Shape From Motion of Nonrigid Objects: The Case of Isometrically Deformable Flat Surfaces

Ricardo Ferreira
ricardo@isr.ist.utl.pt
João Xavier
jxavier@isr.ist.utl.pt
João Costeira
jpc@isr.ist.utl.pt

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Lisbon, Portugal

This work was partially supported by the Fundacao para a Ciencia e Tecnologia (ISR/IST pluriannual funding) through the POSC Program that includes FEDER funds and J. Costeira is also supported by project FCT - PTDC/EEA-ACR/72201/2006.

Abstract

This paper deals with the reconstruction of smooth, flexible, flat surfaces isometrically embedded in 3D when observed from uncalibrated camera images. It provides a cost function which captures the global geometry of the problem. The input data consists of a sequence of matched feature points (i.e. known correspondence), but the algorithm allows for missing data so that all points do not need to be visible in all images. Its performance under these conditions is analyzed. A way of inferring the bend radius at each point is provided which can be used back in the cost function, providing a higher order fitting. The performance is validated using both synthetic data and real image examples, where the reconstruction is compared to ground truth and reference points.

1 Introduction

The purpose of this paper is to allow reconstruction of deformable surfaces isometrically embedded in 3D, e.g. a flag waving at the wind or someone waving a sheet of paper, from image data. It is assumed that a set of non-calibrated images is available with matched features between them. For simplicity the cameras here are restricted to be scale-orthographic, each modeled by a single scale factor and extrinsic parameters. Figure 1 illustrates the acquisition process. From a single flat surface, modeled as a set of features, 3D isometric embeddings are generated by passing the feature points through embedding functions $\mathscr{I}^k$. These are then viewed by different cameras yielding the observed images.

There are two problems that can be formulated with images of isometrically embedded surfaces. The first consists of estimating the 3D embedding (pose) of the surface in a particular image k, when the flat surface is assumed to be known. In figure 1 this roughly means estimating (b) given (a) and (c). This problem will be called the pose estimation problem, but will not be the focus of this paper. The second problem is given several images estimate the surface that generates them. In the figure this means from several observations (c), obtain (a). Here this problem shall be named the surface estimation problem and the embeddings (b) are not considered important. Although not done yet, a future objective will be to unify

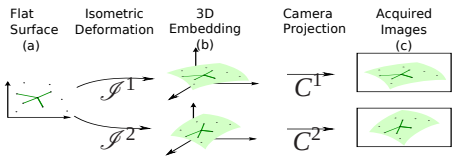

Figure 1: Acquisition model of the isometrically embedded surface observed by a camera.

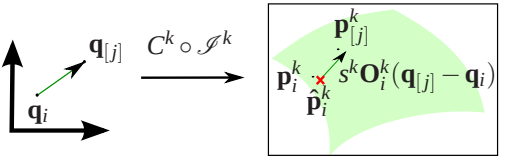

Figure 2: Visualization of the variables involved.

both problems, i.e. estimate the generating surface (a) and the various embeddings (b) given only the set of images (c). This can be trivially achieved by first estimating the surface and then applying a known pose estimation algorithm, future work will focus on integrating both in a single problem.

1.1 Previous Work

The pose estimation problem has been receiving a lot of attention lately. Some works in this area include [10] which describe a closed form solution to the problem of registering a camera observation to an a-priori known model embedded in 3D. In a different approach described in [11] the authors propose to recover 3D structure from a single camera by learning the statistical deformation model of deformable surfaces. Other work (in which the isometry property is not forced) include [7] which attempt to estimate the embedding function by minimizing compound energy and inferring a smoothing parameter and [8]. In [4] the authors propose to use range data information as well. Non-rigid reconstructions have received a lot of attention lately (see for example [1]) but the only known paper that deals directly with the surface estimation problem is [6].

1.2 Contribution

This paper models accurately flexible flat surfaces following a similar strategy presented in [6] where an initial approximation is given by a sequential algorithm. This approximation is then fed to a global cost function, further refining the result and hopefully converging to the global minimum. To better capture global constraints instead of just measuring local fit in which integration error can accumulate a different optimization function is used. Here we deal with the more realistic assumption that data can be missing, which is of utmost importance in deformable surfaces where self occlusion and partial observations are common. This paper handles missing data and provides the required performance tests. Finally, one important contribution is how to obtain second order information about the 3D embeddings, such as how much the surface bends, which can be used for measurements and reused in the cost function. Although the surfaces are assumed to be locally planar, second order information provides relevant information in the presence of a sparser data set, as so often occurs.

In summary, this paper extends the current state of the art in structure from motion of nonrigid flat surfaces by accuratley modeling what happens when surfaces bend and provides a reliable way to estimate the surface.

