Simultaneous Reconstruction and Tracking of non-planar Templates

Sebastian Lieberknecht[1], Selim Benhimane[1] and Slobodan Ilic[2]

[1]metaio GmbH, Munich, Germany
[2]Technische Universtität München, Garching, Germany

Abstract. In this paper, we address the problem of simultaneous tracking and reconstruction of non-planar templates in real-time. Classical approaches to template tracking assume planarity and do not attempt to recover the shape of an object. Structure from motion approaches use feature points to recover camera pose and reconstruct the scene from those features, but do not produce dense 3D surface models. Finally, deformable surface tracking approaches assume a static camera and impose strong deformation priors to recover dense 3D shapes.
The proposed method simultaneously recovers the camera motion and deforms the template such that an approximation of the underlying 3D structure is recovered. Spatial smoothing is not explicitly imposed, thus templates of smooth and non-smooth objects can be equally handled. The problem is formalized as an energy minimization based on image intensity differences. Quantitative and qualitative evaluation on both real and synthetic data is presented, we compare the proposed approach to related methods and demonstrate that the recovered camera pose is close to the ground truth even in presence of strong blur and low texture.

1 Introduction

Template tracking is one of the fundamental problems in computer vision and a multitude of impressive techniques have been proposed in the literature [12, 1, 4, 9]. They mainly concentrate on planar templates and estimate camera motion by energy minimization. The applications of template tracking are wide and include, but are not limited to, vision-based control, human-computer interfaces, augmented reality, robotics, surveillance, medical imaging and visual reconstruction. In many applications, the planarity assumption is good enough, but in general that is not the case. For that reason Silveira and Malis [17] and Bartoli and Zisserman [2] considered computing 2D warpings of the reference templates while tracking them. The real depth and camera motion are then obtained by decomposing the estimated warpings.

Motivated by the fact that the world is not planar and driven by the emerging needs of simultaneous recovery of the structures and motion of the camera, we address the problem of simultaneous tracking and reconstruction of a non-planar template in real time. The model of the template is represented as a triangular mesh. We start with a planar shape and simultaneously recover camera motion

and deform the shape such that the underlying 3D structure is approximately recovered. As we use all pixels of the template, the object does not necessarily have to be well textured and contain many feature points. This is different from classical Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM) techniques that primarily rely on sparse feature points, such as *e.g.* Klein and Murray's PTAM [10] of which Newcombe and Davidson [13] use the camera poses and sparse feature map to create a dense reconstruction. They perform very well, but depend on the amount of the observed features and tend to be sensitive to the amount of blur.

Unlike methods which rely on prior deformation models [16, 15] and assume fixed camera position, we solve for camera motion and do not impose any constraints on the model deformation, therefore we can equally reconstruct and track templates that are smooth or have creases. However, since the problem is ill-posed, we have made certain assumptions: we use templates of a predefined size, assume that in its initial/reference position the entire template is visible and is not self-occluded, and finally we restrict mesh vertices to only move along the camera rays, thus having one degree of freedom per vertex.

We evaluated the performance of our method on both synthetic and real video sequences. Further, we performed quantitative analysis and compared the method to ground truth measurements and to standard planar template tracking methods and PTAM. Our experiments indicate that, even with the approximate shape we recover, the tracking precision increased and turned out to be much more stable than tracking of planar templates and deals better with blur and low-textured surfaces than PTAM.

In the remainder of the paper, we first discuss related works, then describe our method in detail, finally present experimental results and conclude.

2 Related Work

Template tracking has always been assuming the planarity of the object of interest to be tracked. Since Lucas-Kanade [12], the real-time constraint was enforced and in recent works [1, 3] it became standard. Improvements in convergence speed and robustness in the calibrated camera setting were especially achieved by the method of Benhimane and Malis [4]. For those reasons, we in part relied on their method.

Other researchers [14, 7, 17] also proposed to find deformations of an object in a sequence of acquired images. These methods generally consist of estimation of the parameters of the warping function that registers the reference image, in which the object is mainly planar, to the input image where the object is deformed. Pilet *et al.* [14] and Gay-Bellile *et al.* [7] relied on feature points. While the former can deal with a huge amount of outliers, the latter is relatively sensitive to them. Datta *et al.* [5] use affine warps and integrated the idea of articulated points as hard constraints into the minimization, *i.e.* they force patches to move according to their connectivity. Hilsmann *et al.* [8] re-texture the surface of a deforming object realistically by estimating both the changes

in geometry and photometry, they also explicitly model external occlusions to further improve the quality of the augmentation. Silveira and Malis [17] use 2D warps and present a generic framework for template tracking which can undergo deformations. In all of these cases, the warping is done in image space and therefore does not provide a 3D shape, but instead 2D warpings of the images as in deformable registration. To recover the 3D shape, the recovered 2D warpings are decomposed into a rigid motion and according depths.

