Simultaneous Point Matching and 3D Deformable Surface Reconstruction *

Appu Shaji[1] Aydin Varol[1] Lorenzo Torresani[2] Pascal Fua[1]
[1] Computer Vision Laboratory, EPFL, Lausanne, Switzerland
[2] Dartmouth College, Hanover, NH, U.S.A.
appu.shaji@epfl.ch , aydin.varol@epfl.ch , lorenzo@cs.dartmouth.edu , pascal.fua@epfl.ch

Abstract

It has been shown that the 3D shape of a deformable surface in an image can be recovered by establishing correspondences between that image and a reference one in which the shape is known. These matches can then be used to set-up a convex optimization problem in terms of the shape parameters, which is easily solved. However, in many cases, the correspondences are hard to establish reliably.

In this paper, we show that we can solve simultaneously for both 3D shape and correspondences, thereby using 3D shape constraints to guide the image matching and increasing robustness, for example when the textures are repetitive.

This involves solving a mixed integer quadratic problem. While optimizing this problem is NP-hard in general, we show that its solution can nevertheless be approximated effectively by a branch-and-bound algorithm.

1. Introduction

A number of techniques have recently been proposed to recover the shape of a deformable 3D surface from a single image when point correspondences can be established with a reference image in which the shape is known [18, 30, 21]. Although these algorithms tolerate some mismatches, they will fail if there are too many of them, as happens in the presence of repetitive patterns or when the texture quality is too poor to guarantee reliable correspondences.

In the case of rigid objects, such difficulties can be overcome by taking into account the constraints imposed by the epipolar geometry. However, for deformable surfaces, the constraints are much weaker and most existing algorithms establish correspondences based solely on local appearance without considering the spatial layout of features and the constraints it imposes. In other words, available information is not fully exploited when computing the correspondences and it is left to robust estimators to deal with mistakes when attempting to recover deforming 3D shapes.

*This work was supported in part by the Swiss National Science Foundation.

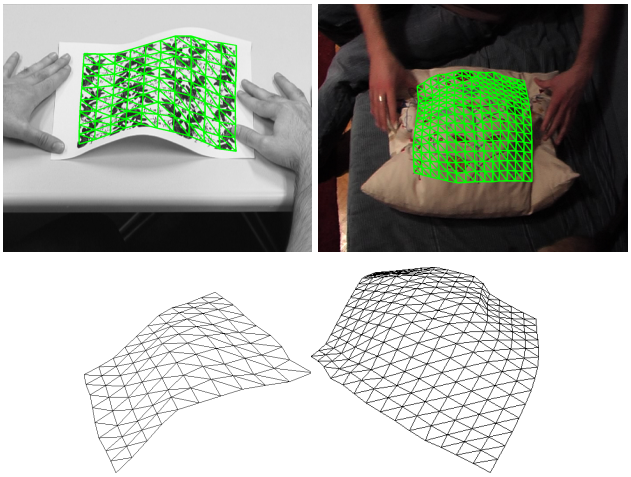


Figure 1. 3D reconstruction of textured deformable surfaces from single view.

Recent work [2, 13, 28, 11] has addressed the problem of non-rigid feature matching by seeking correspondences such that the 2D arrangements of the matching features in the two images are consistent. These 2D constraints, however, do not take into account 3D perspective distortion effects and may therefore not accurately reflect 3D geometry. By contrast, in this paper, we show that we can simultaneously establish correspondences and impose projective constraints, which yields much higher quality matches in difficult situations that are plagued by ambiguities.

Our algorithm solves simultaneously for 2D correspondences and 3D shape by optimizing a single objective function over both the set of all possible correspondences and the set of all possible shapes. This amounts to solving a mixed integer quadratic problem, which is NP-hard in general. We propose a branch-and-bound solver for this problem and show that, in practice, this method yields excellent approximate solutions. In challenging situations such as the one depicted in Fig. 1, our approach outperforms reconstruction methods relying on correspondences pre-computed by matching appearance descriptors.

Our main contribution therefore is a novel branch-and-

bound formulation of the 3D reconstruction problem. By simultaneously establishing 3D correspondences and recovering 3D shape, we obtain good results in challenging situations, such as when repetitive patterns are present, where traditional methods such as [18, 30, 20] requiring correspondence as input are destined to fail. To the best of our knowledge, we are the first to show that this can be done without a prior model and using only a single input image.

2. Related Work

In this section we briefly review the existing literature on both monocular reconstruction of deformable surfaces and feature point matching between two views.

Reconstruction of Deformable 3D Surfaces: Monocular reconstruction of deformable surfaces is inherently underconstrained as many different 3D shapes can produce the same image projection. To overcome these ambiguities numerous approaches have been proposed. The earliest ones relied on physically inspired models and attempted to capture intrinsic physical properties of the deforming objects [24]. Modal analysis [17] has been often used in conjunction with such models to reduce the degrees of freedom in the problem.

To overcome the limitations of the physically inspired models, statistical learning methods have been applied to build linear and non-linear surface deformation models from 3D training data [9, 5]. These models can be both accurate and easy to learn but can only represent surfaces behaving in the same way as those of the training examples. Furthermore, in practice availability of 3D training data is scarce and expensive to acquire.

A number of methods have been proposed for model-free 3D reconstruction from 2D tracking data. Non-rigid structure from motion algorithms [7, 6, 27] constrain the reconstruction by assuming that the 3D shapes lie in or near a linear subspace, which is estimated from the given 2D motion. Other approaches [20, 30] have employed temporal consistency instead of shape smoothness to recover 3D surfaces from tracking data.