2 Reconstruction of Flat Surfaces

The 2-D flat surface is sampled as a cloud of point features $\mathcal{Q} = \{\mathbf{q}_i \in \mathbb{R}^2\}$ which generates K distinct 3-D point clouds by the isometric functions $\mathcal{I}^k$ as $\mathcal{I}^k(\mathcal{Q}) = \{\mathcal{I}^k(\mathbf{q}_i) : \mathbf{q}_i \in \mathcal{Q}\}$. An isometric function is here loosely defined as not changing the intrinsic interpoint distance

(see [7] for a more formal definition). For the purpose of this paper this definition is enough, together with the fact that the Jacobian matrix (i.e. derivative matrix) of these functions are Stiefel matrices (i.e. two columns of a 3×3 orthogonal matrix). Each of these sets of points $\mathscr{I}^k(\mathscr{Q})$ correspond to the 3-D points of the bent flag or sheet of paper which are assumed to be observed by cameras. These observations yield the point sets $\mathscr{P}^k = C^k \circ \mathscr{I}^k(\mathscr{Q}) = \{\mathbf{p}_i^k \in \mathbb{R}^2 : \mathbf{p}_i^k = C^k \circ \mathscr{I}^k(\mathbf{q}_i), \mathbf{q}_i \in \mathscr{Q}\}$, where each C^k models a camera projection function (see figure 2). Since camera motion is an isometry, C^k can be reduced to the projection of a canonic camera at the origin.

For a dense enough point cloud, the smooth, locally planar approximation is valid for embeddings with the isometries $\mathscr{I}^k$, preserving extrinsic distance and angles between neighboring points. Note that this is always true for the intrinsic distance, but extrinsically only if the considered points are contained in a planar submanifold. This approximation will be exploited to allow surface reconstruction.

General isometry functions are hard to characterize, so instead of using them directly, the known properties of their push-forwards (in a loose sense, their Jacobian matrices) will be exploited. As stated before, when $\mathscr{I}^k$ are isometries the corresponding push forward at a point $\mathbf{q}_i$ (here denoted as ${\mathscr{I}^k}_{\mathbf{q}_i *}$) are represented as a Stiefel matrix when an orthonormal tangent space basis is chosen. Hence, 2D tangent vectors of $\mathbb{R}^2$ at the same point are sent to 3D vectors by matrix multiplication by this Stiefel matrix. The push forward of the observation function is here represented by $\mathbf{C}^k$. By composition, the transformation of tangent vectors at a point $\mathbf{q}_i$ to tangent vectors in the observed image is given by a 2×2 matrix $\mathbf{S}_i^k = \mathbf{C}^k \circ {\mathscr{I}^k}_{\mathbf{q}_i *}$.

When canonic orthographic cameras are used, the push-forward is $\mathbf{C}^k = \begin{bmatrix}\mathbf{I}_{2\times 2} & \mathbf{0}\end{bmatrix}$ which, when composed with ${\mathscr{I}^k}_{\mathbf{q}_i *}$, results in $\mathbf{S}_i^k$ matrices that are 2×2 submatrices of larger orthogonal 3×3 matrices (i.e. a Stiefel matrix without the bottom row). Henceforth these matrices will be called Sub-Stiefel. Using the Cauchy Interlacing Theorem [8], these can be characterized as the set of 2×2 matrices with the largest singular value equal to 1: $\mathbb{SS} = \left\{\mathbf{S}_{2\times 2} : \begin{bmatrix}\mathbf{S} & * \\ * & *\end{bmatrix} \in \mathbb{SO}(3)\right\} = \{\mathbf{S}_{2\times 2} : \sigma_{\max}(\mathbf{S}) = 1\}$. If canonic scaled orthographic cameras are used a further s^k multiplicative scale factor is needed.

Hence the surface estimation problem is formulated as: estimate the 2D point cloud $\mathscr{Q}$ from multiple observations $\mathscr{P}^k$. Although it is not assumed that all points are visible in all images, it is assumed that their correspondence is known.

The notation $[j]_i$ means the jth neighboring index point of i. If point i has n_i neighbors, then $j \in \{1 \dots n_i\}$. Note that although point $\mathbf{q}_i$ is always a neighbor of point $\mathbf{q}_{[j]_i}$, there is no guarantee that in a particular image this neighbor is seen. Hence some additional care is needed when stating "$\mathbf{p}_i^k$ is a neighbor of point $\mathbf{p}_{[j]_i}^k$". To make notation less dense, using the context there should not be any confusion from dropping the i in the notation hence $[j]$ simply means the jth neighboring index point of the current point i. Since it is usual to cycle through all the visible points, the set $\mathscr{N} = \{(i,j,k) : \mathbf{p}_i^k \text{ and } \mathbf{p}_{[j]}^k \text{ are visible}\}$ is defined. This set is the union of smaller per-image sets $\mathscr{N}^k = \{(i,j) : \mathbf{p}_i^k \text{ and } \mathbf{p}_{[j]}^k \text{ are visible}\}$.

