Camera Tracking for SLAM in Deformable Maps

Jose Lamarca[0000−0002−8824−9390] and J.M.M. Montiel[0000−0002−3627−7306]

Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, 50018, Spain {jlamarca,josemari}@unizar.es

Abstract. The current SLAM algorithms cannot work without assuming rigidity. We propose the first real-time tracking thread for monocular VSLAM systems that manages deformable scenes. It is based on top of the Shape-from-Template (SfT) methods to code the scene deformation model. Our proposal is a sequential method that manages efficiently large templates, i.e. deformable maps estimating at the same time the camera pose and deformation. It also can be relocated in case of tracking loss. We have created a new dataset to evaluate our system. Our results show the robustness of the method in deformable environments while running in real time with errors under 3% in depth estimation.

Keywords: Non Rigid Structure-from-Motion, Shape-from-Template, Deformation Models, Deformable SLAM, Non-Rigid SLAM

1 Introduction

Recovering 3D scenes from monocular RGB-only images is a significantly challenging problem in Computer Vision. Under the rigidity assumption, Structure-from-Motion (SfM) methods provide the theoretical basis for the solution in static environments. Nonetheless, this assumption renders invalid for deforming scenes as most medical imaging scenarios. In the case of the non-rigid scenes the theoretical foundations are not yet well defined.

We can distinguish two types of algorithms that manage non rigid 3D reconstruction: Non-Rigid Structure-from-Motion (NRSfM), which are mostly batch processes, and Shape-from-Template (SfT), which work frame-to-frame. The main difference between these methods is that NRSfM learns the deformation model from the observations while SfT assumes a previously defined deformation model to estimate the deformation for each image.

Rigid methods like Visual SLAM (Simultaneous Localisation and Mapping) have made headway to work sequentially with scenes bigger than the camera field of view [16, 7, 13, 8]. Meanwhile, non-rigid methods are mostly focused on reconstructing structures which are entirely imaged and tracked, for example, surfaces, [6, 17, 24], faces [4, 25, 2, 20], or articulated objects [23, 15].

We conceive the first real-time tracking thread integrated in a SLAM system that can locate the camera and estimate the deformation of the surface based on top of a SfT algorithm following [24, 17, 21, 3]. Our method includes automatic data association and PnP+RANSAC relocalisation algorithm. We code the deformable map as a template which consists of a mesh with a deformation model. Our template is represented as a 3D surface triangular mesh with spatial and temporal regularisers that are rotation and translation invariant. We have selected it because it is suitable for implementing physical models and with barycentric coordinates we can relate the observations with the template.

We evaluate our algorithm with experimental validation over real data both for camera location and scene deformation. This is the first work that focuses on recovering the deformable 3D just from partial images. Thus, we have created a new dataset to experiment with partially-imaged template for sake of future comparison.

2 Problem formulation

2.1 Template definition

We code the deformable structure of the scene as a known template $\mathcal{T} \subset \mathbb{R}^3$. The template is modelled as a surface mesh composed of planar triangular facets $\mathcal{F}$ that connect a set of nodes $\mathcal{V}$. The facet f is defined in the frame i by its three nodes $V^i_{f_j} = \{V^i_{f,h}\} \;\; h = 1 \ldots 3$. The mesh is measured through observable points $\mathcal{X}$ which lie inside the facets. To code a point $X_j \in \mathcal{X}$ in frame i wrt. its facet f_j nodes, we use a piecewise linear interpolation through the barycentric coordinates $\mathbf{b}_j = [b_{j,1}, b_{j,2}, b_{j,3}]^\top$ by means of the function $\varphi : [\mathbb{R}^3, \mathbb{R}^{3x3}] \rightarrow \mathbb{R}^3$:

$$\mathbf{X}^i_j = \varphi(\mathbf{b}_j, \mathbf{V}^i_{f_j}) = \sum_{h=1}^{3} b_{j,h} \mathbf{V}^i_{f_j,h} \tag{1}$$