Recently, several authors [20, 18] have proposed algorithms that can estimate a 3D surface from individual images by exploiting inextensibility constraints. Although these methods remove the need of tracking over whole sequences, they still require correspondences between two views as input. For template-based methods, such as [20], these correspondences are established between a reference view in which the 3D shape is known and a query frame where the surface deformation is to be recovered. Similarly, template-free methods [29] require correspondences between individual frames in a video sequence.

Matching: Feature-based deformable surface reconstruction methods require establishing wide-baseline correspondences between two views of the deforming surface. To this end, most techniques [20, 21] use SIFT [15] correspondences. In some of the earlier studies fast matching methods such as randomized trees [14] are also used for detecting 2D deformations in real-time [19]. In addition, for 2D surface tracking applications, cross-correlation based matching methods are employed to track 2D features throughout the whole sequence [20]. Common to all these methods, the outlier matches are explicitly detected in the reconstruction phase using robust estimation methods. Thus, their performance highly depends on the quality of the inlier matches that are provided by the matching algorithm.

Most prior methods for non-rigid feature correspondence have proposed matching appearance descriptors under a smooth, or piece-wise smooth, parametric transformation describing the 2D geometric mapping relating the feature sets in the two views [25, 1]. However, these simple parametric 2D transformations are not suitable for highly deformable objects, such as clothing.

Recently, several authors [2, 13, 28, 11] have proposed to cast non-rigid feature correspondence as a graph matching problem. The feature sets extracted from the two images are viewed as two separate graphs, where the nodes represent features and the edges encode the spatial arrangement between pairs of features. The matching process then aims at establishing correspondences that match points having similar appearance, while preserving as much as possible the spatial relationships between the features. However, in these models, edge similarity is defined in terms of simple 2D geometric consistency measures, which cannot accurately model 3D effects such as perspective projection or deformations orthogonal to the camera plane.

By contrast in our work, we propose a graph matching objective directly expressed in terms of non-rigid 3D shape parameters and similarity between feature descriptors. The optimization of this objective leads to simultaneous outlier rejection, point matching, and 3D surface reconstruction. This joint scheme is well studied for the camera pose estimation using rigid objects [8, 23] but it has not been studied for the deformable surface reconstruction problem. Our approach is similar in spirit to methods for direct non-rigid 3D modeling from video [6, 26] which directly optimize shape over raw image data, without pre-computed correspondences. Our methods extends these approaches to address the problem of surface reconstruction from individual image pairs.

3. Formulation

In [21] it is shown that, given enough 2D point correspondences between an *input image* and a *reference image* in which the shape of a deformable surface is represented by a known 3–D mesh, the shape in the input image can be

recovered[1]. Shape recovery is accomplished by minimizing a convex objective function, provided that the correspondences are acquired beforehand and not too many of them are erroneous.

In this section, we first briefly summarize the formulation of [21] and then recast the problem as one of simultaneously solving for shape and correspondence. We introduce the corresponding optimization procedure in the next section and show in the results section that it allows us to overcome the limitations of the earlier approach.

3.1. Shape Recovery as Convex Optimization

We represent a surface as a triangulated mesh made of N_v vertices $\mathbf{v}_i = [x_i, y_i, z_i]^T$, $1 \leq i \leq N_v$ connected by a set E_{mesh} of N_e edges. We stack the vertices into a vector $\mathtt{V} = [\mathbf{v}_1^T, \cdots, \mathbf{v}_{N_v}^T]^T \in \mathbb{R}^{3N_v}$. Without loss of generality, we describe the vertices in the camera reference frame. We assume that the camera is calibrated, with known intrinsic and extrinsic parameters.

Let $\mathtt{P}' = \{p'_1, \ldots, p'_{n'}\}$ and $\mathtt{P} = \{p_1, \ldots, p_n\}$ be the two feature sets extracted from the reference image and the target image, respectively. Let us start by assuming that we are given the set of correspondences $C \subseteq \{1, \ldots, n'\} \times \{1, \ldots, n\}$ between these two sets: $(r,t) \in C$ indicates that feature $p'_r \in \mathtt{P}'$ in the reference image matches point $p_t \in \mathtt{P}$ in the target image. Since the 3D surface for the reference image is known, for each feature point $p'_r \in \mathtt{P}'$ we can compute its mesh point $\mathbf{p}'_r \in \mathbb{R}^3$. Each 3D point $\mathbf{p}'_r$ can be expressed as a weighted sum of the vertices of the mesh facet it belongs to. The weights are the barycentric coordinates of $\mathbf{p}'_r$ and do not change as the surface deforms. Thus, for each $(r,t) \in C$, we can compactly write that $\mathbf{p}'_r$ must project to feature point p_t in the target image as:

$$\mathtt{M}_{(r,t)}\mathtt{V} = 0 \ , \tag{1}$$

where $\mathtt{M}_{(r,t)} \in \mathbb{R}^{2\times 3N_v}$ is the projection matrix, which can be computed in terms of the image coordinates p_t and the barycentric coordinates of point $\mathbf{p}'_r$ [20]. Stacking these matrices for all $|C|$ correspondences yields a $2|C| \times 3N_v$ matrix $\mathtt{M}$ and jointly minimizing the reprojection error for all correspondences amounts to minimizing

$$\mathrm{E}^{\text{reproj}}(\mathtt{V}) = \|\mathtt{M}\mathtt{V}\|_2^2 = \sum_{(r,t)\in C} \|\mathtt{M}_{(r,t)}\mathtt{V}\|_2^2 \ . \tag{2}$$

The matrix $\mathtt{M}$, however, is very poorly conditioned. As a result, since the correspondences are always slightly noisy, simply solving the system in the least-squares sense does not return satisfactory solutions. Instead, it was found necessary to add several constraints [22].