2.1 Cost Function

The objective of this paper is to solve the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{(i,j,k)\in\mathscr{N}} \left\| \mathbf{p}_{[j]}^k - \left(\hat{\mathbf{p}}_i^k + s_i^k \mathbf{O}_i^k(\mathbf{q}_{[j]} - \mathbf{q}_i)\right)\right\|^2 \\ \text{s.t.} \quad & \hat{\mathbf{p}}_i^k \in \mathbb{R}^2,\ \mathbf{O}_i^k \in \mathbb{SS},\ \mathbf{q}_i \in \mathbb{R}^2,\ \{s_i^k\} \in \text{ camera model set} \end{aligned} \tag{1}$$

Here, the variables $\hat{\mathbf{p}}_i^k$ represent the reconstructed image observations (in a noiseless, bendless case, the optimum solution would be $\hat{\mathbf{p}}_i^k = \mathbf{p}_i^k$). The optimization problem roughly states that each neighboring points in each image $\mathbf{p}_{[j]}^k$ must agree with their reconstructions through the aquisition model (current reconstructed point $\hat{\mathbf{p}}_i^k$, Sub-Stiefel matrices $\mathbf{O}_i^k$ and surface estimates $\mathbf{q}_i$). See figure 2.

Here the terms $\hat{\mathbf{p}}_i^k$ are used to impose per-image global constraints, minimizing integration error where the reconstructed surface might be locally correct but starts to skew or slightly distort when viewed globally. These variables also provide an entry point into fusing the pose estimation problem with the surface estimation problem which is still considered future work.

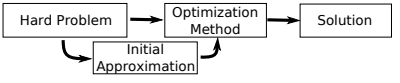

Figure 3: Overview of the optimization strategy. First an initial approximated solution is computed to start the optimization.

The cost function is simple to integrate with the optimization strategy considered in [5], here overviewed for convenience. To solve the highly non-convex cost function presented, first an approximate solution is obtained and later this is used to iterate an optimization algorithm directly on the cost function (see figure 3). The approximate solution is obtained by relaxing the optimization problem and later projecting the variables to their constraint sets. The first simplification is to fix the variables $\hat{\mathbf{p}}_i^k = \mathbf{p}_i^k$. These will be later unfrozen when solving the global cost function.

2.2 Factorization With Missing Data

Dealing with missing data is implemented during the factorization step. The cost function can be simplified so as to allow bilinear factorization techniques to be used:

$$\min \sum_{(i,j,k)\in\mathcal{N}} \left\| (\mathbf{p}_{[j]}^k - \mathbf{p}_i^k) - \hat{\mathbf{O}}_i^k \mathbf{w}_{i[j]} \right\|^2 \quad \text{s.t.} \quad \hat{\mathbf{O}}_i^k \in \mathbb{GL}(2),\ \mathbf{w}_{i[j]} \in \mathbb{R}^2 \tag{2}$$

where the variables $\mathbf{w}_{i[j]}$ relax $\mathbf{q}_{[j]} - \mathbf{q}_i$ and $\hat{\mathbf{O}}_i^k$ relax the constraints $s^k \mathbf{O}_i^k$. Note that all hard-to-solve constraints are effectively eliminated. If missing data is not considered, this can be put in matrix form as

$$\mathbf{V}_i = \begin{bmatrix} \mathbf{v}_{i[1]}^1 & \cdots & \mathbf{v}_{i[n_i]}^1 \\ \vdots & \ddots & \vdots \\ \mathbf{v}_{i[1]}^K & \cdots & \mathbf{v}_{i[n_i]}^K \end{bmatrix} \qquad \hat{\mathbf{O}}_i = \begin{bmatrix} \hat{\mathbf{O}}_i^1 \\ \vdots \\ \hat{\mathbf{O}}_i^K \end{bmatrix} \qquad \mathbf{W}_i = \begin{bmatrix} \mathbf{w}_{i[1]} & \cdots & \mathbf{w}_{i[n_i]} \end{bmatrix}$$

where $\mathbf{v}_{i[j]}^k = \mathbf{p}_{[j]}^k - \mathbf{p}_i^k$ are the available observations. This allows it to be rewritten, decoupled at each point, as:

$$\sum_i \min \left\| \mathbf{V}_i - \hat{\mathbf{O}}_i \mathbf{W}_i \right\|^2 \quad \text{s.t.} \quad \hat{\mathbf{O}}_i \in \mathbb{R}^{2K\times 2},\ \mathbf{W}_i \in \mathbb{R}^{2\times n_i} \tag{3}$$

Rank factorization techniques can be applied directly to this problem. When missing data is considered, matrix $\mathbf{V}_i$ has missing entries. This is a well known problem and several algorithms exist, either that solve it exactly (in some cases) or provide an approximate solution ([3] is a recent survey on the problem). These algorithms usually provide the needed rank factorization as output.

