RECONSTRUCTION OF ISOMETRICALLY DEFORMABLE FLAT SURFACES IN 3D FROM MULTIPLE CAMERA IMAGES

R. Ferreira, J. Xavier, J. Costeira

Instituto de Sistemas e Robótica, Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisboa
ricardo@isr.ist.utl.pt, jxavier@isr.ist.utl.pt, jpc@isr.ist.utl.pt

ABSTRACT

This paper deals with the reconstruction of smooth, flexible, isometrically embedded flat surfaces in 3D, such as a sheet of paper or a flag waving in the wind, from a set of 2D projected observations such as camera images. To solve the problem, a set of matched features of the waving object at different poses is needed, which are then applied to the reconstruction algorithm here described. The complete algorithm consists of 2 phases, the first obtaining an initial approximation from local features, the second uses this result to iterate a global cost function, trying to achieve a better estimate. To validate the algorithm, synthetic data with noise is generated, reconstructed and compared to ground truth data. Also, a second experiment consisting of real images of a sheet of paper is shown.

Index Terms— Machine vision, Isometric Non-Rigid Reconstruction, Manifold Learning

1. INTRODUCTION

This paper provides a solution to the problem of reconstructing isometrically embedded flat surfaces in 3D from a set of partially calibrated images where only intrinsic parameters are known. The proposed algorithm is inspired from manifold learning techniques and can be used in this setting as well.

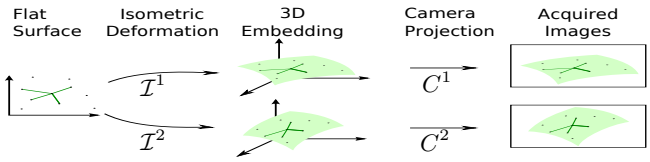
The motivating application here is to infer the structure of a non-rigid isometric surface observed in multiple images. The example provided is to reconstruct a waving sheet of textured paper (or a flag) observed in multiple images.

Prior work in a similar area, inferring the 3D embedding from an observed image assuming the surface model to be known, includes [1] which provide a closed form solution to the problem of registering a camera observation to an a-priori known model embedded in 3D. Also in [2] the authors propose to learn the statistical deformation model of deformable surfaces and use the gained knowledge to recover 3D structure from a single camera, usable even in low texture settings. Other work include [3] which propose to estimate smooth image wraps (not necessarily isometric) by minimizing compound energy while inferring a smoothing parameter as well. To our knowledge, this is the first work that attempts to learn the underlying surface from multiple camera observations.

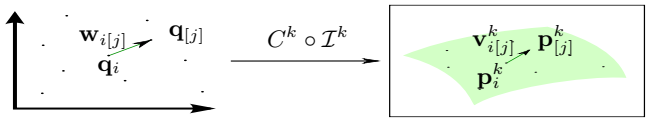
In a seemingly unrelated branch of non linear dimensionality reduction, manifold learning attempts to infer low dimensional structure from very high dimensionality data (see for example [4] for a survey). Here the problem has a different description, where an embedded object is assumed to be completely known a-priori, but represented in such high dimensionality as to be awkward to use. Hence the need to describe the same object (with little loss of information) in a lower dimensional space sometimes with some added characteristics such as linearity. Here the concepts of isometry and smooth embedding [5] are often used to justify the methods.

The paper is structured as to first provide a mathematical description of the problem in section 2, then describes a fast method to obtain an initial suboptimal solution in section 3. The global optimization is described in section 4. Finally, results are provided in section 5 and conclusions are drawn.

2. MATHEMATICAL FORMULATION



(a) Acquisition model.



(b) Close up and notation used.

Fig. 1. Here $\mathcal{I}^i : \mathbb{R}^2 \Rightarrow \mathbb{R}^3$ are nonlinear embedding isometries and $C^i : \mathbb{R}^3 \Rightarrow \mathbb{R}^2$ are camera projections. Image tangent vectors are represented by $\mathbf{v}^k_{ij}$, while $\mathbf{w}_{ij}$ is the corresponding reconstructed tangent vector.