[1]Note that this problem differs from traditional stereo since the shapes in the input and reference image may be different.

Since most materials do not perceptibly shrink or extend while deforming, the deformations must be such that the distances between vertices are preserved, which means,

$$||\mathbf{v}_k - \mathbf{v}_j||_2 \leq l_{j,k}, \ \forall (j,k) \in E_{\text{mesh}} \ ,$$

where $l_{j,k}$ represents the geodesic distance between vertices j and k. Note that the constraint is formulated as an inequality because we use a discrete representation of a continuous surface. As a result, when the surface folds, the *Euclidean* distance between two vertices can decrease without any change in the geodesic one.

While the inequalities introduced above prevent the mesh from expanding, they still allow it to shrink to a single point. This could be remedied by maximizing the mesh area under our constraints. However, this would yield a non-convex problem. Instead, the approach in [21] exploits the fact that, in the perspective camera model, the lines-of-sight are not parallel. Thus the largest distance between two points is reached when the surface is furthest away from the camera. Therefore, for each correspondence $(r,t) \in C$ a reconstruction for $\mathbf{p}_t$ can be obtained by maximizing the depth d_t along its line-of-sight $\mathbf{s}_t$. This term can be computed as

$$d_t = \mathbf{p}_t^T \mathbf{s}_t = \mathtt{V}^T \mathtt{B}_r^T \mathbf{s}_t \ , \tag{3}$$

where $\mathtt{B}_r$ is the $3 \times 3N_v$ matrix containing the barycentric coordinates of point $\mathbf{p}'_r$ placed to correctly match the vertices of the facet to which the point belongs. In order to enforce this for all vertices, the following linear term was added to the one of Eq. (2)

$$\mathrm{E}^{\text{depth}}(\mathtt{V}) = -\sum_{(r,t)\in C} \mathbf{s}_t^T \mathtt{B}_r \mathtt{V} \ , \tag{4}$$

To further stabilize the system, the approach of [21] included a quadratic regularization term $\mathrm{E}^{\text{deform}}(\mathtt{V})$ that penalizes any local deformation that deviates from a deformation model trained on inextensible meshes.

Bringing all these terms together, the 3D shape is recovered by solving the convex optimization problem

$$\begin{aligned} &\min_{\mathtt{V}} \ \mathrm{E}^{\text{reproj}}(\mathtt{V}) + w_1 \mathrm{E}^{\text{depth}}(\mathtt{V}) + w_2 \mathrm{E}^{\text{deform}}(\mathtt{V}) \\ &\text{subject to } ||\mathbf{v}_k - \mathbf{v}_j||_2 \leq l_{j,k}, \ \forall (j,k) \in E_{\text{mesh}} \ , \end{aligned} \tag{5}$$

where w_1 and w_2 are weights that control the relative importance of the distance and smoothness terms with respect to the reprojection error one.

3.2. Solving for both Correspondences and Shape

To remove the requirement that the correspondences be known *a priori*, let us again consider the feature-point sets $\mathtt{P}'$ in the reference image and $\mathtt{P}$ in the input image, which were introduced at the beginning of Section 3.1. Let $A =$

$\{1,\ldots,n'\}\times\{1,\ldots,n\}$ be the set of all possible correspondences between these two sets. Let us use a binary variable $x_{(r,t)} \in \{0,1\}$ to indicate whether $\mathbf{p}'_r$ matches $\mathbf{p}_t$, with value 1 indicating an active correspondence. We collect all these binary variables into a vector $\mathbf{x} \in \{0,1\}^A$ describing the set of correspondences for a specific matching configuration. Since any point can match at most one point in the other image, the set of feasible correspondences reduces to

$$\mathcal{M} = \{\mathbf{x} \in \{0,1\}^A | \quad \sum_{t=1}^{n} x_{(r,t)} \leq 1 \; \forall r \in \{1,...,n'\}, \; \sum_{r=1}^{n'} x_{(r,t)} \leq 1 \; \forall t \in \{1,...,n\}\} \quad (6)$$

We now rewrite the objective in Eq. (5) as a function of the active correspondences specified by $\mathbf{x} \in \mathcal{M}$:

$$E(\mathbf{x},\mathtt{V}) = \mathrm{E}^{\mathrm{reproj}}(\mathbf{x},\mathtt{V}) + w_1\mathrm{E}^{\mathrm{depth}}(\mathbf{x},\mathtt{V}) + w_2\mathrm{E}^{\mathrm{deform}}(\mathtt{V}) \quad (7)$$

$$\mathrm{E}^{\mathrm{reproj}}(\mathbf{x},\mathtt{V}) = \sum_{(r,t)\in A} x_{(r,t)} \|\mathtt{M}_{(r,t)}\mathtt{V}\|_1 \quad (8)$$

$$\mathrm{E}^{\mathrm{depth}}(\mathbf{x},\mathtt{V}) = \sum_{(r,t)\in A} x_{(r,t)} \mathbf{s}_t^T \mathtt{B}_r \mathtt{V}. \quad (9)$$

Since we observed that the scheme described by Eq. (5) fails when many outliers are present, we replaced the L2 norm in Eq. (2) by the L1 norm, which is known to be more robust [12]. An added benefit is to reduce the number of quadratic terms in the objective, thus making optimization simpler.