Once a solution $\hat{\mathbf{O}}_i^*, \mathbf{W}_i^*$ ($\mathbf{p}_1$ on figure 4) has been found, it is far from unique. Any set of matrices $\mathbf{G}_i \in \mathbb{GL}(2)$ generates another equally valid solution $\hat{\mathbf{O}}_i^* \mathbf{G}_i^{-1}$ and $\mathbf{G}_i \mathbf{W}_i^*$. The set of all possible solutions generates set $\mathscr{B}$ in the figure. Next, these remaining degrees of freedom will be used to find a solution which is as close as possible to the sets where

$\mathbf{w}_{i[j]} = \mathbf{q}_{[j]} - \mathbf{q}_i$ (set $\mathscr{C}$ in the figure) and the set where $\mathbf{O}_i^k \in \mathbb{SS}$ (set $\mathscr{A}$ in the figure). Here an approximate solution is given in a sequence of steps.

2.3 Shape Consistency

First the shape consistency $\mathbf{w}_{i[j]} = \mathbf{q}_{[j]} - \mathbf{q}_i$ is approximated. This is accomplished by solving the problem

$$\min \sum_{i,j} \|\mathbf{G}_i \mathbf{w}^*_{i[j]} - (\mathbf{q}_{[j]} - \mathbf{q}_i)\|^2_{\mathbb{GL}(2)} \quad \text{s.t.} \quad \|[\mathbf{G}_1 \dots \mathbf{G}_N \ \mathbf{q}_1 \dots \mathbf{q}_N]\| = 1 \tag{4}$$

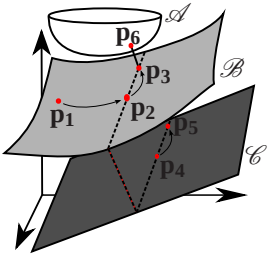

Figure 4: Illustration of the approximation algorithm. The ambient space is the cartesian product $\hat{\mathbf{O}}_i \times \mathbf{W}_i$.

The notation implies a $\mathbb{GL}(2)$ invariant norm in the cost function. The unit norm constraint is needed to eliminate the trivial solution $\mathbf{G}_i = \mathbf{0}$ and $\mathbf{q}_i = 0$. As [5] shows, this is a sparse eigenvalue problem, solvable using adequate software such as MATLAB's `svds`.

In figure 4 we present a graphical sketch of the several steps involved. The solution of this problem moves $\mathbf{p}_1$ to $\mathbf{p}_2$ which represents the point $(\hat{\mathbf{O}}_i^* \mathbf{G}_i^{-1}, \mathbf{G}_i \mathbf{W}_i^*)$. Note that the solution is still in set $\mathscr{B}$, meaning that it is still a valid solution to the problems in equations 2 and 3. This step also computes a first solution to the unfolded variables $\mathbf{q}_i$, in the figure this is represented as point $\mathbf{p}_4$. Note that the solution is still not unique since left multiplying by a single matrix $\mathbf{G} \in \mathbb{GL}(2)$ every $\mathbf{G}_i^*$ and $\mathbf{q}_i^*$, solutions of problem 4, is still a solution. This is depicted in the figure as the dotted lines passing through $\mathbf{p}_2$ and $\mathbf{p}_4$, which are equidistant to each other.

2.4 Model Consistency

Model consistency means forcing matrices $\hat{\mathbf{O}}_i$ as close to obeying the "isometric embedding observed by scale-orthographic cameras model" as close as possible ($\hat{\mathbf{O}}_i = s^k \mathbf{O}_i$, see problem 1). One possibility is to solve the optimization problem

$$\min \sum_{(i,k)} \mathrm{d}^2_{\mathbb{SS}} \left(\hat{\mathbf{O}}_i^{k^*} \ \mathbf{G}/s^k \right) \quad \text{s.t.} \quad \mathbf{G} \in \mathbb{GL}(2), \ s^k \in \mathbb{R} \tag{5}$$

where $\mathrm{d}^2_{\mathbb{SS}}$ denotes a square distance function to the Sub-Stieffel matrix set. Here the function $\mathrm{d}^2_{\mathbb{SS}}(\mathbf{O}) = \log^2 \sigma_{\max}(\mathbf{O})$ is used, as hinted by the Cauchy Interlacing Theorem mentioned earlier, since it provides expected statistical properties to the solution. To solve the problem, a Newton-like optimization algorithm is run, which can tipically provide a solution in 4 or 5 iterations.