Mathematically the 2-D flat manifold ($\mathbb{R}^2$) is approximated by a cloud of point features $\mathcal{Q} = \left\{\mathbf{q}_i \in \mathbb{R}^2\right\}$. A set of K embedding isometric functions (here loosely defined as functions that do not change the intrinsic distance between points, see [5] for a more rigorous definition) $\mathcal{I}^k : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ yield 3-D point clouds $\mathcal{I}^k(\mathcal{Q}) = \left\{\mathcal{I}^k(\mathbf{q}_i) : \mathbf{q}_i \in \mathcal{Q}\right\}$. These 3-D point clouds are assumed to be observed by cameras, yielding the observations $\mathcal{P}^k = C^k \circ \mathcal{I}^k(\mathcal{Q})$, where C^k is the camera projection function (see figure 1.a). Note that camera motion is an isometry hence can be absorbed by $\mathcal{I}^k$ allowing C^k to be seen as the projection to a canonic camera at the origin.

From these observations, the shape estimation problem is formulated: estimate the 2D point cloud $\mathcal{Q}$ from multiple observations $\mathcal{P}^k$. It can be formulated using orthographic cameras, scaled orthographic cameras or other cameras, but it is assumed that the point correspondence between each $\mathcal{P}^k$ is known. This is considered to be a different problem and is not dealt with here. Also, this paper assumes all points are visible in all images.

Assuming the point cloud is dense enough so that a locally planar approximation is possible, the extrinsic distance and angles between neighboring points is approximately preserved by the isometries $\mathcal{I}^k$. Note that this is always true for the intrinsic distance, but for the extrinsic distance it is only true if the considered points are contained in a planar submanifold. This is the approximation that

will be exploited to allow reconstruction of the objects. Since general isometry functions are hard to characterize, the known properties of their push-forwards (in a loose sense, their "derivatives") will be exploited instead. The key observation is that when $\mathcal{I}^k$ is an isometry then the corresponding push forward at a point $\mathbf{q}_i$ (here denoted as $\mathcal{I}^k{}_{\mathbf{q}_i *}$) is represented as a Stiefel matrix (i.e. two columns of a 3×3 orthogonal matrix) in any orthonormal base. 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 $\mathbf{S}_i^k = \mathbf{C}^k \circ \mathcal{I}^k{}_{\mathbf{q}_i *}$.

When the cameras are orthographic $\mathbf{C}^k = \begin{bmatrix}\mathbf{I}_{2\times 2} & \mathbf{0}\end{bmatrix}$ these $\mathbf{S}_i^k$ matrices are 2×2 submatrices of larger orthogonal 3×3 matrices (i.e. a Stiefel matrix without the bottom row). Henceforth these particular matrices will be designated as Sub-Stiefel. Using the Cauchy Interlacing Theorem [6], these matrices can be characterized as the set of 2×2 matrices with the largest singular value equal to 1: $\mathbb{SS} = \left\{\mathbf{S}_{2\times 2} : \begin{bmatrix}\mathbf{S} & * \\ * & *\end{bmatrix} \in \mathbb{SO}(3)\right\} = \{\mathbf{S}_{2\times 2} : \sigma_{\max}(\mathbf{S}) = 1\}$. These matrices will play an important role in the next sections.

The objective will be to solve the following optimization problem, where the notation $[j]$ means the jth neighboring index point of i (there should not be any confusion from dropping the i in the notation):

$$\begin{array}{ll}\min & \sum_{i,j,k} \left\| \overbrace{(\mathbf{p}_{[j]}^k - \mathbf{p}_i^k)}^{\mathbf{v}_{i[j]}^k} - s_i^k \mathbf{O}_i^k \overbrace{(\mathbf{q}_{[j]} - \mathbf{q}_i)}^{\mathbf{w}_{i[j]}} \right\|^2 \\ \text{s.t.} & \mathbf{O}_i^k \in \mathbb{SS},\ \mathbf{q}_i \in \mathbb{R}^2,\ \{s_i^k\} \in \text{ camera model set}\end{array} \tag{1}$$

which roughly states that at each ith point in the kth image, there must exist a Sub-Stiefel matrix describing its neighbors $[j]$. Please see figure 1.b for a visual description of the variables. Here, for orthographic cameras the third constraint should be $s_i^k = 1$ (as described above), for scaled orthography it should read $s_i^k = s_j^k\ \forall k \in \{1...K\}$ and for a less constrained camera (even more than a paraperspective camera) these parameters are free $s_i^k \in \mathbb{R}$.

