

Structure from Motion

ToF-aided 3D Reconstruction of Isometric Surfaces

S. Jafar Hosseini and Helder Araujo

*Institute of Systems and Robotics, Department of Electrical and Computer Engineering,
University of Coimbra, Coimbra, Portugal*

Keywords: Structure from Motion, Isometric Surface, ToF Camera, 3D Reconstruction.

Abstract: This paper deals with structure-from-motion (SfM) for non-rigid surfaces that undergo isometric motion. Our SfM framework aims at the joint estimation of the 3D surface and the camera motion by combining a ToF range sensor and a monocular RGB camera through a template-based approach. Our goal is to use the 2D low-resolution depth estimates provided by the TOF camera, in order to facilitate the estimation of non-rigid structure using the high-resolution images obtained by means of a RGB camera. In this paper, we model isometric surfaces with a triangular mesh. The ToF sensor is used to obtain the depth of a sparse set of 3D feature points, from which the depth of the mesh vertices can be recovered using a multivariate linear system. Subsequently, we form a non-linear constraint based on the projected length of each edge. A second non-linear constraint is then used for minimizing re-projection errors. These constraints are finally incorporated into an optimization scheme to solve for structure and motion. Experimental results show that the proposed approach has good performance even if only a low-resolution depth image is used.

1 INTRODUCTION

Structure-from-motion can be defined as the problem of simultaneous inference of the motion of a camera and the 3D geometry of the scene solely from a sequence of images. SfM was also extended to the case of deformable objects. Non-rigid SfM is under-constrained, which means that the recovery of non-rigid 3D shape is an inherently ambiguous problem (Paladini et al., 2009; Dai et al., 2012). Given a specific configuration of points on the image plane, different 3D non-rigid shapes and camera motions can be found that fit the measurements. To solve this ambiguity, prior knowledge on the shape and motion should be used to constrain the solution. For example, Aanaes et al. (Aans and Kahl, 2002) impose the prior knowledge that the reconstructed shape does not vary much from frame to frame while Del Bue et al. (Del-Bue et al., 2006) impose the constraint that some of the points on the object are rigid. The priors can be divided in two main categories: the statistical and the physical priors. For instance, the methods relying on the low-rank factorization paradigm (Aans and Kahl, 2002; Del-Bue et al., 2006) can be classified as statistical approaches. Learning approaches such as (Zhou et al., 2012; Salzmann et al., 2007; Srivastava et al., 2009; Gay-Bellile et al., 2006) also belong

to the statistical approaches. Physical constraints include spatial and temporal priors on the surface to reconstruct (Gumerov et al., 2004; Prasad et al., 2006). A physical prior of particular interest is the hypothesis of having an inextensible (i.e. isometric) surface (Shen et al., 2010; Perriollat et al., 2010; Salzmann et al., 2008). In this paper, we consider this type of surface. This hypothesis means that the length of the geodesics between every two points on the surface should not change across time, which makes sense for many types of material such as paper and some types of fabric.

3D reconstruction of non-rigid surfaces from images is an under-constrained problem and many different kinds of priors have been introduced to restrict the space of possible shapes to a manageable size. Based on the type of the surface model (or representation) used, we can classify the algorithms for reconstruction of deformable surfaces. The point-wise methods only reconstruct the 3D position of a relatively small number of feature points resulting in a sparse reconstruction of the 3D surface (Perriollat et al., 2010). Physics-based models such as superquadrics (Metaxas and Terzopoulos, 1993), triangular meshes (Salzmann et al., 2008) or Thin-Plate Splines (TPS) (Perriollat et al., 2010) have been also utilized in other algorithms. In TPS, the 3D surface is

represented as a parametric 2D-3D map between the template image space and the 3D space. Then, a parametric model is fit to a sparse set of reconstructed 3D points in order to obtain a smooth surface which is not actually used in the 3D reconstruction process.

There has been increasing interest in learning techniques that build surface deformation models from training data. More recently, linear models have been learned for SfM applications (Torresani et al., 2003; Llado et al., 2005). There has also been a number of attempts at performing 3D surface reconstruction without using a deformation model. One approach is to use lighting information in addition to texture clues to constrain the reconstruction process (White and Forsyth, 2006), which has only been demonstrated under very restrictive assumptions on lighting conditions and is therefore not generally applicable.

A common assumption in deformable surface reconstruction is to consider that the surface is inextensible.

In (Perriollat et al., 2010), the authors propose a dedicated algorithm that enforces the inextensibility constraints. However, the inextensibility constraint alone is not sufficient to reconstruct the surface. Another sort of implementation is given by (Salzmann and Fua, 2007; Salzmann et al., 2008). In these papers, a convex cost function combining the depth of the reconstructed points and the negative of the reprojection error is maximized while enforcing the inequality constraints arising from the surface inextensibility. The resulting formulation can be easily turned into a SOCP problem. A similar approach is explored in (Shen et al., 2010). The approach of (Perriollat et al., 2010) is a point-wise method. The approaches of (Salzmann and Fua, 2007; Salzmann et al., 2008; Shen et al., 2010) use a triangular mesh as surface model, and the inextensibility constraints are applied to the vertices of the mesh.

1.1 Model and Approach

In this work, we aim at the combined inference of the 3D surface and the camera motion while preserving the geodesics by using a RGB camera aided by a ToF range sensor. Usually, RGB cameras have high image resolutions. With these cameras, one can use efficient algorithms to calculate the depth of the scene, recover object shape or reveal structure, but at a high computational cost. ToF cameras deliver depth map of the scene in real-time but with insufficient resolution for some applications. So, a combination of a common camera and a ToF sensor can exploit the capabilities of both. We assume that the fields of view of both the RGB and ToF cameras mostly overlap. The goal of the algorithm is to allow the 3D reconstruction when

matching is difficult and depth estimates are available for a limited number of points on the surface. The developed approach performs SfM under the constraint that the deformation be isometric.

