Fast and Robust Monocular 3D Deformable Shape Estimation for Inextensible and Smooth Surfaces.

Luis Ferraz, Xavier Binefa

Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain
{luis.ferraz, xavier.binefa}@upf.edu

Abstract

We present a method for recovering fast and robustly the 3D shape of inextensible and smooth surfaces from a monocular image. We propose a weighted iterative least squares approach to minimize the reprojection error between 2D-3D point correspondences preserving the 3D lengths. In addition, a local 3D smoothness constraint for each mesh vertex is proposed to increase the robustness to noisy correspondences and occluded or poorly represented facets. Moreover, the proposed method updates automatically the relevance of each constraint in order to maximize the smoothness and minimize the reprojection error. Experimental results shown that our approach obtains accurate results and is faster than state-of-the-art algorithms using similar constraints.

1. Introduction

Monocular 3D shape estimation of deformable surfaces has aroused renewal interest since the inextensibility property of surfaces, which constraints the lengths of the edges between neighboring surface points, have shown to be enough to recover their 3D shape [9][7][2][5]. 3D shapes can be recovered from single images providing enough 2D-3D correspondences between those single images and one in which the surface’s shape is already known. This known surface can be represented by sparse points [5] or by a triangulated 3D mesh [9][7].

The length of the edges between neighboring vertices can be used as rigid constraints [8][9], assuming that geodesic distances can be approximated by euclidean distances using equality constraints under $L_2^2$-norm. [8] minimizes a linearization for $L_2$ edge length constraints into the nullspace of the reprojection constraints and [9] minimizes the correspondence reprojection errors into the nullspace obtained from a linear approximation of $L_2$ edge length constraints. In terms of computational cost, the linearization method in [8] and the iterative computation of the nullspace and the column space in [9] become expensive.

For non-necessarily smooth surfaces, in [6][7] the length of the edges are used as relaxed constraints assuming that geodesic distances are always bigger or equal to the euclidean ones using inequality constraints, which are minimized using convex optimization [1]. This approach assumes that each facet may represent several surface folds.

Concretely, for smooth surfaces, [2] proposes to obtain an initial 3D shape using convex optimization in order to fit a 3D bicubic spline. After that, the resulting surface is refined by means of a Free Form Deformation model. However, although its results are accurate, it is computationally expensive.

The goal of the proposed method is to minimize fast and robustly the 3D euclidean error of the length of the edges between the estimated mesh and the known 3D surface, assuming noisy 2D-3D correspondences with outliers, partial occlusions and poorly represented regions. In this paper we evaluate the ways to use inextensibility property without learning and temporal consistency, although both can be added as in [6].

2. Constrained 3D shape estimation

In this section we introduce the problem statement and the proposed constraints for inextensible and smooth surfaces. As in state-of-the-art literature we assume a perspective camera with known intrinsic parameters and, for notation simplicity, all the point coordinates are expressed in the camera referential.

2.1 Problem statement

The surface to be estimated is represented as a triangulated 3D mesh with $n_v$ vertices $v_i = [x_i, y_i, z_i]^T$ and $n_e$ edges providing $L_2^2$ edge length constraints $L = [l_1...l_n_e]^T$, being the mesh parametrized by $V = [v_1^T...v_n_v^T]^T$. In order to estimate $V$, a set of $n_e$ 3D-to-2D correspondences is obtained between the known 3D
mesh and the input image. Correspondences relate each 3D point \(x_j\) expressed in terms of its barycentric coordinates \([a_j, b_j, c_j]\) in the facet \(f\) with a 2D feature point \(p_j = [u_j, v_j]^T\) in the input image.

Each 3D point \(x_j\) on the known surface can be expressed in barycentric coordinates as
\[
x_j = a_j v_{f1} + b_j v_{f2} + c_j v_{f3} = T_j V
\]
where \(v_{f1}, v_{f2}\) and \(v_{f3}\) are the three vertices of facet \(f\) and \(T_j\) is the transformation matrix for the barycentric coordinates of the correspondence \(j\), \(1 \leq j \leq n_c\).

Let the correspondence between \(x_j\) and \(p_j\) be expressed as,
\[
KT_j V = d \begin{bmatrix} u_j \\ v_j \\ 1 \end{bmatrix}
\]
where \(K\) denotes the intrinsic camera parameters and \(d\) is a scalar representing the depth.

2.2 Minimizing the edge length error using \(L_2\)-norm

Geodesic distances between 3D mesh vertices on smooth surfaces are similar to the Euclidean ones. Thus, \(V\) can be rigidly constrained as in [9] minimizing the quadratic \(L_2\) distance between pairs of vertices,
\[
l_r = ||v_j - v_k|| = ||E_r V||
\]
where \(v_j\) and \(v_k\) are the end vertices of the edge \(r\), being \(E_r\) the transformation matrix, \(1 \leq r \leq n_e\).