Refering back to figure 4, the solution to this problem $\mathbf{G}^*$ moves points $\mathbf{p}_2$ and $\mathbf{p}_4$ to $\mathbf{p}_3$ and $\mathbf{p}_5$ respectively. This cost function does not provide the closest $\mathbf{O}_i^k \in \mathbb{SS}$ which is represented as point $\mathbf{p}_6$ in the figure. This will be important in the next section. Note that the solution is still not unique (a global rotation and scaling of the points is still allowed). This is expected and under scale-orthographic cameras cannot be solved.

2.5 Global Solution

Here an initial approximation to almost all variables is known. To solve the optimization problem 1 a simple coordinate cycling optimization method is used. Variables $\hat{\mathbf{p}}_i^k$ are unfrozen, retaining their initial approximate value. Since the matrices $\mathbf{O}_i^k$ are the only ones not available, the coordinate cycling algorithm starts by computing these.

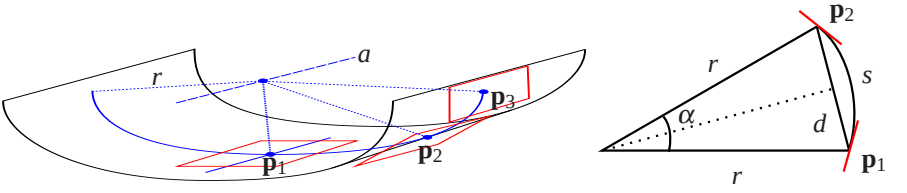

Figure 5: Left: Second order model. At each point there is a direction through which the surface does not bend and another of maximum bend. Here, r denotes the bend radius and a denotes the axis of revolution. The tangent space at each point varies smoothly as the surface bends. Right: Cut along the axis of bend.

Fixing all variables but $\mathbf{O}_i^k$, the problem to be solved is

$$\min \sum_{i,k} \left\| \mathbf{A}_i^k - \mathbf{O}_i^k \mathbf{B}_i^k \right\|^2 \quad \text{s.t.} \quad \mathbf{O}_i^k \in \mathbb{SS} \tag{6}$$

in which matrix $A = [\mathbf{p}_{[1]}^k - \hat{\mathbf{p}}_i^k \dots \mathbf{p}_{[n_i]}^k - \hat{\mathbf{p}}_i^k]$ and matrix $\mathbf{B}_i^k = s_i^k [\mathbf{q}_{[1]} - \mathbf{q}_i \dots \mathbf{q}_{[n_i]} - \mathbf{q}_i]$. The cost function can be decoupled resulting in various smaller problems, each involving a single $\mathbf{O}_i^k$ matrix. This problem is known as the 2×2 Sub-Stiefel Procrustes problem, and is solvable by finding the real roots of a 6 degree polynomial using Gröbner basis techniques applied to the polynomial system of equations resulting from the Karush-Kuhn-Tucker conditions.

Since the cost function is a linear least squares problem in the variables $\hat{\mathbf{p}}_i^k$ and $\mathbf{q}_i$, these can be solved simultaneously. The system is also a linear least squares problem in the variables s^k making it also easy to iterate over these. In the next sections it will be shown how to further improve these estimates by introducing a second order model which takes into account intrinsic characteristics of flat surfaces.

3 Estimating Extrinsic Bend

When a flat surface is embedded in 3-D one important property is verified: through every point of the embedded surface there is a direction where it is locally linear (see for example [9]). This means that through each point, the embedding is allowed to curve in only one direction, making it locally like a cylinder (this is a second order approximation). Since the word "curvature" has a very precise meaning (all the surfaces here considered have 0 curvature), here this embedding-specific second order property shall be called "bend" at a point, and its "bend radius" will be the radius of the smallest osculating circle through that point. In differential geometry this terminology refers to the sectional curvature.

An interesting question that can be posed is if these local cylinders can be estimated and somehow incorporated back into the cost function. This section provides the affirmative answer to this question.

The first key point is to notice that on a cylinder, there is a direction through which the affine tangent spaces are parallel, and orthogonally these affine tangent spaces change the quickest (see figure 5). The second key point is that this rate of change can be measured by the estimated surface and Sub-Stiefel matrices.

The idea is that Sub-Stiefel matrices can be completed into rotation matrices with positive

determinant with only a sign ambiguity

$$\underbrace{\begin{bmatrix} o_{11} & o_{12} \\ o_{21} & o_{22} \end{bmatrix}}_{\text{Sub-Stiefel Matrix}} \Rightarrow \underbrace{\begin{bmatrix} o_{11} & o_{12} & o_{13} \\ o_{21} & o_{22} & o_{23} \\ o_{31} & o_{32} & o_{33} \end{bmatrix}}_{\text{Rotation Matrix 1}} \text{ or } \underbrace{\begin{bmatrix} o_{11} & o_{12} & -o_{13} \\ o_{21} & o_{22} & -o_{23} \\ -o_{31} & -o_{32} & o_{33} \end{bmatrix}}_{\text{Rotation Matrix 2}}$$

by choosing a sign for $o_{31} = \pm\sqrt{1-o_{11}^2-o_{21}^2}$ (if it is zero choose another column). Then there a single choice for o_{32} such that $o_{32}^2 = 1-o_{12}^2-o_{22}^2$ and the second column is orthogonal to the first. The third column is simply the cross product of the first two columns (so as to yield a rotation matrix with positive determinant).