1.2 Outline of the Paper

This paper is organized as follows: to model an isometric surface, a triangular mesh as well as a planar reference configuration is used. In Section 3, the matching between data from the range and RGB cameras is described. Next, the estimation of the depth of the mesh vertices based on the depth of the feature points is described. The entire approach for estimation of the 3D shape and motion is based on minimizing the sum of both the re-projection errors and the errors on the projected length of the mesh edges. Experimental results and quantitative evaluation are presented in the last section. We show that our approach is able to handle the isometry constraint indirectly without having to directly apply this constraint. In addition, it obviates the need for a dense set of 3D points lying on the surface by effective use of a ToF sensor.

2 NOTATION AND BACKGROUND

2.1 Notation

Matrices are represented as bold capital letters ($\mathbf{A} \in \mathbb{R}^{n \times m}$, n rows and m columns). Vectors are represented as bold small letters ($\mathbf{a} \in \mathbb{R}^n$, n elements). By default, a vector is considered a column. Small letters (a) represent one dimensional elements. By default, the j th column vector of \mathbf{A} is specified as \mathbf{a}_j . The j th element of a vector \mathbf{a} is written as a_j . The element of \mathbf{A} in the row i and column j is represented as $A_{i,j}$. $\mathbf{A}^{(1:2)}$ and $\mathbf{a}^{(1:2)}$ indicate the first 2 rows of \mathbf{A} and \mathbf{a} . $\mathbf{A}^{(3)}$ and $\mathbf{a}^{(3)}$ denote the third row of \mathbf{A} and \mathbf{a} , respectively. Regular capital letters (A) indicate one dimensional constants. We use \mathbb{R} after a vector or matrix to denote that it is represented up to a scale factor.

2.2 Barycentric Coordinates

In geometry, the barycentric coordinate system is a coordinate system in which the location of a point of a simplex (a triangle, tetrahedron, etc.) is specified as the center of mass, or barycenter, of masses placed at its vertices.

3 COMBINING DEPTH AND RGB IMAGES

3.1 Mapping Between Depth and RGB Images

The resolutions of the depth and RGB images are different. A major issue that directly arises from the difference in resolution is that a pixel-to-pixel correspondence between the two images can not be established even if the FOVs fully overlap. Therefore the two images have to be registered so that the mapping between the pixels in the ToF image and in the RGB image can be established. The depth map provided by the ToF camera is sparse and affected by errors. Several methods can be used to improve the resolution of the depth images (Diebel and Thrun, 2005; Kim et al., 2009; Yang et al., 2007; Kim et al., 2011) allowing the estimation of a dense depth image. We will use a simple approach based on linear interpolation.

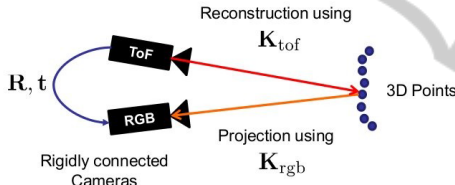


Figure 1: RGB/ToF camera setup.

To estimate depth for all the pixels of the RGB image, based on the depth provided by the ToF camera, a simple linear approach is used. We assume that the relative pose between both cameras, specified by the rotation matrix \mathbf{R}' and translation vector \mathbf{t}' has been estimated. We also assume that both cameras are internally calibrated, i.e., their intrinsic parameters are known. Let \mathbf{p}_{tof} and \mathbf{p}_{rgb} represent the 3D coordinates of a 3D point in the coordinate system of the ToF and RGB cameras, respectively.

We use a pinhole camera model for both the RGB and ToF cameras. Assume that the relative pose of the RGB camera and ToF sensor is fixed with a rotation \mathbf{R}' and a translation \mathbf{t}' : $\mathbf{p}_{rgb} = \mathbf{R}' \mathbf{p}_{tof} + \mathbf{t}'$ as shown in Figure 1. The point cloud \mathbf{p}_{tof} is obtained directly from the calibrated ToF camera. Since the relative pose is known as well as the intrinsic parameters for both cameras, \mathbf{p}_{rgb} can be obtained from \mathbf{p}_{tof} . To estimate depth for all points of the RGB image, a simple linear interpolation procedure is used. For each 2D point of the RGB image, we select the 4 closest neighbors whose depth was obtained from the depth image. Then, a bilinear interpolation is performed.

Another possibility would be to select the 3 closest neighboring points (therefore, defining a triangle) and assume that the corresponding 3D points define a plane. An estimate for the depth of the point could then be obtained by intersecting its projecting ray with the 3D plane defined by the three 3D points.