As in [9], Eq. (3) is locally approximated using an iterative approach. In the \(n\)-th iteration \(V_{n+1} = V_n + \Delta_n\) and \(\|E_r (V_n + \Delta_n)\| = l_r\), which can be expressed as,
\[
2V_n^T E_r^T E_r \Delta_n + \Delta_n^T E_r^T E_r \Delta_n = l_r^2 - V_n^T E_r^T E_r V_n
\]
and approximated by,
\[
2V_n^T E_r^T E_r \Delta_n \approx l_r^2 - V_n^T E_r^T E_r V_n
\]
The \(n_c\) equality constraints showed in (5) can be written as,
\[
F_n \Delta_n = g_n
\]
where \(F_n\) is a \(n_c \times 3n_v\) matrix and \(g_k\) is a \(n_c\) vector.

2.3 Minimizing iteratively the reprojection error between correspondence points

The reprojection error is the distance between \(p_j\) and the projection of \(x_j\) on the image plane. Rewriting Eq. (2) as in [8], two linear constraints denoting the reprojection error are obtained,
\[
\left( K_{2 \times 3} \begin{bmatrix} u_j \\ v_j \end{bmatrix} \right) T_j V = 0
\]
where \(K_{2 \times 3}\) and \(K_3\) are the first 2 rows and last row of \(K\), respectively. The \(n_c\) pairs of linear constraints can be concatenated and written as,
\[
MV = 0
\]
where \(M\) is a \(2n_c \times 3n_v\) matrix.

We need to minimize \(M\) jointly with the edge length constraints shown in Eq. (6), thus, the Eq. (8) must be minimized respect to \(\Delta_n\) and rewritten as,
\[
M(V_n + \Delta_n) = 0
\]
As \(M\) is a linear set of equations, the iterative approach to minimize the reprojection error can be expressed as,
\[
MV_n + M\Delta_n = 0
\]
where not all of them have a low level of noise and are correct. Therefore, to increase the robustness and reduce the effect of outliers, the reprojection error \(c_j\) provided by each correspondence is used, as in [6], to weight each constraint equation as,
\[
w_j = \exp\left(-\frac{c_j}{\text{median}(c_j)}\right)
\]
where \(1 \leq j \leq n_c\). However, Eq. (11) globally biases the weights, it does not take into account that concrete facets can be more noisy than others. Thus, in our formulation Eq. (11) is locally constrained as,
\[
w_j' = \sum_{f \in N_f} w_j \geq 1
\]
where \(N_f\) denotes the correspondences laying on the facet \(f\). Our local constraint ensures the rank of \(M\) do not decrease and ensures a minimal relevance for each facet \(f\).

2.4 Mesh regularization for occlusion and poorly represented regions

Occlusions and poorly represented regions are a common issue in 3D deformable surface estimation. We propose a low computational cost and robust constraint to regularize the 3D mesh estimation. Our approach assumes that neighboring vertices should suffer similar displacements, a similar idea was proposed in [4] for 2D tracking of deformable surfaces showing its performance.

For each mesh vertex \(v_i\), we obtain 3 weighted smoothing constraints,
\[
\frac{1}{w_i} \left( \Delta v_i - \frac{1}{\sum_{k \in N_k} w_k} \sum_{k \in N_k} w_k \Delta v_k \right) = 0
\]
where \(N_k\) denotes the neighborhood of \(v_i\) and \(\Delta v_i\) represent iterative increments for \(v_i\). The weights \(w_i\) and \(w_k\) denote the relevance of each mesh vertex depending on the reprojection error in their surrounding facets,
\[
w_k = \sum_{f \in N_f} w_j
\]
where \(N_f\) denotes the facets in the neighborhood of \(v_k\).

The Eq. (13) generates \(3n_v\) constraints, which can be written as,
\[
S \Delta_n = 0
\]
where \(S\) is a \(3n_v \times 3n_v\) matrix.

Being \(S\) a weighted matrix, which provides rigid constraints for those facets in the mesh occluded or poorly represented.
2.5 Recovering the 3D shape under $L_2$-norm

We recover the 3D shape using the constraints previously shown into an iterative least squares approach. In our approach the function to be minimized is controlled by the edge length constraints, Eq. (6), allowing to the reprojection error constraints, shown in Eq. (10), bias the minimization while Eq. (15) maintains the full rank when reprojection errors increase. Our function to be iteratively minimized using the Gauss-Newton method can be written as,

$$\min_{\Delta_n} \| F_n \Delta_n - g_n \| + \alpha \| MV_n + M \Delta_n \| + \beta \| S \Delta_n \|$$

(16)

where $\alpha$ and $\beta$ are regularization parameters for the reprojection error term and the smoothing term.

From Eq. (16), $V$ is iteratively estimated as $V_{n+1} = V_n + \Delta_n$ until $\max_i(\Delta_n^2) < \epsilon$ or a maximum number of iterations. We adapt $\epsilon$ in function of the edge lengths in $L$ as $\epsilon = 0.1 \max(L)$, obtaining a consistent threshold with the mesh distances.