Considering the objective function and set of constraints the problem is not trivial to solve even in the simplest orthographic camera case. The approach taken is to first obtain a "good enough" (later defined) approximation and then use an iterative optimization algorithm to improve the solution.

3. SUB-OPTIMAL SOLUTION TO ISOMETRIC RECONSTRUCTION OF FLEXIBLE MANIFOLDS

This section provides a way to compute an initial approximation for the problem in equation (1). The full problem is broken in several much simpler subproblems, chained together to obtain the final result. This paper describes an approach that is applicable to several types of cameras, and degrades gracefully in the presence of noise.

To describe how an approximate solution is obtained the problem is broken into 4 chained sub-problems, described individually in the next sub sections: **1.** Discover local neighbors from the observed images, that is, for each i, build the index set $\{[j]\}$; **2.** Use bilinear factorization to freeze some degrees of freedom in problem (1); **3.** Impose the shape consistency constraint $\mathbf{w}_{ij} \approx \mathbf{q}_i - \mathbf{q}_i$ where the approximation is properly defined in the next sections; **4.** Impose the model consistency constraint $\mathbf{O}_i^k \in \mathbb{SS}$ by freezing the remaining degrees of freedom not previously used.

3.1 Neighbor Estimation. The formulation presented hints at the need to find local neighbors. This problem is not trivial since the actual distances between points is not known, only the distance between camera projected points. The problem is that if the surface appears folded in an image, some intrinsically far points might seem close together. The converse holds more information though: as long as there is a single image observing a pair of points as not being near each other they'll be excluded as neighbors. So a simple algorithm for neighbor estimation might be summarized as: a pair of points are considered neighbors as long as they're observed in all images within a certain ball, which can vary from image to image, and from point to point. These balls can be chosen to make sure a certain minimum number of neighbors exist for each point.

Please note that the output of this algorithm does not guarantee that all returned points are actual neighbors, but it seems to provide good results, especially as the number of images increases, minimizing the chance of a "collapsed" area of the surface in every image. In this paper, the results were obtained by imposing that each point have 8 neighbors.

3.2 Bilinear Factorization. As shown in [7], bilinear factorization is a powerfull tool used to solve many engineering problems. To apply the factorization method, the constraints will be temporarily relaxed, which also allows the s_i^k to be fused with the $\mathbf{O}_i^k$ matrices, yielding the much simpler problem:

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

When all neighbors of all points are seen in every images, grouping variables pointwise in observation, model and shape matrices

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

where $\mathbf{v}_{i[j]}^k = \mathbf{p}_{[j]}^k - \mathbf{p}_i^k$ as in problem (1) and n_i is the number of neighbors of point i, allows the problem to be rewritten, decoupled at each point:

$$\sum_i \left(\begin{array}{ll}\min & \left\| \mathbf{V}_i - \hat{\mathbf{O}}_i \mathbf{W}_i \right\|^2 \\ \text{s.t.} & \hat{\mathbf{O}}_i \in \mathbb{R}^{2K\times 2},\ \mathbf{W}_i \in \mathbb{R}^{2\times n_i}\end{array} \right) \tag{3}$$

Here rank factorization techniques apply directly, yielding a pair of $\hat{\mathbf{O}}_i^*$ and $\mathbf{W}_i^*$ matrices for each point. Since the solution is not unique (for any matrices $\mathbf{G}_i \in \mathbb{GL}(2)$, $\hat{\mathbf{O}}_i^* \mathbf{G}_i^{-1}$ and $\mathbf{G}_i \mathbf{W}_i^*$ are also solutions), there are some degrees of freedom that still need to be fixed. These will be used to approximate the discarded constraints.