3.2 Recovery of the Mesh Depth

Given a sparse set of 3D feature points $\mathbf{p}^{ref} = \{\mathbf{p}_1^{ref}, \dots, \mathbf{p}_N^{ref}\}$ on a reference template with a known shape (usually a flat surface), and a set of 2D image points $\mathbf{q} = \{\mathbf{q}_1, \dots, \mathbf{q}_N\}$ tracked on the RGB input image of the same surface but with a different and unknown deformation. As already stated, we represent the surface as a triangulated 3D mesh with n_v vertices \mathbf{v}_i (and n_{tr} triangles) concatenated in a vector $\mathbf{s} = [\mathbf{v}_1^T, \dots, \mathbf{v}_{n_v}^T]^T$, and denote by \mathbf{s}^{ref} the reference mesh, and \mathbf{s} the mesh we seek to recover. Let \mathbf{p}_i be a feature point on the mesh \mathbf{s} corresponding to the point \mathbf{p}_i^{ref} in the reference configuration. We can express \mathbf{p}_i in terms of the barycentric coordinates of the triangle it belongs to:

$$\mathbf{p}_i = \sum_{j=1}^3 a_{ij} \mathbf{v}_j^{[i]} \quad (1)$$

where the a_{ij} are the barycentric coordinates and $\mathbf{v}_j^{[i]}$ are the vertices of the triangle containing the point \mathbf{p}_i . Since we are dealing with rigid triangles, these barycentric coordinates remain constant for each point and can be easily computed from points \mathbf{p}_i^{ref} and the mesh \mathbf{s}^{ref} . Let us denote by $\mathbf{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_N\}$ the set of barycentric coordinates associated to the 3D feature points, where $\mathbf{a}_i = [a_{i1}, a_{i2}, a_{i3}]$. The rigidity of a triangle enforces that the sum of the relative depths around a closed triangle be zero. Assuming that the depth of the vertices of a triangle is denoted as $v_{z,1}$, $v_{z,2}$ and $v_{z,3}$, we have: $(v_{z,1} - v_{z,2}) + (v_{z,2} - v_{z,3}) + (v_{z,3} - v_{z,1}) = 0$. Substituting $(v_{z,1} - v_{z,2})$, $(v_{z,2} - v_{z,3})$ and $(v_{z,3} - v_{z,1})$ for r_{z1} , r_{z2} and r_{z3} , respectively, which denote the relative depth of the edges of the triangle, we can represent the above equation differently as: $r_{z1} + r_{z2} + r_{z3} = 0$ where $r_{z1} = v_{z,1} - v_{z,2}$, $r_{z2} = v_{z,2} - v_{z,3}$, and $r_{z3} = v_{z,3} - v_{z,1}$. Having the above equations for any triangle of the mesh makes a total of $n_{tr} + n_e$ (the number of triangles + the number of edges) linear equations which can be jointly expressed as $\mathbf{M}_1_{(n_{tr}+n_e) \times (n_v+n_e)} \mathbf{x}_1_{(n_v+n_e) \times 1} = \mathbf{0}$. This homogeneous system of equations must be satisfied at each time instant (i.e. for any deformation). However, finding a unique solution is not possible. More

specifically, \mathbf{M}_1 is rank-deficient by n_v , that is, it does not have $n_v + n_e$ linearly independent columns ($\text{rank}(\mathbf{M}_1) = n_e$). So, there will be a n_v -dimensional basis for the solution space to $\mathbf{M}_1 \mathbf{x}_1 = 0$. Any solution is a linear combination of basis vectors. In order to constrain the solution space and determine just one solution out of the infinite possibilities, in a way that this linear system matches only one particular deformation, it is necessary to add n_v independent equations. To add additional constraints, we augment this system with the z coordinate of few properly distributed feature points in this arrangement: using the method described in the previous section, we can obtain an estimate for the depth of a feature point i , indicated by $p_{z,j}$. From the Equation 1, we can derive

$$p_{z,i} = a_{i1}v_{z,1}^{[i]} + a_{i2}v_{z,2}^{[i]} + a_{i3}v_{z,3}^{[i]}.$$

This non-homogeneous system of equations can be represented as $\mathbf{M}_{2N \times n_v} \mathbf{x}_{2n_v \times 1} = \mathbf{p}_z$. It can be verified

that $\mathbf{x}_1 = \begin{bmatrix} \mathbf{r}_z \\ \mathbf{x}_2 \end{bmatrix}$. \mathbf{r}_z is a n_e -vector of the relative depth of the edges. Having the above equation for any feature point results in N linear independent equations. Putting together both sets of equations just explained, we end up with $n_{tot} = n_{tr} + n_e + N$ linear equations ($\mathbf{M}\mathbf{x}_1 = \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_z \end{bmatrix}$) where the only unknowns are the depth of the vertices and of the edges (i.e. $n_v + n_e$ unknowns), which means that the resulting linear system is overdetermined. In fact, we obtain $n_e + N$ independent equations out of n_{tot} equations. Yet, this is not enough to find the right single solution because there are still an infinitude of further solutions that minimize $\left\| \mathbf{M}\mathbf{x}_1 - \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_z \end{bmatrix} \right\|$ in the least-squares sense. One possible approach after the 3D coordinates are estimated is to fit an initial surface using cubic spline data interpolation, to the data which consists in xy -coordinates of the feature points on the reference configuration as input and their z -coordinates on the input deformation as output. Once the parameters of the interpolant have been found, we can obtain initial estimates of depth for the vertices, with their xy -coordinates on the reference configuration as input. The interpolated depth has proved to be very close to the correct one. Then, we add an equality constraint for each vertex as $\mathbf{I}_{n_v \times n_v} \mathbf{x}_2 = \mathbf{v}'_z$ (\mathbf{v}'_z is the interpolated depth of the vertices). The new linear system $\mathbf{M}_{new} \mathbf{x}_1 = \mathbf{b}$ has most likely full column-rank. So, the number of independent equations out of $n_{tot} + n_v$ equations would be $n_e + n_v$. Since the number of independent equations is equal to the number of unknowns, there must be a unique solution, which can be computed via the normal equations. In principle, finding the least-squares estimate is recommended.