The camera is assumed projective, the observable point $\mathbf{X}^i_j \in \mathcal{T}$ defined in $\mathbb{R}^3$ is viewed in the frame i with the camera located in the pose $\mathbf{T}_i$ through the projective function $\pi : [\mathrm{Sim}\left(3\right), \mathbb{R}^3] \rightarrow \mathbb{R}^2$.

$$\pi\left(\mathbf{T}_i, \mathbf{X}^i_j\right) = \begin{bmatrix} f_u \frac{x^i_j}{z^i_j} + c_u \\ f_v \frac{y^i_j}{z^i_j} + c_v \end{bmatrix} \tag{2}$$

$$\left[x^i_j \; y^i_j \; z^i_j \right]^T = \mathbf{R}^i \mathbf{X}^i_j + \mathbf{t}^i \tag{3}$$

Where $\mathbf{R}^i \in SO(3)$ and $\mathbf{t}^i \in \mathbb{R}^3$ are respectively the rotation and the translation of the transformation $\mathbf{T}_i$ and $\{f_u, f_v, c_u, c_v\}$ are the focal lengths and the principal point that define the projective calibration for the camera. The algorithm works under the assumption of previously knowing the template. This is a common assumption of template methods. We efectivelly compute it by means of a rigid VSLAM algorithm [16]. We initialise the template from a 3D reconstruction of the shape surface at rest. We use Poisson surface reconstruction as it is proposed in [12] to construct the template triangular mesh from the sparse point cloud. Once the template is generated, only cloud points which lie close to a facet are retained and then projected into the mesh facets where their barycentric coordinates are computed.

3 Optimisation

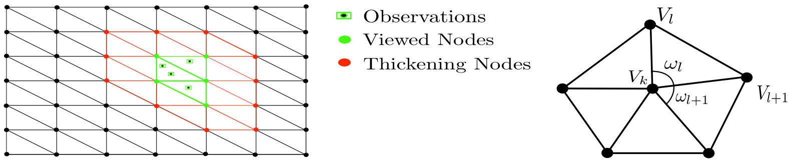

Fig. 1. Left: Two step region definition for the case of three observations inside two unconnected facets. $d_{\mathcal{K}} = 1$ for the thickening $\mathcal{K}_i$. Right: Ring of neighbours $\mathcal{N}_k$ of the node K.

We recover the camera pose and the deformation only in the template region detected by the camera. We define the *observation region*, $\mathcal{O}_i$, as the template nodes belonging to a facet with one or more matched observations in the current image i. We dilate the $\mathcal{O}_i$ region with a layer that we call *thickening layer*, $\mathcal{K}_i$ whose thickness is $d_{\mathcal{K}}$. We call the template region estimated in the local step *local map*, $\mathcal{L}_i$. It is defined as $\mathcal{L}_i = \mathcal{O}_i \cup \mathcal{K}_i$ (Fig. 1).

We propose the next optimisation to recover both the camera pose T_i and the position of the local map nodes $V_k^i \in \mathcal{L}_i$, in frame i:

$$
\begin{aligned}
\underset{T_i, V_k^i \in \mathcal{L}_i}{\arg\min}\ & \frac{1}{N_\bullet} \sum_j \rho \left(\left\| \pi_i \left(T_i, \varphi(\mathbf{b}_j, V_{f_j}^i) \right) - x_j^i \right\|^2 \right) \\
& + \frac{\lambda_d}{N_\bullet} \sum_k \sum_{l \in \mathcal{N}_k} \left(\frac{\left\| V_k^i - V_l^i \right\| - \left\| V_k^0 - V_l^0 \right\|}{\left\| V_k^0 - V_l^0 \right\|} \right)^2 \\
& + \frac{\lambda_L}{N_\bullet} \sum_k \left(\| \delta_k^i \| - \| \delta_k^0 \| \right)^2 \sum_{l \in \mathcal{N}_k} \frac{1}{\left\| V_k - V_l \right\|^2} \\
& + \frac{\lambda_T}{S N_\bullet} \sum_k \left\| V_k^i - V_k^{i-1} \right\|
\end{aligned}
\tag{4}
$$

The weights of the regularisers $\lambda_L, \lambda_d, \lambda_t$ are defined with respect to a unit weight for the data term. Additionally, we consider different normalisation factors to correct the final weight assigned to each term. We consider a correction depending on the number of addends, denoted as $N_\bullet$, in the summation of the corresponding regularising term and a scale correction for the temporal term.