3.3 Shape Consistency. This section will use the degrees of freedom left in matrices $\mathbf{G}_i$ to approximate, in a later defined sense, the constraints $\mathbf{w}_{i[j]} = \mathbf{q}_{[j]} - \mathbf{q}_i$ that were relaxed between problem formulations 1 and 2. Up to now, the solution is not anchored in space, consisting only of sets of "free" vectors. Furthermore, these vectors are not consistent with each other, in the sense that $\mathbf{w}_{ij} \neq -\mathbf{w}_{ji}$ whenever both vectors exist and that $\mathbf{w}_{ik} \neq \mathbf{w}_{ij} + \mathbf{w}_{jk}$ whenever these 3 vectors exist (i.e. whenever the corresponding points are neighbors).

Defining the surface neighbor vector matrices as

$$\mathbf{Q}_i = \begin{bmatrix}\mathbf{q}_{[1]} - \mathbf{q}_i, & \mathbf{q}_{[2]} - \mathbf{q}_i, & ..., & \mathbf{q}_{[n_i]} - \mathbf{q}_i\end{bmatrix}$$

(notice the neighbor $[\cdot]$ notation). Imposing shape consistency means finding matrices $\mathbf{G}_i$ and points $\mathbf{q}_i$ such that $\mathbf{G}_i \mathbf{W}_i^* - \mathbf{Q}_i \approx \mathbf{0}$. The left side is clearly linear on a larger matrix containing all points $\mathbf{Q} = [\mathbf{q}_1\ \mathbf{q}_2 ... \mathbf{q}_N]$ and matrices $\mathbf{G}_i$, hinting at the possibility of existing matrices $\mathbf{A}_i$ and $\mathbf{X} = [\mathbf{Q}, \mathbf{G}_1, \mathbf{G}_2 \dots \mathbf{G}_N]$ such that the previous equation can be written as $\mathbf{X}\mathbf{A}_i \approx \mathbf{0}$. These matrixes in fact exist.

There's an additional property that needs to be understood. It stems from the fact that if a matrix $\mathbf{X}$ is given that satisfies $\mathbf{X}\mathbf{A}_i =$

$\mathbf{0}$, then any pre-multiplication of this by another matrix $\mathbf{G}$ is also a solution: $\mathbf{GXA}_i = \mathbf{0}$. This means that the problem is ill determined, and that the best that can be done is provide a solution up to a global linear transformation. There is also a trivial solution which consists of making the lines of $\mathbf{Q}$ constant (i.e. all points the same) and $\mathbf{G}_i = \mathbf{0}$, implying that matrices $\mathbf{A}_i$ have a kernel with eigenvector $[\mathbf{1}_{1\times N},\ \mathbf{0}_{1\times 2N}]^T$. This trivial solution must be excluded by allowing only solutions on the orthogonal complement of this constraint. With this in mind, an optimization problem can be formulated as

$$\begin{array}{ll} \min & \sum_i \|\mathbf{X}\mathbf{A}_i\|^2_{\mathbb{GL}(2)} \\ \text{s.t.} & \mathbf{X} \in \mathbb{R}^{2\times 3N},\ \mathbf{X}[\mathbf{1}_{1\times N}\ \mathbf{0}_{1\times 2N}]^T = 0 \end{array} \tag{4}$$

Here the norm subscript is used to hint that this must be a $\mathbb{GL}(2)$ left-invariant function. This problem reduces to a sparse eigenvalue problem of a sparse symmetric matrix when a standard left invariant function with a simple solution that serves the purpose is used: $\|\mathbf{X}\mathbf{A}_i\|^2_{\mathbb{GL}(2)} = \text{tr}\left\{\mathbf{A}_i^T\mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T\right)^{-1}\mathbf{X}\mathbf{A}_i\right\}$. Using the property of the trace function $\text{tr}(AB) = \text{tr}(AB)$, the change of variables $\mathbf{Y} = \left(\mathbf{X}\mathbf{X}^T\right)^{-1/2}\mathbf{X}$, and the fact that the function is $\mathbb{GL}(2)$ left invariant, the problem results in a sparse eigenvalue problem:

$$\begin{array}{ll} \min & \text{tr}\left\{\mathbf{Y}A\mathbf{Y}^T\right\} \\ \text{s.t.} & \mathbf{Y}\mathbf{Y}^T = \mathbf{I}, \mathbf{Y} \in \mathbb{R}^{2\times 3N}, \mathbf{Y}[\mathbf{1}_{1\times N}\ \mathbf{0}_{1\times 2N}]^T = \mathbf{0} \end{array} \tag{5}$$