4 GLOBAL METRIC ESTIMATION OF STRUCTURE AND MOTION

Next we describe two non-linear constraints applied to the estimation problem. These two constraints are used to solve for SfM so that metric reconstruction of the shape is achieved and the motion matrices lie on the appropriate motion manifold. Furthermore, when there are too few correspondences without additional knowledge (as is the case here), shape recovery would not be effective. So, we need to limit the space of possible shapes by applying a deformation model. This model adequately fills in the missing information while being flexible enough to allow reconstruction of complex deformations (Salzmann et al., 2007). We assume we can model the mesh deformation as a linear combination of a mean shape \mathbf{s}_0 and n_m basis shapes (deformation modes) $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_{n_m}]$:

$$\mathbf{s} = \mathbf{s}_0 + \sum_{k=1}^{n_m} w_k \mathbf{s}_k = \mathbf{s}_0 + \mathbf{S}\mathbf{w} \quad (2)$$

4.1 Constraint 1: Projected Length

Assume that the RGB camera motion relative to the world coordinate system is expressed as a rotation matrix \mathbf{R} and a translation vector \mathbf{t} . A common approach to solve for the camera motion and surface structure is to minimize the image re-projection error, namely by bundle adjustment. The cost function being minimized is the geometric distance between the image points and the re-projected points. However, we are going to adapt bundle adjustment to our own problem rather than use it directly, as follows: the errors to be minimized will be the difference between the observed and predicted projected lengths of an edge.

Orthographic Camera: Under orthographic projection, if we assume that the mesh vertices are registered with respect to the image centroid, we can drop the translation vector. The modified formulation of bundle adjustment can be specified as the following non-linear constraint:

$$e_{pl} = \sum_{i=1}^{n_e} \left(l_i - \left\| \mathbf{R}^{(1:2)} \left[\mathbf{s}_1^{[i]} - \mathbf{s}_2^{[i]} \right] \right\| \right)^2 \quad (3)$$

where the leftmost term is the measurement (observation) of the projected length of an edge. (the computation of l_i is trivial with the help of estimated mesh depth) n_e is the number of edges. $\mathbf{s}_1^{[i]}$ and $\mathbf{s}_2^{[i]}$ denote 2 entries of the mesh, which account for the ending

vertices of the edge i . e_{pl} can be also expressed as a quadratic function.

Perspective Camera: In this case, we formulate a non-linear constraint based on what we call "unnormlized projected length", as:

$$e_{pl} = \sum_{i=1}^{n_e} \left(l_i - \left\| \mathbf{K}_{rgb}^{\circ} [\mathbf{R} | \mathbf{t}] \left[\begin{array}{c} \mathbf{s}_1^{[i]} \\ 1 \end{array} \right] - \left[\begin{array}{c} \mathbf{s}_2^{[i]} \\ 1 \end{array} \right] \right\| \right)^2 \quad (4)$$

where \mathbf{K}_{rgb}° is a known calibration matrix equivalent to $\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$. From the estimated mesh depth, l_i can be easily measured using simple mathematical manipulation. Since there is a subtraction in the above cost function, the translation vector \mathbf{t} can be removed. Also, note that the 2-norm is applied to the first 2 entries of a 3-vector to estimate the square of unnormlized projected length. So, only the 2 first rows of the product of $\mathbf{K}_{rgb}^{\circ} \cdot \mathbf{R}$ are involved in the constraint:

$$e_{pl} = \sum_{i=1}^{n_e} \left(l_i - \left\| \mathbf{f}^{[i]}(\mathbf{R}^{(1:2)}, \mathbf{w}) \right\| \right)^2 \quad (5)$$

4.2 Constraint 2: Reprojection Error

Several difficulties may affect the estimation of the depths namely:

- Errors due to the depth interpolation;
- Irregular distribution of the feature points over the object surface.

As a result of these factors, the depth estimate for the mesh vertices may be significantly inaccurate. In addition, there are also reprojection errors, that is, errors on the image positions of the 3D feature points. We should thus account for the reprojection error by adding a term to the function to be optimized. By combining Equations 1 and 2, we'll have:

$$\mathbf{p}_i = \sum_{j=1}^3 a_{ij} (\mathbf{s}_{0j}^{[i]} + \mathbf{S}_j^{[i]} \mathbf{w}) \quad (6)$$

where $\mathbf{s}_{0j}^{[i]}$ and $\mathbf{S}_j^{[i]}$ are the subvector of \mathbf{s}_0 and the submatrix of \mathbf{S} (respectively), corresponding to the vertex j of the triangle in which the feature point i resides. The term corresponding to the reprojection error can be obtained as indicated below.

Orthographic Camera:

$$e_{ba} = \sum_{i=1}^N \left\| \mathbf{q}_i - \mathbf{R}^{(1:2)} \mathbf{p}_i \right\|^2 \quad (7)$$

Perspective Camera:

$$e_{ba} = \sum_{i=1}^N \left\| \lambda_i \begin{bmatrix} \mathbf{q}_i \\ 1 \end{bmatrix} - \left[\mathbf{K}_{rgb}^{\circ} [\mathbf{R} | \mathbf{t}] \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} \right] \right\|^2 \quad (8)$$

The projective depths λ_i can be determined using the estimated depth for feature's image points on the RGB image. Subsequently, errors in λ_i (induced by the first condition mentioned above) would introduce false search directions in the e_{re} -based minimization problem. Therefore, it is advantageous to reformulate the above equations so that λ_i is removed from them. So, we take into account the equation below:

$$\lambda_i \begin{bmatrix} q_i \\ 1 \end{bmatrix} = \mathbf{K}_{rgb}^{\circ} \left[\sum_{j=1}^3 a_{ij} \mathbf{R} \cdot \mathbf{v}_j^{[i]} \right] + \mathbf{K}_{rgb}^{\circ} \cdot \mathbf{t} \quad (9)$$