However, minimizing this objective would not yield the desired answer since it would result in the trivial solution in which no correspondence is activated. We therefore introduce two additional terms:

- We penalize unmatched features by means of an occlusion term which decreases as more points are matched:

$$\mathrm{E}^{\mathrm{occ}}(\mathbf{x}) = 1 - \frac{1}{\min(n',n)} \sum_{(r,t)\in A} x_{(r,t)} \quad . \quad (10)$$

- We encourage correspondences between similar feature points by defining

$$\mathrm{E}^{\mathrm{match}}(\mathbf{x}) = \sum_{(r,t)\in A} x_{(r,t)} c^{\mathrm{match}}_{(r,t)} \quad , \quad (11)$$

where $c^{\mathrm{match}}_{(r,t)}$ is a measure of appearance difference between the two features. In practice, we set this measure proportional to the inverse of the dot product between SIFT descriptors computed at the feature points.

Adding all terms together yields the optimization problem of mixed integer quadratic form

$$\begin{aligned} \min_{\mathbf{x},\mathtt{V}} \; & \underbrace{\mathrm{E}^{\mathrm{reproj}}(\mathbf{x},\mathtt{V}) + w_1\mathrm{E}_{\mathrm{depth}}(\mathbf{x},\mathtt{V})}_{\text{mixed integer term}} + \\ & w_2(\underbrace{\mathrm{E}^{\mathrm{match}}(\mathbf{x}) + \mathrm{E}^{\mathrm{occ}}(\mathbf{x})}_{\text{binary terms}}) + \underbrace{w_3\mathrm{E}^{\mathrm{deform}}(\mathtt{V})}_{\text{linear term}} \\ \text{subject to } & \|\mathbf{v}_k - \mathbf{v}_j\|_2 \leq l_{j,k}, \; \forall (j,k) \in E_{\mathrm{mesh}} \\ & \mathbf{x} \in \mathcal{M} \end{aligned} \quad (12)$$

Unfortunately, this formulation, unlike that of Eq. (5), is not convex due to the integer constraints. In addition, mixed integer programs are NP-hard in general [4]. However, we show in the following section that this particular problem has a specific structure, which can be exploited by a branch-and-bound algorithm to effectively approximate the solution.

4. Method

To compute an approximate solution of Eq. (12), we pursue a branch-and-bound strategy. It involves iteratively partitioning the solution space into mutually exclusive subdomains. At each iteration, we reduce the original problem into a set of 'easier' or *relaxed* problems that are approximate versions of the original one. As we will see, when the problem is as well structured as ours, it is possible to prune away sub-domains from the search tree effectively, thereby avoiding complete enumeration of an exponentially large solution space. Fig. 2 shows an example of our estimated reconstruction, which improves as the optimization progresses.

In our case, the relaxed problems are easy-to-solve linear programs that are tight enough to approximate the original problem well and therefore to give good bounds on the potential solutions. In the remainder of this section, we first show how we reduce our quadratic terms into linear ones by introducing auxiliary variables, and then introduce our branch-and-bound scheme.

Relaxations: To obtain a linear objective, we move the quadratic terms to the constraint set by bounding them from above with auxiliary variables $\alpha_{(r,t)}, \beta_{(r,t)}, \gamma_{(r,t)}$ [4]. Let $\mathtt{M}^{(1)}_{(r,t)}$ and $\mathtt{M}^{(2)}_{(r,t)}$ be the first and the second row of $\mathtt{M}_{(r,t)}$, respectively. Then, each reprojection error term $x_{(r,t)}\|\mathtt{M}_{(r,t)}\mathtt{V}\|_1$ can be bounded by rewriting it as $\left(x_{(r,t)}|\mathtt{M}^{(1)}_{(r,t)}\mathtt{V}| + x_{(r,t)}|\mathtt{M}^{(2)}_{(r,t)}\mathtt{V}|\right)$ and by constraining each of these two terms independently[2]. This gives rise to the

[2] In order to obtain linear constraints we rewrite any bound of the form $|a| \leq b$ as the pair of linear constrains $a \leq b$, and $-a \leq b$.

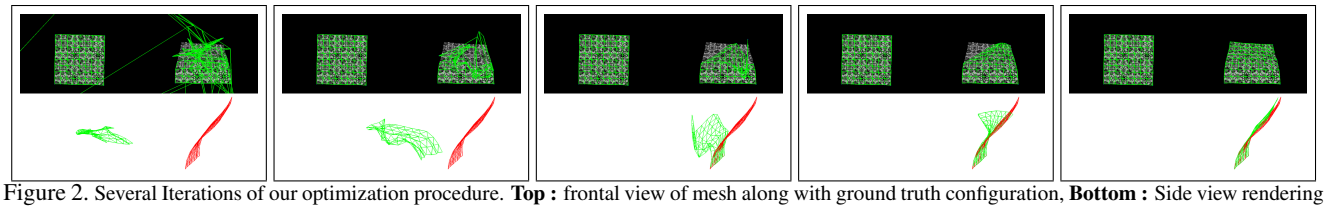


Figure 2. Several Iterations of our optimization procedure. **Top :** frontal view of mesh along with ground truth configuration, **Bottom :** Side view rendering of 3D mesh estimate (green) along with ground truth configuration (red).The leftmost images depict the surface estimate after the first iteration, which was obtained by solving the relaxed problem for a large domain. As we reduce the domain, the quality of the estimates increases. The final one is depicted by the rightmost images

following equivalent reformulation of Eq. (12):

$$\min_{\mathbf{x},\mathtt{V},\alpha,\beta,\gamma} \sum_{(r,t)\in A} \big(\alpha_{(r,t)} + \beta_{(r,t)} + w_1\gamma_{(r,t)}\big) + w_2(\mathrm{E}^{\mathrm{match}}(\mathbf{x}) + \mathrm{E}^{\mathrm{occ}}(\mathbf{x})) + w_3\mathrm{E}^{\mathrm{deform}}(\mathtt{V})$$