Here the distance between two Sub-Stiefel matrices is defined as the minimum of the 4 possible completion combinations. Hence if $\mathbf{O}$ and $\mathbf{S}$ are Sub-Stiefel, call their 2 possible completions $\mathbf{O}^+$, $\mathbf{O}^-$ and $\mathbf{S}^+$, $\mathbf{S}^-$, so the distance is $\mathbf{d}(\mathbf{O},\mathbf{S}) = \min_{a,b\in\{+,-\}}\left\{\mathbf{d}\left(\mathbf{O}^a,\mathbf{S}^b\right)\right\}$ where the distance between rotation matrices [7] is $\mathbf{d}(\mathbf{R}_1,\mathbf{R}_2) = \operatorname{acos}\left((\operatorname{tr}(\mathbf{R}_1^T\mathbf{R}_2)-1)/2\right)$

Hence for a cylinder, there is a direction where this distance function will be 0 when applied to the corresponding Sub-Stiefel matrices, and an orthogonal direction where this distance function will vary linearly with the distance to the current point. Mathematically it means that at each point i in each image k, there's a vector $\mathbf{b}_i^k$ such that for each neighbor $[j]$

$$\mathbf{d}\left(\mathbf{O}_i^k,\mathbf{O}_{[j]}^k\right) = \left|\mathbf{b}_i^{kT}(\mathbf{q}_{[j]}-\mathbf{q}_i)\right| \Leftrightarrow \mathbf{d}\left(\mathbf{O}_i^k,\mathbf{O}_{[j]}^k\right)^2 = (\mathbf{q}_{[j]}-\mathbf{q}_i)^T\mathbf{b}_i^k\mathbf{b}_i^{kT}\underbrace{(\mathbf{q}_{[j]}-\mathbf{q}_i)}_{\mathbf{w}_{i[j]}}$$

writting $\mathbf{b}_i^k\mathbf{b}_i^{kT} = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \end{bmatrix}$ the former can be rewritten using the kronecker product as

$$\underbrace{\mathbf{d}\left(\mathbf{O}_i^k,\mathbf{O}_{[j]}^k\right)^2}_{D_{i[j]}^k} = \underbrace{\mathbf{w}_{i[j]}^T\otimes\mathbf{w}_{i[j]}^T\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_{i[j]}}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{7}$$

Hence a way of estimating $\mathbf{x} = [x_1, x_2, x_3]^T$ is by the constrained least squares problem (assuming all other variables known):

$$\min \sum_{j:(i,j)\in\mathcal{N}^k}\left\|D_{i[j]}^k - \mathbf{A}_{i[j]}\mathbf{x}\right\|^2 \text{ s.t. } \operatorname{rank}\left(\begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \end{bmatrix}\right) \leq 1 \tag{8}$$

This problem is simply solved by replacing the constraint with $x_1x_3 - x_2^2 = 0$ (zero determinant) and the use of Lagrange multipliers. This yields a 4 polynomial system of equations which can be solved through gröbner basis by finding the zeros of a degree 4 polynomial.

A singular value decomposition on the resulting matrix yields the direction of greatest bend and the rate of change of the tangent plane. It is easy to see that the bend radius is the inverse of the rate of change.

3.1 Cost Function Using Second Order Information

When the manifold bends, the equality that tangent vectors in the image can be approximated by $\mathbf{w}_{i[j]} = \mathbf{p}_{[j]} - \mathbf{p}_i$ is only a first order approximation. In particular, intrinsic vector lengths are always larger than the image extrinsic information, as shown in figure 5, where intrinsic distance is s, while extrinsically it is d (before projection on the camera plane). Here a simple correction is implemented.

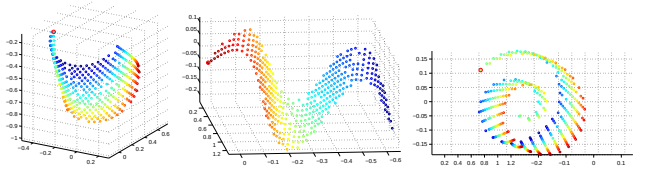

Figure 6: 20×20 grid embedded in 3D by 3 different embedding functions. From the left: half cylinder, sine wave and swiss roll.