After some simple algebraic manipulation and replacing the vertices with the linear deformation mode, we obtain:

$$\begin{bmatrix} a_{i1} \mathbf{A}_i & a_{i2} \mathbf{A}_i & a_{i3} \mathbf{A}_i \end{bmatrix}_{2 \times 9} \begin{bmatrix} \mathbf{R} \cdot \mathbf{s}_1^{[i]} \\ \mathbf{R} \cdot \mathbf{s}_2^{[i]} \\ \mathbf{R} \cdot \mathbf{s}_3^{[i]} \end{bmatrix}_{9 \times 1} + \mathbf{A}_i \cdot \mathbf{t} =$$

$$\begin{bmatrix} g1^{[i]}(\mathbf{R}, \mathbf{w}, \mathbf{t}) \\ g2^{[i]}(\mathbf{R}, \mathbf{w}, \mathbf{t}) \end{bmatrix}_{2 \times 1} = 0 \text{ where } \mathbf{A}_i = \mathbf{K}_{rgb}^{\circ(1:2)} - \mathbf{q}_i \cdot \mathbf{K}_{rgb}^{\circ(3)} \quad (10)$$

This equation provides 2 linear constraints as: $g1^{[i]}(\cdot) = 0$ and $g2^{[i]}(\cdot) = 0$. Thus, the modified e_{re} takes a form free of λ_i as follows: $e_{mre} = \sum_{i=1}^N (g1^{[i]}(\cdot)^2 + g2^{[i]}(\cdot)^2)$, where e_{mre} denotes the modified e_{re} . e_{pl} is a function of $\mathbf{R}^{(1:2)}$ and \mathbf{w} whereas e_{mre} (or e_{re}) is a function of \mathbf{R} , \mathbf{w} and \mathbf{t} . In order to simplify e_{pl} , we modify it by considering that: 1- the translation vector \mathbf{t} is fixed and the camera setup has only rotational movement relative to the world coordinate system. 2- adding the following function to $\mathbf{f}^{[i]}(\mathbf{R}^{(1:2)}, \mathbf{w})$ in the first constraint, we are able to solve for the full matrix \mathbf{R} :

$$f_{rz}^{[i]}(\mathbf{R}^{(3)}, \mathbf{w}) = \left(\mathbf{R}^{(3)} \left[\mathbf{s}_1^{[i]} - \mathbf{s}_2^{[i]} \right] \right) \quad (11)$$

$$e_{rz} = rz_i - f_{rz}^{[i]}(\mathbf{R}^{(3)}, \mathbf{w}) \quad (12)$$

where $rz_i = v_{z,1}^{[i]} - v_{z,2}^{[i]}$. e_{rz} is actually the difference between the observed and predicted relative depths of edge i . Combining $\mathbf{f}^{[i]}(\cdot)$ and $f_{rz}^{[i]}(\cdot)$, it yields:

$$e_{mpl} = \sum_{i=1}^{n_e} \left(\sqrt{(l_i^2 + rz_i^2)} - \left\| \begin{bmatrix} \mathbf{f}^{[i]}(\mathbf{R}^{(1:2)}, \mathbf{w}) \\ f_{rz}^{[i]}(\mathbf{R}^{(3)}, \mathbf{w}) \end{bmatrix} \right\| \right)^2 \quad (13)$$

where e_{mpl} represents a modified version of e_{pl} . As a result, we brought e_{mpl} and e_{mre} into a common form where both are functions of \mathbf{R} and \mathbf{w} .

4.3 Objective Function

So far, we have derived two constraints expressed as two separate non-linear problems. However, we intend to integrate both constraints into one single objective function so that they are taken into account at once, when estimating all the parameters. To do so, we minimize the weighted summation of them in such a way that the reprojection error term is assigned a weight m that accounts for its relative influence within the combined objective function. A block diagram of the overall structure of the approach is demonstrated in Figure 2. In our global optimization, we first consider a simplified formulation of the objective function by excluding the camera motion $[\mathbf{R}|\mathbf{t}]$. We include it back in the second case.

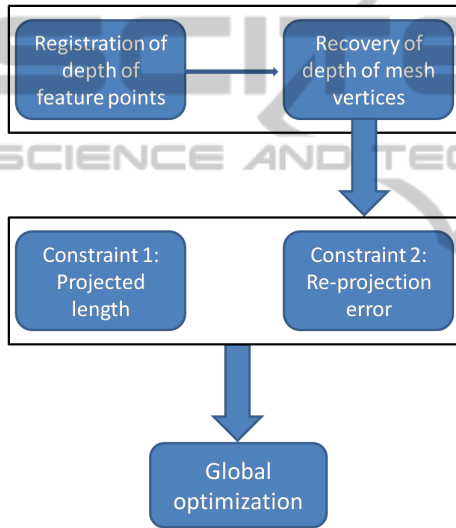


Figure 2: Representation of the approach via block diagram.

4.3.1 Estimation of Structure Only

The constraints are simplified so that the only unknown parameter is the structure (we assume that the camera motion is set to $[\mathbf{I}|0]$).

Orthographic Camera: $\min_{\mathbf{w}} e_{tot} = (e_{pl} + m.e_{re})$

Perspective Camera: $\min_{\mathbf{w}} e_{tot} = (e_{mpl} + m.e_{mre})$

4.3.2 Estimation of Both Structure and Camera Motion

We consider now the full optimization by including the camera motion.