$$\text{subject to } \begin{cases} \alpha_{(r,t)} \geq x_{(r,t)}\mathtt{M}^{(1)}_{(r,t)}\mathtt{V}, \\ \alpha_{(r,t)} \geq -x_{(r,t)}\mathtt{M}^{(1)}_{(r,t)}\mathtt{V} \\ \beta_{(r,t)} \geq x_{(r,t)}\mathtt{M}^{(2)}_{(r,t)}\mathtt{V}, \\ \beta_{(r,t)} \geq -x_{(r,t)}\mathtt{M}^{(2)}_{(r,t)}\mathtt{V} \\ \gamma_{(r,t)} \geq x_{(r,t)}(\mathtt{s}_t^T\mathtt{B}_r)\mathtt{V} \\ \|\mathbf{v}_k - \mathbf{v}_j\|_2 \leq l_{j,k},\ \forall (j,k) \in E_{\mathrm{mesh}} \\ \mathbf{x} \in \mathcal{M} \end{cases} \tag{13}$$

In our sub-problems, we relax the integrability requirement of the assignment variable $x_{(r,t)}$ by its continuous counterpart $\tilde{x}_{(r,t)} \in [0,1]$. Finally, we define bounds on each variable:

$$\mathtt{V}_i^L \leq \mathtt{V}_i \leq \mathtt{V}_i^U, \qquad \forall i = 1,\ldots,3N_v \tag{14}$$

$$\tilde{x}^L_{(r,t)} \leq \tilde{x}_{(r,t)} \leq \tilde{x}^U_{(r,t)}, \qquad \forall (r,t) \in A \tag{15}$$

These bounds define rectangular domains which will be updated by the branch and bound procedure.

Note that each quadratic term bounded by $\alpha_{(r,t)}$, $\beta_{(r,t)}$ or $\gamma_{(r,t)}$ can be written in the general form $x_{(r,t)}\sum_{i=1}^{3N_v} a_i\mathtt{V}_i$ for $a_i \in \mathbb{R}$ appropriately defined in terms of the entries of $\mathtt{M}_{(r,t)}, \mathbf{s}_t$ and $\mathtt{B}_r$. Thus, we express each of these quadratic terms in linear form by replacing it with a new variable y subject to the following constraints

$$\begin{aligned} v^L x_{(r,t)} \quad &\leq y \leq \quad v^U x_{(r,t)} \\ \sum_{i=1}^{3N_v} \mathtt{V}_i - v^L(1-x_{(r,t)}) \quad &\leq y \leq \quad \sum_{i=1}^{3N_v} \mathtt{V}_i - v^U(1-x_{(r,t)}), \end{aligned}$$

where

$$v^L = \sum_{i=1}^{3N_v} a_i\mathtt{V}_i^L, \qquad v^U = \sum_{i=1}^{3N_v} a_i\mathtt{V}_i^U.$$

Inextensibility constraints given by $\|\mathbf{v}_k - \mathbf{v}_j\|_2 \leq l_{j,k}$ are instances of second order cone constraints. It has been shown in [3] that SOCP can be linearized by outer-approximating it by a set of $N \geq 0$ linear constraints. The outer-approximation gap shrinks as we increase the number of linear constraints. For our problem, we observe that $N = 2$ suffices to achieve good convergence. The relaxed set of constraints can thus be represented as:

$$\begin{aligned} |\mathbf{v}_k| \leq \xi^0, \qquad & |\mathbf{v}_j| \leq \eta^0, \\ \xi^1 \quad = \quad & \cos(\tfrac{\pi}{4})\xi^0 + \sin(\tfrac{\pi}{4})\eta^0, \\ \eta^1 \quad \geq \quad & |-\sin(\tfrac{\pi}{4})\xi^0 + \cos(\tfrac{\pi}{4})\eta^0|, \\ \xi_2 \leq l_{j,k}, \qquad & \eta_2 = \tan(\tfrac{\pi}{2^{N+1}})\xi_2. \end{aligned}$$

During an iteration, the relaxed linear program needs not necessarily satisfy all the constraints for the original problem. However, we can use this information to strengthen the formulation by adding additional constraint (known as cutting planes) to the LP formulation. These constraints are easy to generate and constitute simple linear inequalities, which help in restricting the search space for future iterations. Specifically, for the mixed terms which appear in constraints of Eq. (13) cutting plane constraints are easily given by McCormick under-estimators [16]:

for $\lambda > 0$

$$\begin{aligned} \lambda\tilde{x}_{(r,t)}\mathtt{V}_j &\leq \lambda\tilde{x}^L_{(r,t)}\mathtt{V}_j + \lambda\mathtt{V}_j^L\tilde{x}_{(r,t)} - \lambda\tilde{x}^L_{(r,t)}\mathtt{V}_j^L, \\ \lambda\tilde{x}_{(r,t)}\mathtt{V}_j &\leq \lambda\tilde{x}^U_{(r,t)}\mathtt{V}_j + \lambda\mathtt{V}_j^U\tilde{x}_{(r,t)} - \lambda\tilde{x}^U_{(r,t)}\mathtt{V}_j^U. \end{aligned}$$

for $\lambda < 0$

$$\begin{aligned} \lambda\tilde{x}_{(r,t)}\mathtt{V}_j &\leq \lambda\tilde{x}^U_{(r,t)}\mathtt{V}_j + \lambda\mathtt{V}_j^L\tilde{x}_{(r,t)} - \lambda\tilde{x}^U_{(r,t)}\mathtt{V}_j^L, \\ \lambda\tilde{x}_{(r,t)}\mathtt{V}_j &\leq \lambda\tilde{x}^L_{(r,t)}\mathtt{V}_j + \lambda\mathtt{V}_j^U\tilde{x}_{(r,t)} - \lambda\tilde{x}^L_{(r,t)}\mathtt{V}_j^U. \end{aligned}$$

Furthermore, it is often possible to check if there is no feasible solution for a given domain by considering only the values of the constraint functions at the extrema of the given domain. For quadratic constraint functions these feasibility checks can be performed efficiently using interval-arithmetic based methods similar to [10].