This is exactly the formulation problem of the second and third eigenvalue of a symmetric matrix $\mathbf{A} = \sum_i \mathbf{A}_i\mathbf{A}_i^T$ with a least eigenvector known to be $[\mathbf{1}_{1\times N}\ \mathbf{0}_{1\times 2N}]^T$ (as is the case). This can be obtained efficiently with available software taking into account the sparsity of matrix $\mathbf{A}$ (see for example [8]).

Since $\mathbf{Y}$ is obtained from $\mathbf{X}$ by left multiplication by a matrix, and since the cost function is left invariant, $\mathbf{X}^* = \mathbf{Y}^*$ is a solution to the original problem. Matrix $\mathbf{X}^*$ contains all matrices $\mathbf{Q}^*$ and $\mathbf{G}_i^*$ as desired. Matrix $\mathbf{Q}^*$ contains an initial embedding of the points, up to a $\mathbb{GL}(2)$ transformation that will be computed next. From here on it is assumed that the computed $\mathbf{G}_i^*$ have been merged into $\hat{\mathbf{O}}_i^*$ and $\mathbf{W}_i^*$.

3.4 Camera Model Consistency. Bear in mind that there is still a global $\mathbf{G} \in \mathbb{GL}(2)$ ambiguity which, along with the merged s_i^k constants in $\hat{\mathbf{O}}_i^*$ will be used to "straighten the axes" by imposing the still ignored camera acquisition model. The idea is to search for a $\mathbf{G} \in \mathbb{GL}(2)$ matrix that forces the matrices $\hat{\mathbf{O}}_i^*\mathbf{G}/s_i^k$ to minimize some sort of distance $d(\cdot)$ to the Sub-Stiefel matrix set. Due to the characterization given previously, we know that this $d(\cdot)$ should depend on the maximum singular value of its argument, but other than this (and the fact that it should be a distance function) there is no naturally given choice of function, the best that can be done to narrow the choice is impose desired properties for the solution. The following optimization problem is proposed, to force a set of given matrices $\mathbf{S}_i$ to be as close as possible to the Sub-Stiefel set by right multiplication by a common $\mathbf{G}$ (there are strong group theoretic and statistical arguments to use this function):

$$\begin{array}{ll} \min & \sum_i \log^2\left(\sigma_{\max}\left(\mathbf{S}_i\,\mathbf{G}\right)\right) \\ \text{s.t.} & \mathbf{G} \in \mathbb{GL}(2) \end{array} \tag{6}$$

where $\sigma_{\max}(\cdot)$ returns the largest singular value.

This problem can be reduced to an optimization problem in the real projective plane $\mathbb{RP}^2$, which makes it significantly easier to solve, since it is a compact two dimensional differentiable manifold, definitely within the reach of branch and bound algorithms if nothing better is possible. Maybe equally important is that the function can now be visualized, allowing intuition and a clearer idea of how hard it is to solve. Unfortunately sometimes it shows 2 different local minima. Despite this shortcoming, when used to solve the problem at hand it does produce seemingly good results without much concern over which local minimum is used (usually the minima are very close together, indistinguishable when the function is seen globally).

Since $\sigma_{\max}$ is a smooth function of its matrix argument almost everywhere (it is non-smooth when both singular values are equal), gradient vector and Hessian are computable for every function involved. It is relatively straightforward to implement a Newton-like method on the projective space (see for example [9]) with only minor care to avoid the non-smoothness. Note that the minimizer should only be at the non-smoothness with probability 0 (unless artificial examples are used).

Since the maximum eigenvalue function is invariant to rotations, this cost function provides a solution up to a global rotation. This is to be expected since no global referential has been imposed.