The nodes not included in the optimisation, whose position is fixed, $V_k^i \in \{\mathcal{T} \setminus \mathcal{L}_i\}$, are linked with those optimised, hence they are acting as boundary conditions. As a consequence most of the rigid motion between the camera and the template is included in the camera motion estimate $\mathbf{T}_i$.

The regularisers code our deformation model, they are inspired in continuum mechanics where bodies deform generating internal energies due to normal strain and shear strain. The first term is the Cauchy or engineering strain:

$$\sum_{k} \sum_{l \in \mathcal{N}_k} \left(\frac{\left\| V_k^i - V_l^i \right\| - \left\| V_k^0 - V_l^0 \right\|}{\left\| V_k^0 - V_l^0 \right\|} \right)^2 \tag{5}$$

It penalises the normal strain energy. Per each node V_k^i we consider a summation over the ring of its neighbours N_k. Per each neighbour the deformation energy is computed as proportional to the squared ratio between the distance increment and the distance at rest. Unlike other isometry or inextensibility regularisers, [17, 10], it is a dimensionless magnitude, invariant with respect to the facet size. Per each node V_k^i we consider its ring of neighbours $\mathcal{N}_k$ in the computation.

The second regulariser is the bending energy:

$$\sum_{k} \left(\|\delta_k^i\| - \|\delta_k^0\| \right)^2 \sum_{l \in \mathcal{N}_k} \frac{1}{\left\| V_k - V_l \right\|^2} \tag{6}$$

It penalises the shear strain energy. It is coded as the squared ratio between the deflection change and the mean edge length in its ring of neighbours $\mathcal{N}_k$. We use the ratio in order to get dimensionless magnitude invariant to the facet size. The deflection δ_k^i also represents the mean curvature, it is computed by means of the discrete Laplace-Beltrami operator:

$$\delta_k^i = V_k^i - \frac{1}{\sum_{l \in \mathcal{N}_j} \omega_l} \sum_{l \in \mathcal{N}_j} \omega_l V_l^i \tag{7}$$

in order to cope with irregular and obtuse meshes, ω_l is defined by the so-called mean-value coordinates [9]:

$$\omega_l = \frac{\tan(\Omega_{k,l}^1/2) + \tan(\Omega_{k,l}^2/2)}{\left\| V_k^0 - V_l^0 \right\|} \tag{8}$$

The $\Omega_{k,l}^1$ and $\Omega_{k,l}^2$ angles are defined in Figure 1.

The last term codes a temporal smoothing between the nodes in $\mathcal{L}_i$. This term is dimensionless with the term S. This term is the average length of the arcs in the mesh. We optimise with the Levenberg–Marquardt algorithm implemented in the library g2o [14].

4 SLAM Pipeline

To compose the entire tracking thread, we integrate the optimisation in a pipeline with automatic data association working with ORB points, and a DBoW keyframe database [11] that allows relocalisation in case of losing the tracking.

Our optimisation method uses as input the observations of the template points in the current frame. Specifically, multiscale FAST corner to detect the observations, and the ORB descriptor [22] to identify the matches. We apply the classical in VSLAM *active matching*, that sequentially process the image stream. First, the ORB points are detected in the current image. Next, with a camera motion model, it is predicted the camera pose as a function of the past camera poses. Then the last template estimate and the barycentric coordinates, are used to predict where the template points would be imaged. Around the template point prediction it is defined a search region. Among the ORB points inside the search region, the one with the closest ORB descriptor is selected as the observation. We apply a threshold on the ORB similarity to definitively accept a match. The ORB descriptor of the template point is taken from the template initialisation. The similarity is estimated as the Hamming distance between the ORB descriptors. To reduce the number of false negatives, we cluster the matches according to their geometrical innovation, difference between the predicted template point in the image and the detected one. Only the three main clusters of matches are retained.