Branching Procedure: After every iteration, we round the binary terms to the nearest 0-1 values and check if it is possible to update the global lower and upper bounds of the energy function [4]. If there are violated constraints at the end of the above mentioned procedure, we branch on a variable that participates in one or more of violated constraints,

with preference for variables participating in non-convex constraints. When there are multiple candidate variables for the branching operation, we pick the variable with the highest pseudo-cost, an approximate measure of the objective function gain obtained per unit change when this variable is chosen for branching [3].

5. Results

We now present results obtained on both synthetic and real images and then compare against the method of [21]. Our implementation relies on the SCIP mixed integer programming solver [3] whose plug-in nature makes it well-adapted for our purposes.

In all our experiments we use the following optimization weights: the mode weights are set to be 100, the appearance weights to be 0.6, and the depth term to $2/3$.

Synthetic Data: To produce the synthetic images we used a Vicon^{tm} optical motion capture system to represent a real deforming piece of paper as a set of 3D meshes, which we then texture-mapped using the repetitive pattern of Fig. 3 **(a)**. We use an image of a flat version of the mesh as our reference and all others as input meshes in turn. Note that we perform the computations for each frame independently and never exploit temporal consistency.

Our results are summarized by Figs. 3 and 4. We use the points detected by SIFT [15] as feature points, and their corresponding descriptors to measure appearance similarity between points. Although our approach does not need them, we compute correspondences using the SIFT criterion so that we can also run the method of [21], which requires precomputed correspondences as input. In Fig. 3 we plot, for each frame and for each method, the final 3D reconstruction error, the percentage of erroneous correspondences retained at the end of the optimization, and the percentage of facets not covered.

As the texture is highly repetitive, SIFT matching often returns bad correspondences, which cause the method of [21] to fail when there are more than about 30% of them. By contrast, our results remain consistently good throughout the sequence. As shown in the third row of Fig. 4 the approach of [28] yields better correspondences than [15] but still not as good as ours.

In this dataset,the average reconstruction time of each frame is 15 min 32 sec on a 2.00 GHz Core(2) Duo Pentium machine with 2GB of memory, and maximum and minimum reconstruction time for a frame being 34 min 29 sec and 9 min 11 sec respectively. The average reconstruction time for Salzmann et.al [21] is 1 min 10 sec per frame. Please note that, even though our method is computationally more expensive, this expense is warranted by the fact that our results are of much higher quality.

Real Data: As shown in Fig. 6 and Fig. 7, we also applied our approach to the real deforming piece of paper and a deforming cushion. Due to the repeating nature of the texture of the paper, the reconstruction method which relies on pre-computed SIFT correspondences [21] does not perform as well as the proposed method. The resultant inlier correspondences and the 3D reconstructions are depicted in Fig. 5.

6. Conclusion

We have shown that the shape of a deformable 3D surface can be effectively recovered from one single image given that it is known in another, even when correspondences between the two images cannot be easily established *a priori*. This is accomplished by solving simultaneously for shape and for correspondences. We formulate it as a mixed integer quadratic problem, which we solve using a branch-and-bound approach. We demonstrated performance superior to that of state-of-the-art techniques on repetitive textures, which make point matching difficult and unreliable.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *PAMI*, 24(24):509–522, April 2002.

[2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. volume 1, pages 26–33, 2005.

[3] T. Berthold, S. Heinz, and S. Vigerske. Extending a CIP framework to solve MIQCPs. ZIB-Report 09-23, ZIB, 2008.

[4] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2 edition, 1999.

[5] V. Blanz and T. Vetter. A Morphable Model for The Synthesis of 3–D Faces. In *SIGGRAPH*, pages 187–194, Los Angeles, CA, August 1999.

[6] M. Brand. A direct method of 3D factorization of nonrigid motion observed in 2D. In *CVPR*, pages 122–128, 2005.

[7] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, 2000.

[8] M. Chli and A. J. Davison. Active Matching. In *ECCV*.

[9] T. Cootes, G. Edwards, and C. Taylor. Active Appearance Models. In *ECCV*, pages 484–498, Freiburg, Germany, June 1998.

[10] F. Domes and A. Neumaier. Constraint propagation on quadratic constraints. *Constraints*, August 2009.

[11] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. In *CVPR*, pages 1980–1987, 2009.

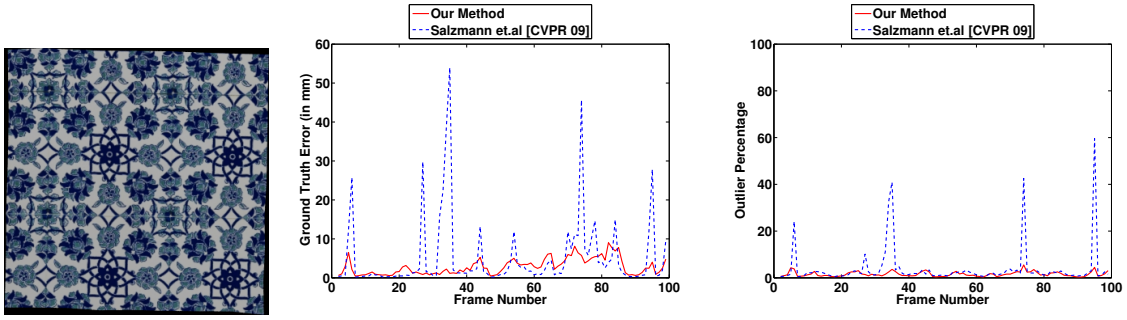
[12] Q. Ke and T. Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR*, June 2005.