Looking back, up to this section a set of matrices $\hat{\mathbf{O}}_k^{i^*}$ (already multiplied by $\mathbf{G}_i^{*-1}$ computed in the previous section) have been computed. These matrices are known up to a global $\mathbf{G}$ matrix. When equation (2) was written, the variables s_i^k were fused with the $\mathbf{O}_i^k$ matrices: $\mathbf{O}_i^k = \hat{\mathbf{O}}_i^k/s_i^k$. These are the matrices that need to be as close to Sub-Stiefel as possible:

$$\begin{array}{ll} \min & \sum_{(i,k)} d^2_{\mathbb{SS}}\left(\hat{\mathbf{O}}_i^{k^*}\,\mathbf{G}/s_i^k\right) \\ \text{s.t.} & \mathbf{G} \in \mathbb{GL}(2),\ \{s_i^k\} \in \text{ camera model set} \end{array}$$

When simple orthographic cameras are used ($s_i^k = 1$) the problem assumes the exact form as problem 6. Interestingly, the same trick used to describe the problem as an optimization problem in projective space can be re-used to allow use of more complicated camera models. When scaled orthographic cameras are used, $s_i^k = s_j^k$ hence there's a unique scale factor s^k for each image:

$$\begin{array}{ll} \min & \sum_k \left(\begin{array}{ll} \min & \sum_i d^2_{\mathbb{SS}}\left(\mathbf{O}_i^{k^*}\,\mathbf{G}/s^k\right) \\ \text{s.t.} & s^k \in \mathbb{R} \end{array} \right) \\ \text{s.t.} & \mathbf{G} \in \mathbb{GL}(2) \end{array}$$

The only change is that there's an additional scale ambiguity in the final reconstruction (adding to the global rotation matrix). This is expected when using scaled orthography.

If the camera model is the least constrained possible $s_i^k \in \mathbb{R}$, the problem is trivial, with solution $s_i^k = \sigma_{max}(\hat{\mathbf{O}}_i^{k*})$. Since $\mathbf{G}$ is not used, this is the global ambiguity. Hence the solution obtained in the previous section is the best that can be hoped for.

4. GLOBAL OPTIMIZATION

From the beginning the objective has been to obtain a solution for problem 1. While the previous section has provided an initial approximation to this problem, it does so through a series of simplifications. This section now proposes to take the previous solution and actually achieve a (possible local) minimizer for the problem. Since the problem is not smooth ($\mathbb{SS}$ is not a smooth manifold), it is hard to implement even a gradient descent method. Here we propose to take a much simpler approach based on coordinate cycling, solving a conceptually important sub-problem in the process.

Taking into account that the previous section obtained approximate solutions $\mathbf{O}_i^{k*}$, $\mathbf{q}_i$ and s_i^k, the proposal is to iteratively obtain a better estimate for one set of these variables, while keeping the remaining ones fixed. Since of these 3 sets the $\mathbf{O}_i^{k*}$ are the only ones that do not obey the constraints (the previous section only approximated them), this is the set of variables that shall be used to start the iterative process.

4.1 Solving for $\mathbf{O}_i^k$. Grouping the problems variables in index j, similarly to what was done in problem 3, and fixing all variables except $\mathbf{O}_i^k$, the subproblem to be solved is

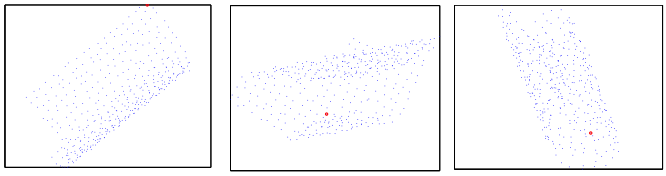
$$\begin{array}{ll} \min & \sum_{i,k} \left\| \mathbf{V}_i^k - \mathbf{O}_i^k \hat{\mathbf{W}}_i^k \right\|^2 \\ \text{s.t.} & \mathbf{O}_i^k \in \mathbb{SS} \end{array} \tag{7}$$

where here $\hat{\mathbf{W}}_i^k = s_i^k \mathbf{W}_i$ absorves s_i^k and $\mathbf{W}_i$ is recomputed from the current $\mathbf{q}_i$. Notice that the terms are not related, hence this can be broken into many subproblems, each involving a single $\mathbf{O}_i^k$ matrix. This problem will be known as the 2×2 Sub-Stiefel Procrustes problem. Although not presented here, its solution can be reduced to finding the real roots of a 6 degree polynomial, obtained through a "once for all time" (data independent) computation of a Gröbner basis from Algebraic Geometry.