Orthographic Camera: $\min_{\mathbf{R}^{(1:2)}, \mathbf{w}} e_{tot} = (e_{pl} + m.e_{re})$

Perspective Camera: $\min_{\mathbf{R}, \mathbf{w}} e_{tot} = (e_{mpl} + m.e_{mre})$

The above optimization problems can be solved using a non-linear minimization algorithm such as

Levenberg-Marquardt (LMA). The rotation estimates obtained from this optimization may not satisfy the orthogonality constraints. So, the optimization algorithm must be fed with a good initialization. To provide initial estimates relatively close to the true ones, we do the following: if initial guesses for $R^{(1:2)}$ and R are not given, they can be initialized using well-known methods that attempt to solve for SfM through non-rigid factorization of $\{q_{ij}\}$ and $\{\lambda_{ij}q_{ij}\}$ from all frames, for instance, as in (Llado et al., 2005). In these methods, the factorization is followed by a refinement step to upgrade the reconstruction to metric. The deformation coefficients \mathbf{w}_k are initialised to random small values. One possible solution to further meet the rotation constraints is to subsequently apply Procrustes (Akhter et al., 2009; Xiao et al., 2004).

4.4 Additional Constraint

Non-linear optimization may converge to local minima. The probability of such occurrence can be reduced by adding a new regularization term that requires the estimated depth data to be as close to the measured one as possible. So, we would have:

$$e_z = \sum_{i=1}^{n_v} \left(v_z^{[i]} - \left(\mathbf{R}^{(3)} \mathbf{s}^{[i]} + \mathbf{t}^{(3)} \right) \right)^2 \quad (14)$$

where $v_z^{[i]}$ is the depth of the vertex i , already recovered and $\mathbf{s}^{[i]}$ is the 3D position corresponding to the vertex i . Notice that this regularization is very dependent on the accuracy of $v_z^{[i]}$.

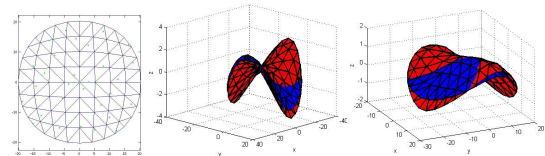


Figure 3: Left: A 9×9 template mesh with sparse feature points - Radius = 20 cm. Right: Metric coordinates in cm - Overlap between the ground-truth shapes (blue) and the recovered ones (red).

5 EXPERIMENTS

5.1 Synthetic Data

Next, we evaluate the methods described above using synthetic data. We synthesized a number of frames of a deforming circle-like paper (radius = 20 cm) approximated by a 9×9 mesh such as the one shown in Figure 2. The reason to use a circular mesh is that it is uniform and has a symmetric shape. Therefore,

it has similar shapes (up to a rotation) for a number of different deformations, which, in fact, brings more complexity to the reconstruction of the right deformation. The inextensible meshes used for training have been built using Blender and PCA was then applied to estimate the deformation model. In order to generate the input data, we get a sparse set of 3D feature points ($N = 32$) well-distributed on the surface of a reference planar mesh. The camera configuration is set up in a way that makes the FOV of the ToF camera be part of the FOV of the 2D camera. The experiments are repeated equally for both the orthographic and perspective cameras. For the perspective case, the camera model is defined such that the focal length is $f = 500$ pixels. The model assumes that the surface is located 50 cm in front of the cameras (along the optical axis). The 3D feature points across the surface are then projected onto the 2D camera and a zero-mean Gaussian noise with 1-pixel standard deviation (Std) was then added to these projections. The depth data of feature points is also generated by adding a zero-mean Gaussian noise with 0.1 cm Std. The results of the quantitative assessment represent an average obtained from five deformations randomly selected. By performing 50 trials for each deformation, each average value was acquired from 250 trials. Two of the estimated deformations and their equivalent ground-truth are qualitatively illustrated in Figure 3.

Table 1: Preliminary results.

Reconstruction error	PRE	MRE	RotationAccuracy
Our approach - Orthographic	0.0608	0.0755	0.002
Our approach - Perspective	0.0603	0.0751	$< 1^\circ$

5.1.1 Reconstruction Error

The accuracy of the method is reported in terms of reconstruction errors. The reconstruction errors are computed with respect to two measures as:

1- Point reconstruction error (PRE): The normalized Euclidean distance between the observed ($\hat{\mathbf{p}}_i$) and estimated (\mathbf{p}_i) world points according to $PRE = \frac{1}{N} \sum_{i=1}^N \left[\frac{\|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2}{\|\hat{\mathbf{p}}_i\|^2} \right]$.

2- Mesh reconstruction error (MRE): The normalized Euclidean distance between the observed ($\hat{\mathbf{v}}_i$) and estimated (\mathbf{v}_i) mesh vertices, which is computed as $MRE = \frac{1}{n_v} \sum_{i=1}^{n_v} \left[\frac{\|\mathbf{v}_i - \hat{\mathbf{v}}_i\|^2}{\|\hat{\mathbf{v}}_i\|^2} \right]$.

The reprojection error of the feature points can be also regarded as another measure of precision. The accuracy of the Stiefel rotation matrix is evaluated based on the orthonormality constraint as $RotationAccuracy = \|\mathbf{R}^{(2 \times 3)} \mathbf{R}^{(2 \times 3)T} - \mathbf{I}^{(2 \times 2)}\|_F^2$. In case of the perspective camera, we compare the axis-angle of the recovered and ground-truth rotations as $RotationAccuracy = |angle - angle|$. Our approach takes into consideration just few feature

points, though we take advantage of the ToF sensor to get the depth of them. We have to notice that the pattern of placement of these points on the surface is of high importance and we need to examine which patterns would yield the best results.