As an approach of relocalisation algorithm, we use a relaxed rigid PnP + RANSAC algorithm. We test the original rigid PnP in five thousand images that contain deformation and we got a recall of 26% successful relocalisation, with the relaxed method up to a 49%. The precision in the relocalisation is close to the 100%.

5 Experiments

Comparison with state of the art SfT. We benchmark our proposal with the standard *Kinect paper dataset*, to compare the performance of our deformation model with respect to state-of-the-art template-based algorithms. Kinect paper dataset is composed of 193 frames, each frame contains around 1300 observations coming form SIFT points. The matches for the observations are also provided. The ground truth for the matched points are computed from a Kinect RGB-D sensor. The benchmark considers a template that can be fit within the camera field of view. To make an homogeneous comparison we fixed the camera and leave the boundaries of the mesh free. In table 1 we show the mean RMS error along the sequence compared with respect to some popular methods [6, 1, 24, 19, 18], results are taken from [19]. Ours gets 4.86 mm at 13 ms per frame, what is comparable with the similar state-of-the-art algorithms [24, 18].

Experimental validation. To analyse the performance of our system, we have created the *mandala dataset*. In this dataset, a mandala blanket is hanged and deformed manually from its back surface, meanwhile a hand-held RGB-D camera closely observes the mandala surface mimicking a scanning movement in circles. Due to the limited field of view of the camera and its proximity to the cloth, the

Table 1. RMSE averaged over all the frames in the sequence.

	[6]	[1]	[19]	[24]	[18]	[5]	Ours
Mean RMSE (mm)	3.97	4.56	3.78	7.47	4.82	3.86	4.86
Runtime per Frame (ms)	2	0.7	7	5	30	116	13

whole mandala is never completely imaged. We run the experiments in a Intel Core i7-7700K CPU @ 4.20GHz 8 with a 32GB of RAM memory.

The sequence is composed of ten thousand frames, there is a first part for initialisation where the cloth remains rigid. After that, the level of hardness of the deformation is progressively increased. The video captures from big displacements in different points of the mandala to wrinkled moments and occlusions.

We evaluate the influence of the thickening layer size, $d_{\mathcal{K}}$. As result of the experiment, we get a system that can run in real-time and have an RMS error of 2.30%, 2.22%, and 2.32% for $d_{\mathcal{K}} = 0, 1$ and 2 respectively. When it comes to runtime, the optimisation algorithm is taking 17, 19 and 20 ms, and the total times per frame are 39, 40 and 41 ms. With $d_{\mathcal{K}=1}$ we get to reduce the error without increasing excessively the time.

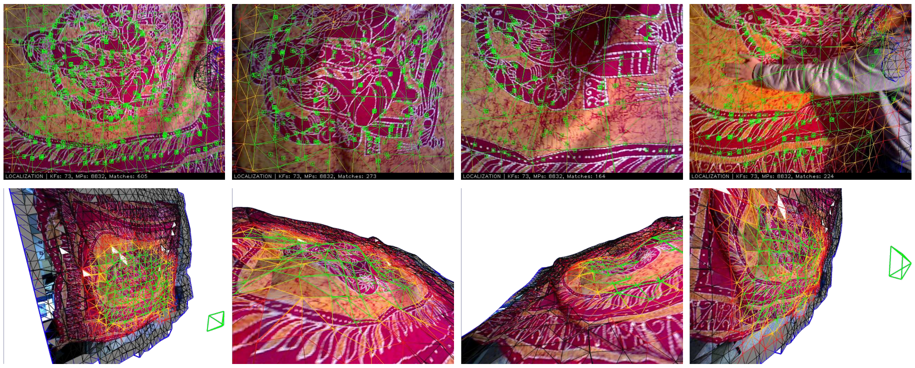
LOCALIZATION | KFs: 73, MPs: 8832, Matches: 605
LOCALIZATION | KFs: 73, MPs: 8832, Matches: 273
LOCALIZATION | KFs: 73, MPs: 8832, Matches: 164
LOCALIZATION | KFs: 73, MPs: 8832, Matches: 224

Fig. 2. From left to right: frames #1347, #2089, #9454, #10739, corresponding to the shape at rest and different deformations. Top: 2D image Bottom: 3D reconstruction

6 Discussion

We present a new tracking method able to work in deformable environment incorporating SfT techniques to a SLAM pipeline. We have developed a full-fledged SLAM tracking thread that can robustly operate with an average time budged of 39 ms per frame in very general scenarios with an error under 3% in a real scene and with a relocalisation algorithm with a recall of a 46% in deformable environments with a precision close to the 100%.