4.2 Solving for $\mathbf{q}_i$ and s_i^k. The cost function is linear in each of these variables hence they can be iterated using simple least square regression techniques.

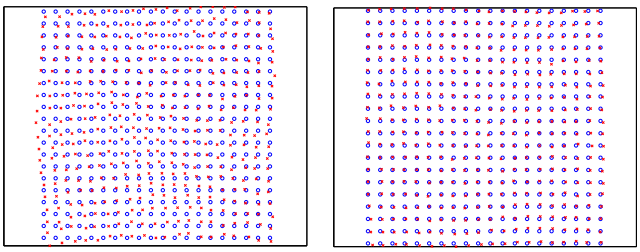
5. RESULTS

Two experiment sets are shown, one with synthetic noisy data, the other with hand clicked real images. In both cases the 8 closest neighbors were used. In the synthetic image case (see figures 2 and 3) a set of 18 images were generated similar to the ones shown, with a significant amount of gaussian noise added (standard deviation about half the intergrid distance). These images were then fed to the algorithm in 2 runs, one where only 6 images were used, the other using all 18 images. The results were then rotated and multiplied by a scalar (global ambiguity of using scaled orthographic cameras) to best fit the ground truth data. Notice that the final reconstructions are close to the ground truth image, without any noticeable shearing. As expected, the number of images helps reduce the amount of noise in the reconstruction.



(a) half cylinder (b) sine wave (c) swiss roll

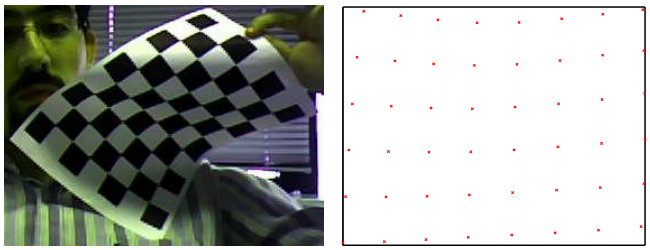
Fig. 2. Input data wrapped around different shapes. The set contains 18 images similar to the ones shown.



(a) 6 Images. (a) 18 Images.

Fig. 3. Result of the algorithm (red crosses) after applying 5 global iterations. Ground truth (without noise) is provided in blue circles for comparison.

In the real image case, 7 images were obtained, using a laptop webcam, at different distances from a waving sheet of paper with an easily identifiable printed pattern. The images were then hand clicked and the points were fed to the algorithm. The results obtained are shown in figure 4. Note that globally the reconstruction appears to have a slight pinch in the middle probably due to some bias in the distortions applied to the sheet of paper; with hand made distortions the central part is usually the most curved.



(a) Acquired image. (a) Results obtained.

Fig. 4. Results of applying the algorithm to 7 camera acquired images.

6. CONCLUSION

An algorithm for reconstructing flat surfaces from images of 3D isometric embeddings of flat surfaces is described. The algorithm consists of two separate phases, first obtaining an approximate solution and then improving it with a global algorithm. Validation was provided, demonstrating that the algorithm produces the expected results.

In the future we hope to improve the global algorithm, by integrating second order information using neighboring Sub-Stiefel matrices. This would help correct the pinching effect seen in some results, at a higher computational cost. Making the algorithm work with only partially visible data is also a priority.

Due to space requirements, some ideas were left without proof, to be included in a followup paper in preparation.

7. REFERENCES

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

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

[3] Adrien Bartoli, "Maximizing the predictivity of smooth deformable image warps through cross-validation," *Journal of Mathematical Imaging and Vision*, vol. 31, no. 2-3, pp. 133–145, Jul 2008.

[4] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik, "Dimensionality reduction: A comparative review," *Neurocomputing - submitted*, 2008.

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

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

[7] Carlo Tomasi and Takeo Kanade, "Shape and motion from image streams: A factorization method," *Int'l J. Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[8] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Society for Industrial and Applied Mathematics, 2000.

[9] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, January 2008.