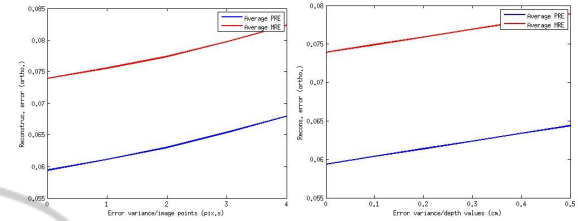


Figure 4: Orthographic camera - Left: Average PRE and average MRE with respect to the increasing noise in image points. Right: Average PRE and average MRE with respect to the increasing noise in depth data.

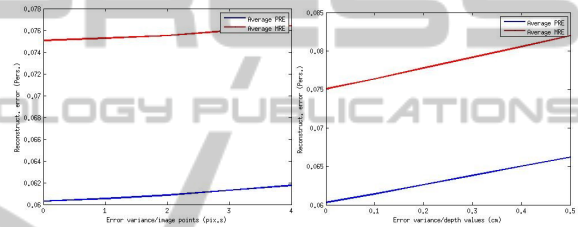


Figure 5: Perspective camera - Left: Average PRE and average MRE with respect to the increasing noise in image points. Right: Average PRE and average MRE with respect to the increasing noise in depth data.

5.1.2 Length of the Edges

When a 3D surface is reconstructed in a truly inextensible way, the length of the recovered edges must be the same as that of the template edges. So, in order to see to what extent the lengths remain the same along the deformation path, we specify a metric to figure out the discrepancy between the initial and recovered lengths as: $IsometryExtent = \left(1 - \left(\frac{1}{n_e} \sum_{i=1}^{n_e} \left(\frac{|L_i - \hat{L}_i|}{\hat{L}_i} \right) \right) \right) \times 100\%$ which has been found to be 95.77% for the proposed method, which indicates that it preserves the length of the edges greatly, confirming that isometry constraint is satisfied to a large degree.

5.1.3 The Impact of Noise

Different levels of noise (whether in image points or in depth data) have been simulated to demonstrate how robustly the approach reacts to the noise. Each of these 2 types of noise has been investigated separately. Figures 4 and 5 illustrate results for increasing levels of Gaussian noise in feature's image points, where the Std varied from 0 to 4 pixels with 1-pixel increments, together with the reconstruction error for

various levels of Gaussian noise in depth of feature points, with $0.1 - cm$ increments of Std, which was computed following the remark that, since the depth variation of the surface itself is small, the deviations from the true depth of every 3D point may be very close together, varying at each trial according to a Gaussian distribution. From the Figures 4 and 5, we may draw the conclusion that the white noise does not make a dramatic impact on the output, ensuring that the performance remains pretty stable and the algorithm carries on efficiently in the face of noise.

5.2 Real Data

We performed also experiments with real data recorded using a camera setup comprising a ToF camera and a RGB camera. The camera setup was calibrated both internally and externally. Bilinear interpolation was applied to estimate the depth of each 2D point track. We used a piece of cardboard to make real inextensible deformations and proceeded with the tracking and matching of few feature points with respect to the reference template using SIFT local feature descriptor. The same deformation model as the one acquired in synthetic experiments was employed. Some deformations and their recovered shapes are shown in Figure 7. Although it was not possible to quantitatively assess the results and do benchmarking, the efficiency of the approach was visible from the 3D reconstruction output.

5.2.1 Comparative Evaluations using Motion Capture Data

Rather than generate the training data synthetically using Blender, we take advantage of datasets recorded using Vicon which is able to capture real deformations accurately. Since the synthetically deformed meshes might not exactly overlap the real deformations, we rebuilt the deformation model based on this real data and redid the experiments. The template configuration is now composed of equal triangles and covers a 20×20 -cm square-like area. As an example, the reconstructed surfaces in Figure 7 look better than the ones in Figure 6. Consequently, when learned with real data, the deformation model would be more robust to the deformations.

As a general rule, two different entities can be compared only when they meet identical conditions which characterize them. To this end we analyzed the state-of-art literature and selected the approach described in (Salzmann et al., 2007). In particular this approach also uses a triangular mesh and can use the same types of data sets required by our approach. As a result, to show how the real training data will influence the 3D

reconstruction, we performed a set of simulations as we already did with Blender data and we compare the performance of our SfM framework to this approach (where the authors use a second-order cone program (SOCP) to accomplish the 3D reconstruction of inextensible surfaces). Their approach is known to be very robust and efficient, where a linear local deformation model integrates local patches into a global surface and requires many feature points distributed throughout the surface. To account for noise in our approach, like before, a Gaussian noise with 1-pixel Std was added to the image points and a Gaussian noise with $0.1 - cm$ Std to the depth data. The SOCP-based approach was evaluated without noise. We obtained the results for 5 deformations after having done 50 trials for each one. From the Table 2, it can be seen that the result of our approach is comparable to that of the SOCP-based method. The reconstruction errors are considerably lower than those in table 1, which may imply that the use of good-quality real data for training might improve significantly the results.

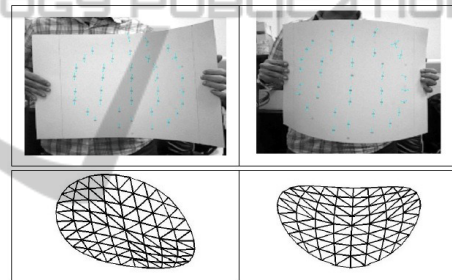


Figure 6: Real deformations; A 20×20 -cm square was selected from the intermediate part of the cardboard and the corresponding circle was reconstructed.