[13] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005.

[14] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *PAMI*, 28(9):1465–1479, Sept. 2006.

[15] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 20(2):91–110, 2004.

[16] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I Convex underestimating problems. *Mathematical Programming*, 10:147–175, December 1976.



(a) Texture-map used to synthesize the images

(b) Average 3D distance to ground-truth

(c) Percentage of outlierscp remaining

Figure 3. **Repetitive texture map and performance evaluation**

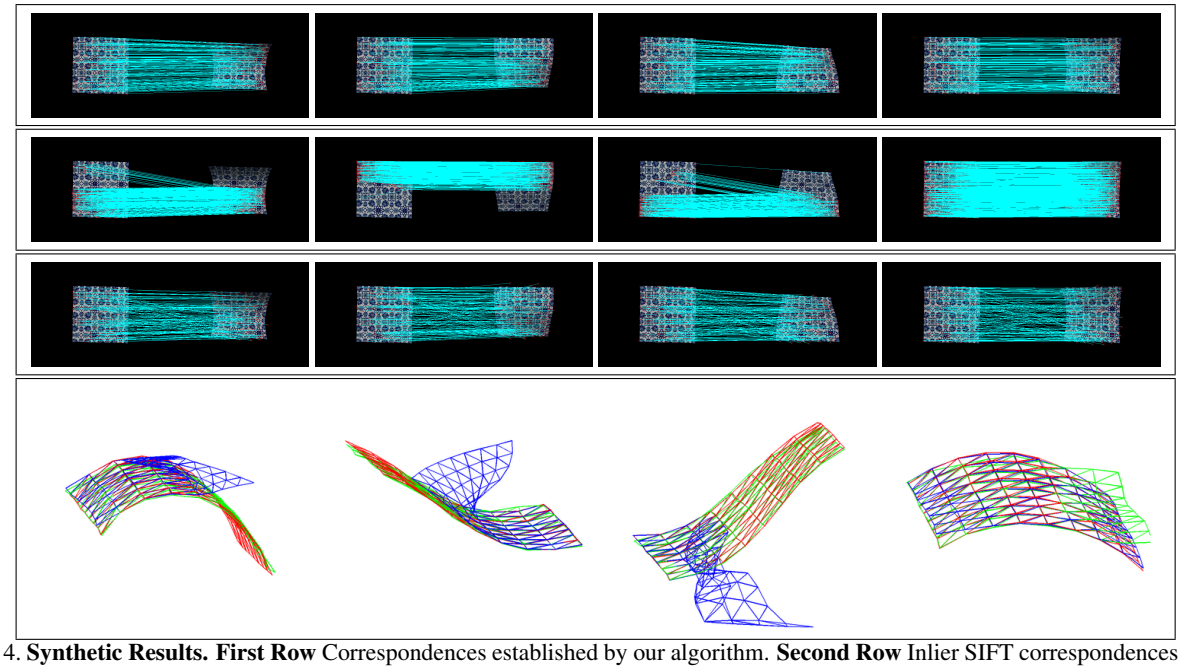


Figure 4. **Synthetic Results. First Row** Correspondences established by our algorithm. **Second Row** Inlier SIFT correspondences according to [21]. **Third Row** Inlier correspondences returned by the method of [28]. **Fourth Row** 3D surface reconstructions. The green mesh is the output of our algorithm, the blue one the output of [21] using SIFT correspondences and the red one the ground-truth mesh.

[17] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *PAMI*, 13:715–729, 1991.

[18] M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. In *BMVC*, 2008.

[19] J. Pilet, V. Lepetit, and P. Fua. Fast Non-Rigid Surface Detection, Registration and Realistic Augmentation. *IJCV*, 76(2), February 2008.

[20] M. Salzmann. *Learning and Recovering 3D Surface Deformations*. PhD thesis, EPFL, Jan 2009.

[21] M. Salzmann and P. Fua. Reconstructing Sharply Folding Surfaces: A Convex Formulation. In *CVPR*, Miami, FL, June 2009.

[22] M. Salzmann, V. Lepetit, and P. Fua. Deformable Surface Tracking Ambiguities. In *CVPR*, Minneapolis, MI, June 2007.

[23] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Pose priors for simultaneously solving alignment and correspondence. In *ECCV*, Marseille, France, October 2008.

[24] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking Models and 3D Object Reconstruction. *IJCV*, 1:211–221, 1987.

[25] P. Torr, A. W. Fitzgibbon, and A. Zisserman. Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *ICCV*, 1998.

[26] L. Torresani and A. Hertzmann. Automatic non-rigid 3d modeling from video. In *ECCV*, pages 299–312, 2004.

[27] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 30(5):878–892, 2008.

[28] L. Torresani, V. Kolmogorov, and C. Rother. Feature Correspondence via Graph Matching: Models and Global Optimization. In *ECCV*, pages II: 596–609, 2008.

[29] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *ICCV*, Kyoto, Japan, September 2009.

[30] J. Zhu, S. Hoi, C. Steven, Z. Xu, and M. Lyu. An effective approach to 3d deformable surface tracking. In *ECCV*, pages 766–779, Marseille, France, 2008.

