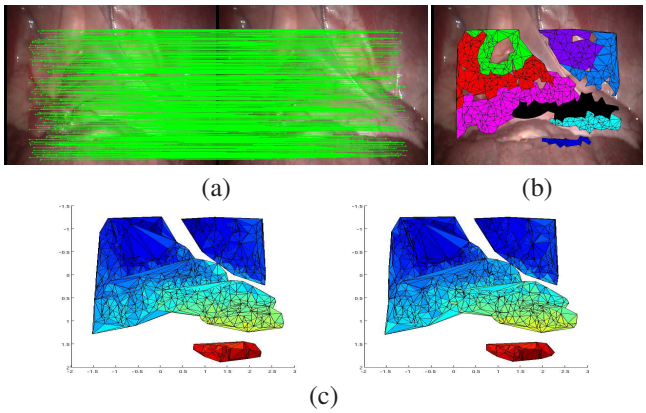


Fig. 10. Surface tracking using the SGBM reconstruction. (a) Features matches after inlier selection. (b) Computed clusters. (c) 3D tracked surface.

(see the middle of the image). Notice that there is no cluster capturing this membrane, even though there are matches and depth information available. This behavior is explained due to that the membrane is almost parallel to the optical axis and its structure cannot be explained with affine transformations.

We have observed that our approach as any NRSfM approach depends on the initial surface reconstruction. Moreover, this surface should be smooth in order to be represented correctly using thin-plate splines with a minimum number of control points. An example of this situation can be observed in the T-shirt sequence, where abrupt changes between adjacent neighbors are present, making difficult to represent the surface patches using thin-plate splines. We have also shown the impact of the feature displacement on the reconstruction accuracy (see Fig. 8). Our approach requires this displacement to be small, since it computes a local optima, where the initial solution is the reconstruction at the previous step.

C. Conclusions and Future Work

In this paper we present a yet simple but efficient scheme to track deformable surfaces. We assume that an initial reconstruction of the object/surface of interest, associated to a reference image frame, is available. Dense structure from motion as well as shape from shading algorithms can be used to generate this initial shape. At first, we extract and match scale invariant features between the reference and current frames. The outlier matches are filtered using an approximation to the maximum clique technique using the distance between matches and the scale information from the detected features as consistency criteria. Secondly, this outlier-free set of matches is subdivided into clusters using local geometric information in the image domain and depth information from the available initial reconstruction. The number of elements of each cluster is further reduced using a thin-plate spline parametrization. Then, the parameters representing the deformed surface, which correspond to the control points of the thin-plate splines, are estimated using a non-linear optimization algorithm that minimizes the reprojection error of all the features matched in the current image. We finally perform experiments in simulated and real data to validate the performance of our approach. We observe that our approach tracks the deformable surface with millimeter accuracy with a considerable small number of control points. Moreover, it is robust to noise in the image coordinates of the matched features. We also observed that the cluster-based formulation allows us to easily discard corrupted patches.

REFERENCES

[1] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 690–696 vol.2.

[2] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito, "Factorization for non-rigid and articulated structure using metric projections," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 2898–2905.

[3] M. Paladini, A. Bartoli, and L. Agapito, "Sequential non-rigid structure-from-motion with the 3d-implicit low-rank shape model," in *European Conference in Computer Vision*, ser. Lecture Notes in Computer Science, vol. 6312. Springer, 2010, pp. 15–28.

[4] A. Agudo, L. Agapito, B. Calvo, and J. M. M. Montiel, "Good vibrations: A modal analysis approach for sequential non-rigid structure from motion," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2014.

[5] M. Salzmann, R. Urtasun, and P. Fua, "Local deformation models for monocular 3D shape recovery," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[6] A. Varol, M. Salzmann, E. Tola, and P. Fua, "Template-free monocular reconstruction of deformable surfaces," in *12th International Conference on Computer Vision*, Sept 2009, pp. 1811–1818.

[7] J. Östlund, A. Varol, D. Ngo, and P. Fua, "Laplacian meshes for monocular 3d shape recovery," in *European Conference on Computer Vision*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7574, pp. 412–425.

[8] A. Bartoli, M. Perriollat, and S. Chambon, "Generalized thin-plate spline warps," *International Journal of Computer Vision*, vol. 88, no. 1, pp. 85–110, May 2010.

[9] D. Pizarro and A. Bartoli, "Feature-based deformable surface detection with self-occlusion reasoning," *International Journal of Computer Vision*, vol. 97, no. 1, pp. 54–70, 2012.

[10] L. Tao and B. Matuszewski, "Non-rigid structure from motion with diffusion maps prior," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1530–1537.

[11] S. Vicente and L. Agapito, "Soft inextensibility constraints for template-free non-rigid reconstruction," in *Computer Vision ECCV 2012*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7574, pp. 426–440.

[12] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[13] M. Visentini-Scarzanella, D. Stoyanov, and G.-Z. Yang, "Metric depth recovery from monocular images using shape-from-shading and specularities," in *IEEE International Conference on Image Processing (ICIP) 2012*, 2012.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal on Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[15] K. Chen, Y. Zhou, Q. Zheng, X. Yang, and L. Song, "Mcm: An efficient geometric constraint method for robust local feature matching," in *MVA'11*, 2011, pp. 190–193.

[16] G. Puerto-Souza and G. Mariottini, "Hierarchical multi-affine (hma) algorithm for fast and accurate feature matching in minimally-invasive surgical images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 2007–2012.

[17] F. L. Bookstein, "Principal warps: thin-plate splines and the decomposition of deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, Jun 1989.

[18] A. Varol, M. Salzmann, P. Fua, and R. Urtasun, "A constrained latent variable model," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 2248–2255.

[19] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.