As figure 5 shows, along the axis of bend, the following correction is needed:

$$s = \alpha r,\ \sin\left(\frac{\alpha}{2}\right) = \frac{d}{2r} \Longrightarrow d = 2r\sin\left(\frac{s}{2r}\right) \tag{9}$$

By letting $r = 1/\|\mathbf{b}_i^k\|$ as the last section showed, $\hat{\mathbf{b}}_i^k = \mathbf{b}_i^k / r$ and finally $\hat{\mathbf{c}}_i^k = \hat{\mathbf{b}}_i^{k\perp}$ denote the orthogonal complement, vector $\mathbf{w}_{i[j]}$ is written in the bend-directions coordinate system:

$$\mathbf{w}_{i[j]} = \hat{\mathbf{c}}_i^k \underbrace{\hat{\mathbf{c}}_i^{kT}\mathbf{w}_{i[j]}}_{\text{no bend component}} + \hat{\mathbf{b}}_i^k \underbrace{\hat{\mathbf{b}}_i^{kT}\mathbf{w}_{i[j]}}_{\text{bend component}} \tag{10}$$

The bend component needs to be corrected by equation 9. Hence the corrected vector will be

$$\hat{\mathbf{w}}_{i[j]} = \underbrace{\begin{bmatrix}\hat{\mathbf{c}}_i^k & \hat{\mathbf{b}}_i^k\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \frac{2r}{\hat{\mathbf{c}}_i^{kT}\mathbf{v}_{i[j]}}\sin\left(\frac{\hat{\mathbf{c}}_i^{kT}\mathbf{v}_{i[j]}}{2r}\right)\end{bmatrix}\begin{bmatrix}\hat{\mathbf{c}}_i^{kT}\\ \hat{\mathbf{b}}_i^{kT}\end{bmatrix}}_{\mathbf{B}_i^k}\mathbf{w}_{i[j]}$$

where $\mathbf{B}_i^k$ is the bend correcting matrix. This can be used back in the cost function 1 as:

$$\begin{aligned}\min\quad & \textstyle\sum_{(i,j,k)\in\mathcal{N}} \left\|\mathbf{p}_{[j]}^k - (\hat{\mathbf{p}}_i^k + s_i^k\mathbf{O}_i^k\mathbf{B}_i^k(\mathbf{q}_{[j]} - \mathbf{q}_i)\right\|^2\\ \text{s.t.}\quad & \hat{\mathbf{p}}_i^k \in \mathbb{R}^2,\ \mathbf{O}_i^k \in \mathbb{SS},\\ & \mathbf{q}_i \in \mathbb{R}^2,\ \{s_i^k\} \in \text{ camera model set}\end{aligned} \tag{11}$$

A possible optimization strategy is to consider $\mathbf{B}_i^k$ constant at each optimization step (initially equal to the identity matrix) and recompute it at the end of each global optimization loop step.

4 Results

The synthetic data set was generated as a 20×20 2D grid with 0.05 units of space between adjacent points. This grid is then rotated, passed through an embedding function and rotated again in 3D. Figure 6 shows some examples of the embeddings. These embeddings are then orthographically projected (z component removed). For each experiment, gaussian noise with a given standard deviation is added and some points are randomly removed to simulate missing data. Finally the results are scaled (scale factor chosen randomly between 0.5 and 2) to simulate scaled-orthography.

Figure 7 provides a performance comparison in the presence of noise and missing data for 10 input images. The data is run through the algorithm and the final result is compared, using mean squared error, with the original 2D grid. Since the output of the algorithm is only given up to a rotation and scaling, the output is first rotated and scaled to match the original grid as closely as possible. Hence if the original grid is given in a matrix $\mathbf{X}$ (one column per data point) and the output of the algorithm in a matrix $\mathbf{Y}$, the final performance measure

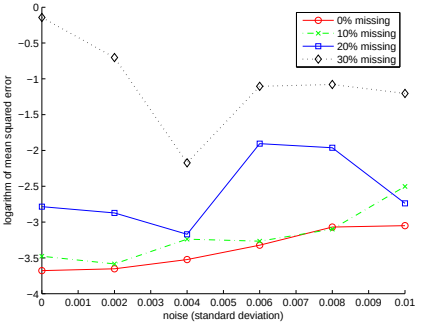

Figure 7: Logarithmic mean squared error as a function of gaussian noise with a given standard deviation and missing data percentage. Each test is performed with 10 synthetic images and 10 neighbors. Noise and outlier are chosen randomly before each experiment.

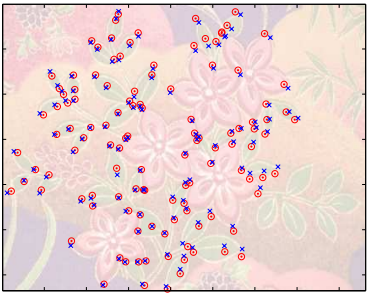

Figure 8: Reconstruction of the bed cover cloth overlaid on an image taken of the cloth laying flat. Blue crosses are the reference clicked points, red circles are the results given by the algorithm.