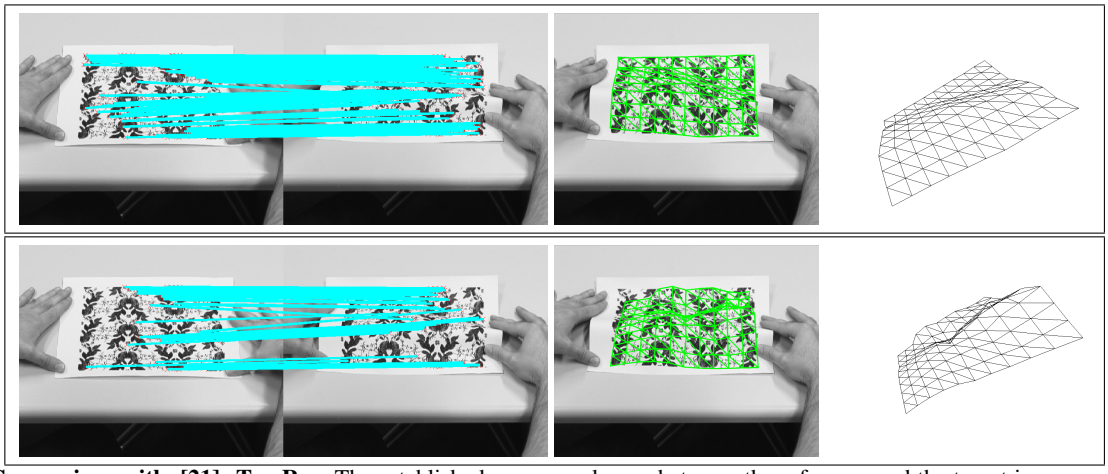


Figure 5. **Comparison with [21]. Top Row** The established correspondences between the reference and the target image, reconstructed 3D mesh reprojected into the target image, and the same mesh seen from a different viewpoint, respectively. **Bottom Row** Similar outputs for the method of [21].

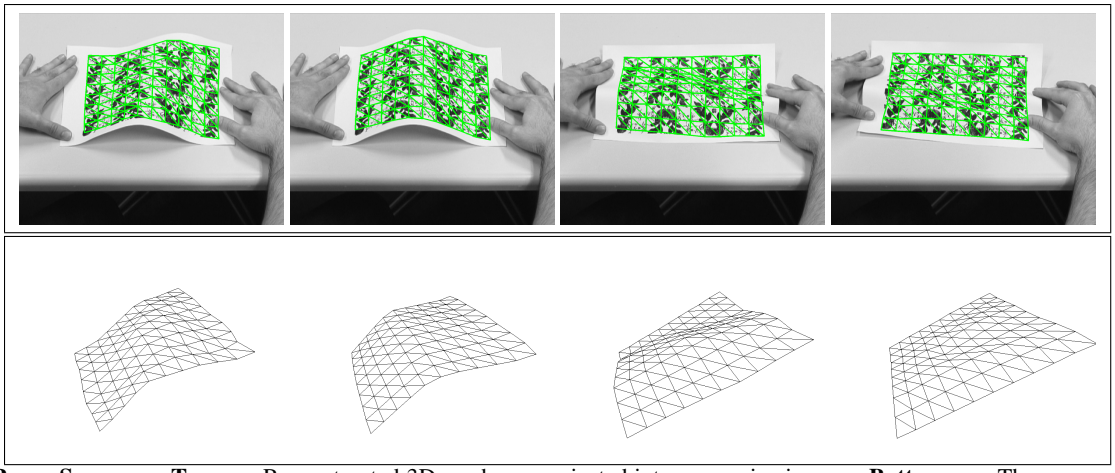


Figure 6. **Paper Sequence. Top row** Reconstructed 3D meshes reprojected into successive images. **Bottom row** The same meshes seen from a different viewpoint.

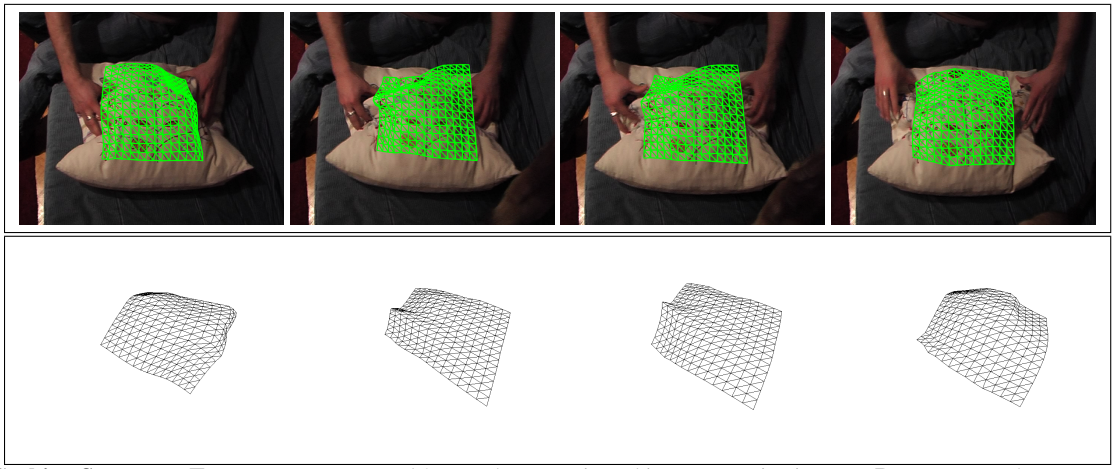


Figure 7. **Cushion Sequence. Top row** Reconstructed 3D meshes reprojected into successive images. **Bottom row** The same meshes seen from a different viewpoint.