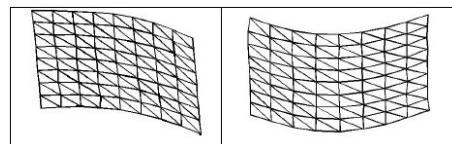


Figure 7: The reconstructed shape of the corresponding squares in Figure 6.

Table 2: Comparison between the proposed approach and the SOCP-based one.

Reconstruction error	PRE	MRE
Our approach	0.0120	0.0185
SOCP-based approach	0.0162	0.0217

6 CONCLUSIONS

In this paper, we have proposed a SfM framework combining a monocular camera and a ToF sensor to

reconstruct surfaces which deform isometrically. The ToF camera was used to provide us with the depth of a sparse set of feature points, from which we can recover the depth of the mesh using a multivariate linear system. The key advantage of the RGB/ToF system is to benefit from the high-resolution RGB data in combination with the low-resolution depth information. We proposed an approach to inextensible surface reconstruction, which is formulated as an optimization problem. Finally, we carried out a set of experiments showing that the approach generates good results in cases where 3D points are well-distributed. As next objective, we will extend the approach to deal with non-rigid surfaces which are not isometric e.g. conformational surfaces and etc.

REFERENCES

- Aans, H. and Kahl, F. (2002). Estimation of deformable structure and motion. *Workshop on Vision and Modelling of Dynamic Scenes, ECCV, Denmark*.
- Akhter, I., Sheikh, Y., and Khan, S. (2009). In defense of orthonormality constraints for nonrigid structure from motion. pages 1534–1541. CVPR.
- Dai, Y., Li, H., and He, M. (2012). A simple prior-free method for non-rigid structure-from-motion factorization. pages 2018–2025. CVPR.
- Del-Bue, A., Llad, X., and Agapito, L. (2006). Non-rigid metric shape and motion recovery from uncalibrated images using priors. IEEE Conference on Computer Vision and Pattern Recognition, New York.
- Diebel, J. and Thrun, S. (2005). An application of markov random fields to range sensing. Proc. NIPS.
- Gay-Bellile, V., Perriollat, M., Bartoli, A., and Sayd, P. (2006). Image registration by combining thin-plate splines with a 3d morphable model. International Conference on Image Processing.
- Gumerov, N., Zandifar, A., Duraiswami, R., and Davis, L. (2004). Structure of applicable surfaces from single views. European Conference on Computer Vision.
- Kim, H., Tai, Y.-W., and Brown, M. (2011). High quality depth map upsampling for 3d-tof cameras. pages 1623 – 1630. Inso Kweon Computer Vision (ICCV), IEEE International Conference, Barcelona.
- Kim, Y., Theobalt, C., Diebel, J., Kosecka, J., Miscusik, B., and Thrun, S. (2009). Multi-view image and tof sensor fusion for dense 3d reconstruction. pages 1542–1549. Computer Vision Workshops (ICCV Workshops).
- Llado, X., Bue, A., and Agapito, L. (2005). Non-rigid 3d factorization for projective reconstruction. BMVC.
- Metaxas, D. and Terzopoulos, D. (1993). Constrained deformable superquadrics and nonrigid motion tracking. *PAMI 15*, pages 580–591.
- Paladini, M., Bue, A., Stosic, M., Dodig, M., Xavier, J., and Agapito, L. (2009). Factorization for non-rigid and articulated structure using metric projections. page 28982905. Proc. IEEE Conf. on Computer Vision and Pattern Recognition.
- Perriollat, M., Hartley, R., and Bartoli (2010). Monocular template-based reconstruction of inextensible surfaces. *International Journal of Computer Vision*.
- Prasad, M., Zisserman, A., and Fitzgibbon, A. (2006). Single view reconstruction of curved surfaces. pages 1345–1354. IEEE Conference on Computer Vision and Pattern Recognition.
- Salzmann, M. and Fua, P. (2007). Reconstructing sharply folding surfaces: A convex formulation. IEEE Conference on Computer Vision and Pattern Recognition.
- Salzmann, M., Hartley, R., and Fua, P. (2007). Convex optimization for deformable surface 3-d tracking. IEEE International Conference on Computer Vision.
- Salzmann, M., Moreno-Noguer, F., Lepetit, V., and Fua, P. (2008). Closed-form solution to non-rigid 3d surface registration. pages 581–594. European Conference on Computer Vision.
- Shen, S., Shi, W., and Liu, Y. (2010). Monocular 3-d tracking of inextensible deformable surfaces under l2-norm. *IEEE Transactions on Image Processing 19*, pages 512–521.
- Srivastava, S., Saxena, A., Theobalt, C., and Thrun, S. (2009). Rapid interactive 3d reconstruction from a single image. In *VMV*, pages 19–28.
- Torresani, L., Hertzmann, A., and Bregler, C. (2003). Learning non-rigid 3d shape from 2d motion. *NIPS*, pages 580–591.
- White, R. and Forsyth, D. (2006). Combining cues: Shape from shading and texture. CVPR.
- Xiao, J., x. Chai, J., and Kanade, T. (2004). A closed-form solution to non-rigid shape and motion recovery. pages 573–587. ECCV.
- Yang, R., Davis, J., and Nister, D. (2007). Spatial-depth super resolution for range images. pages 1–8. Computer Vision and Pattern Recognition, CVPR '07. IEEE Conference, Minneapolis, MN.
- Zhou, H., Li, X., and Sadka, A. (2012). Nonrigid structure-from-motion from 2-d images using markov chain monte carlo. 14(1):168–177.