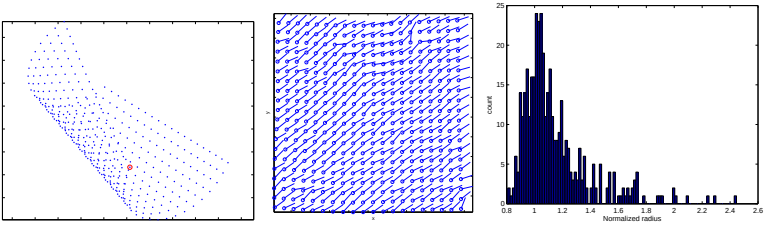

Figure 9: Results of estimating the bending axis and radius from the data. Left: Image to be measured. Middle: The computed axis of bend at each point is shown. Right: histogram of the computed radius at each point (the radius is normalized to be 1 unit).

function is:

$$\begin{array}{ll} \min & \log \frac{1}{N} \left\| \mathbf{X} - s\mathbf{R}\mathbf{Y} \right\|^2 \\ \text{s.t.} & s \in \mathbb{R},\ \mathbf{R} \in \mathbb{SO}(2) \end{array} \tag{12}$$

As the results show, gaussian noise does not affect performance as much as missing data. The results also show that 10%

In a different experiment, bend radius and axis are computed for an embedding wrapped around a cylinder (bend radius at each point is 1). Figure 9 shows the obtained results. Note that although at most points the bend axis is similar, there are a few outliers. The histogram shows that bend radius estimates concentrate around the correct value but some noise exist.

In a real world example, 12 images of a bed cover were taken (see figure 10) at various angles and differently folded. In these images, 118 different points were hand clicked (when visible) and the algorithm was run on them. The results obtained are shown in figure 8 (overlaid on a picture taken of the flat cloth fabric). Results provide a benchmark for real world data, hand clicked, not very dense for the amount of bending, and for embeddings not truly obeying the isometric properties since cloth is easily sheared.

5 Conclusions

This paper describes how to reconstruct a non-rigid surface isometrically embedded in 3D that is observed by multiple images with occlusion. It also describes how to measure the

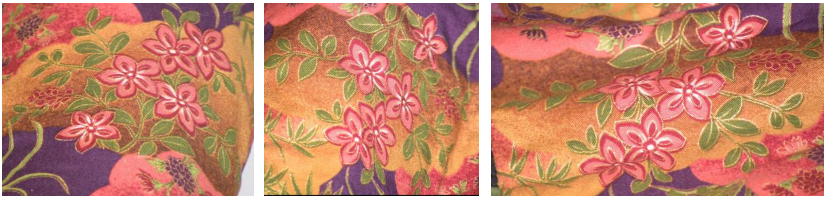

Figure 10: Images (3 out of 12) taken of a bed cover made of cloth.

amount of bend at each point. Results are provided for different noise levels and amount of missing data. Also, a real world example is provided for evaluation. As stated in the text, a future direction is to fuse the pose estimation problem with the surface estimation problem.

References

[1] Ijaz Akhter, Yaser Ajmal Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. In *Neural Information Processing Systems*, December 2008.

[2] Adrien Bartoli. Maximizing the predictivity of smooth deformable image warps through cross-validation. *Journal of Mathematical Imaging and Vision*, 31(2-3):133–145, Jul 2008.

[3] Pei Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *Int. J. Comput. Vision*, 80(1):125–142, 2008. ISSN 0920-5691. doi: http://dx.doi.org/10.1007/s11263-008-0135-7.

[4] Toby Collins, Adrien Bartoli, and Robert Fisher. Automatic quasi-isometric surface recovery and registration from 4d range data. *Proceedings of the BMVA Symposium on 3D Video - Analysis, Display and Applications*, Feb 2008.

[5] Ricardo Ferreira, João Xavier, and João Costeira. Reconstruction of isometrically deformable flat surfaces in 3d from multiple camera images. *ICASSP*, 2009.

[6] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[7] John M. Lee. *Riemannian Manifolds: An Introduction to Curvature*. Springer, 1997.

[8] Mathieu Perriollat, Richard Hartley, and Adrien Bartoli. Monocular template-based reconstruction of inextensible surfaces. *BMVC*, 2008.

[9] Helmut Pottmann and Johannes Wallner. *Computational Line Geometry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001. ISBN 3540420584.

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

[11] Mathieu Salzmann, Raquel Urtasun, and Pascal Fua. Local deformation models for monocular 3d shape recovery. *CVPR*, 2008.