Acknowledgements. The authors are with the I3A, Universidad de Zaragoza, Spain. This research was funded by the Spanish government with the projects DPI2015-67275-P and the FPI grant BES-2016-078678.

References

1. Bartoli, A., Grard, Y., Chadebecq, F., Collins, T.: On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2026–2033 (June 2012). https://doi.org/10.1109/CVPR.2012.6247906
2. Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., Sayd, P.: Coarse-to-fine low-rank structure-from-motion. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
3. Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T., Pizarro, D.: Shape-from-template. IEEE transactions on pattern analysis and machine intelligence **37**(10), 2099–2118 (2015)
4. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. vol. 2, pp. 690–696. IEEE (2000)
5. Brunet, F., Hartley, R., Bartoli, A., Navab, N., Malgouyres, R.: Monocular template-based reconstruction of smooth and inextensible surfaces. In: Asian Conference on Computer Vision. pp. 52–66. Springer (2010)
6. Chhatkuli, A., Pizarro, D., Bartoli, A.: Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity. In: BMVC (2014)
7. Concha, A., Civera, J.: DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 5686–5693. IEEE (2015)
8. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
9. Floater, M.S.: Mean value coordinates. Computer Aided Geometric Design **20**(1), 19–27 (2003)
10. Gallardo, M., Collins, T., Bartoli, A.: Can we jointly register and reconstruct creased surfaces by shape-from-template accurately? In: European Conference on Computer Vision. pp. 105–120. Springer (2016)
11. Gálvez-López, D., Tardós, J.D.: Bags of binary words for fast place recognition in image sequences. IEEE Transactions on Robotics **28**(5), 1188–1197 (October 2012). https://doi.org/10.1109/TRO.2012.2197158
12. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. pp. 61–70. Eurographics Association (2006)
13. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. pp. 225–234. IEEE (2007)
14. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g2o: A general framework for graph optimization. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. pp. 3607–3613. IEEE (2011)
15. Lee, M., Choi, C.H., Oh, S.: A procrustean markov process for non-rigid structure recovery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1550–1557 (2014)
16. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics **31**(5), 1147–1163 (2015)

17. Ngo, D.T., Östlund, J., Fua, P.: Template-based monocular 3d shape recovery using laplacian meshes. IEEE transactions on pattern analysis and machine intelligence **38**(1), 172–187 (2016)
18. Östlund, J., Varol, A., Ngo, D.T., Fua, P.: Laplacian meshes for monocular 3d shape recovery. In: European conference on computer vision. pp. 412–425. Springer (2012)
19. Özgür, E., Bartoli, A.: Particle-sft: A provably-convergent, fast shape-from-template algorithm. International Journal of Computer Vision **123**(2), 184–205 (2017)
20. Paladini, M., Bartoli, A., Agapito, L.: Sequential non-rigid structure-from-motion with the 3d-implicit low-rank shape model. In: European Conference on computer vision. pp. 15–28. Springer (2010)
21. Perriollat, M., Hartley, R., Bartoli, A.: Monocular template-based reconstruction of inextensible surfaces. International journal of computer vision **95**(2), 124–137 (2011)
22. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE international conference on. pp. 2564–2571. IEEE (2011)
23. Russell, C., Fayad, J., Agapito, L.: Energy based multiple model fitting for non-rigid structure from motion. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 3009–3016. IEEE (2011)
24. Salzmann, M., Fua, P.: Linear local models for monocular reconstruction of deformable surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(5), 931–944 (2011)
25. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(5), 878–892 (2008)