The regularization parameters in Eq. (16), are iteratively updated depending on the correctness in the correspondences,

$$\alpha = 0.001 \frac{n_n}{\sum w_j} \quad \beta = \frac{1}{\sum w_j}$$

(17)

Thus, when reprojection errors increase, the reprojection term takes relevance to reduce its error in next iterations and, simultaneously, the smooth term takes relevance in order to prevent non-smooth surface deformations. Moreover, both term equations are also internally weighted by Eq. (11) constrained with Eq. (12) and Eq. (14) in order to take into account local errors.

Finally, our algorithm (shown in Algorithm 1) uses an iterative procedure, as in [6], in order to detect outliers. However, our approach computes the maximum radius for each iteration taking into account the $e_j$ values, preventing that all the correspondences are considered as outliers. We compute the radius as the maximum value between an empirically fixed value and 2 median($e_j$).

3. Experimental Results

In this section the performance of our algorithm versus algorithms shown in [9] and [6] is evaluated in terms of tracking in two main aspects, their computational cost and their accuracy recovering smooth 3D shapes. The evaluation has been done using standard datasets containing synthetic and real image sequences. The algorithm by Shen et al. [9] was implemented by us in Matlab using $\epsilon = 0.1 \max(L)$ as finishing criteria and discarding outliers every 10 iterations, being discarded if $e_j > 3 \text{median}(e_j)$. In contrast the algorithm by Salzmann et al. [6] was obtained from

Algorithm 1 Robust 3D Smooth Shape Recovering

for each input image do
radius $\leftarrow$ 50
Compute $M$ using Eq. (7)
for $i = 1 \rightarrow 10$ do
Compute the weighted $M$ using Eq. (11) (12)
Compute $S$ using Eq. (13)
Compute $\alpha$ and $\beta$ using Eq. (17)
for $n = 1 \rightarrow 10$ do
Compute $F_n$ and $g_n$ using Eq. (5)
Estimate $\Delta_n$ using Eq. (16)
$V_n \leftarrow V_n + \Delta_n$
if $\max_i(\Delta_n^2) < 0.1 \max(L)$ then
break
end if
end for
Discard correspondences with $e_j > \text{radius}$
radius $\leftarrow \max(0.7 \text{radius}, 2 \text{median}(e_j))$
end for

Figure 1: Performance comparison for the cardboard dataset. Top: in terms of computational speed (sc). Bottom: in terms of mesh 3D error (mm).

http://cvlab.epfl.ch/research/surface/deformable/ Our algorithm and [9] were completely developed using Matlab, while [6] minimize its objective function using the SeDuMi package. The computational speed evaluations have been performed in a 2.5GHz Core2Duo Laptop.

Experiment I We have applied the 3 algorithms on a subset of 150 frames taken from the cardboard dataset, which has been generated using an optical motion capture system. For each facet and pair of frames 10 perfect correspondences have been randomly created (1280 correspondences in the mesh), which have been perturbed adding zero mean gaussian noise of variance 5. Fig. (1) shows that our method is several times faster than the other methods providing the best accuracy.

Experiment II We now show the reconstruction results on two real image datasets. In both sequences, 10 random samples for each facet in the first frame have been created and tracked using standard cross correla-

1http://cvlab.epfl.ch/data/dsr/
2http://www.brnt.eu/accv2010datasets.php
tion. This yields to noisy correspondences with outliers for the following frames. In Fig. (2), first and second columns show the reconstruction results for the last frames in both sequences, which start with an undeformed mesh. The third row in both columns show the comparison in terms of computational speed with the other two methods, showing that our method is also several times faster than the others on real image sequences. However, using these correspondences the method of Shen et al. has failed in the second sequence.

Experiment III This experiment shows the robustness of our algorithm under self-occlusions reconstructing the self-occluded bending paper dataset. The correspondences in this dataset has been extracted using [3]. In Fig. (2), the third column show the reconstructed mesh under a severe self-occlusion.

4. Conclusions and future work

In this paper a fast and robust method to estimate 3D deformable shapes from monocular image sequences of smooth and inextensible surfaces is proposed. Our method iteratively reconstructs 3D meshes constrained by the edge lengths using rigid L2-norm constraints, the reprojection error of their correspondences and a local smoothing term for each vertex. Moreover, our method updates automatically the relevance of each constraint in order to maximize the smoothness and minimize the reprojection error. Experimental results show that our method accurately reconstructs 3D meshes outperforming clearly the other methods in terms of computational speed.

As future work, to reduce the ambiguities in complex sequences we should be adding temporal consistency and previously learned local deformable models to our minimization function.

Acknowledgments: This work was partially funded by CONSUMEDIA research project (IPT-2011-1015-430000).

References