On a separate track, deformable surface tracking from monocular videos has been developed. Because of the inherent ambiguity, deformation models have been introduced to constrain deformations of particular objects like *e.g.* paper and clothes [16, 15, 19]. These approaches generally output the 3D surface meshes. However, they do not provide the relative camera/object motion in the image sequence, require heavily textured objects and generally do not work in real-time.

Simultaneous recovery of the camera motion and the 3D shape is also related to SfM [18] and SLAM [6] techniques. Both techniques strictly rely on image features and incremental reconstruction of an observed scene, while neither of them operates on the dense pixel level. The system proposed by Newcombe and Davidson [13] indeed produces a dense reconstruction using a movable camera; it relies on PTAM [10] to precisely recover the motion of the camera, they also use its sparse feature map to initialize a dense optical flow method [20].

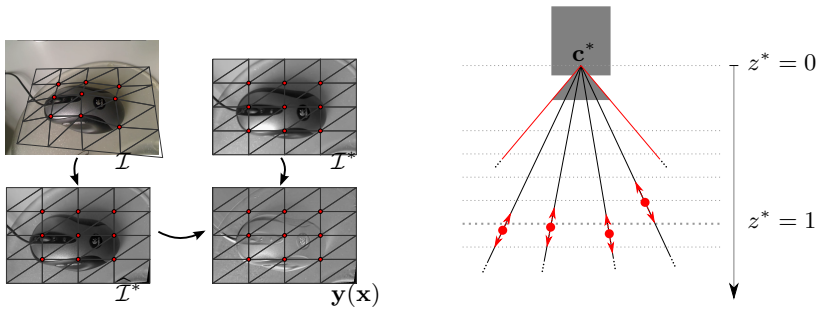
Most of the previously mentioned methods are using feature points and/or define constraints on the possible model deformations. Relying on features usually implies that the observed object has to be well textured. Instead of using a set of extracted feature points in the image, we use all available pixels of the template which in turn enables tracking of low textured templates. We simultaneously recover the camera motion and the approximate shape of the non-planar template. Our method exhibits fast convergence, is robust under blur, works in real-time and recovers quite precise camera pose given the on-line reconstruction of the approximate template's shape.

3 Method

The task of the algorithm is to estimate updates of the mesh M and the camera pose $\mathbf{T}$ given a novel image $\mathcal{I}$ of the object and relying of estimates on mesh and pose, denoted as $\widehat{M}$ and $\widehat{\mathbf{T}}$ obtained in the previous frame. We assume that, ignoring occlusion and drastic lighting changes, the reference image $\mathcal{I}^*$ can be constructed from $\mathcal{I}$ by back-warping each face f given the true pose and the recovered mesh. Given that we only know their approximations $\widehat{\mathbf{T}}$ and $\widehat{M}$, we produce an estimated image $\widehat{\mathcal{I}}^*$ by applying a homography $\mathbf{G}$ to each face of the mesh. This is illustrated in Figure 1(a). As the mesh is defined piece-wise planar, warping a single face f is conducted by the homography:

$$\mathbf{G}(\mathbf{T}, \mathbf{n}_f^*) = \mathbf{K}(\mathbf{R} + \mathbf{t}\mathbf{n}_f^{*\top})\mathbf{K}^{-1}\mathbf{G}_f. \tag{1}$$

Here, $\mathbf{K}$ denotes the known 3×3 camera intrinsics, $\mathbf{n}_f^* \in \mathbb{R}^3$ is the normal of face f scaled by the inverse of the distance d_f^* of the face to the camera center $\mathbf{c}^*$



(a) Overview of the method (b) Parameterization of z_i^*

Fig. 1. (a) The mesh is overlaid onto the object, with highlighted movable vertices. Out of the camera image $\mathcal{I}$ the estimate of the reference image $\widehat{\mathcal{I}}^*$ is unwarped. The error $\mathbf{y}(\mathbf{x}) = \widehat{\mathcal{I}}^* - \mathcal{I}^*$ is subject to iterative minimization. (b) The vertices of the mesh are free to move along their respective projection ray, *i.e.* (u_i^*, v_i^*) are fixed but z_i^* may change.

in the reference frame; the camera pose $\mathbf{T}$ is decomposed to get $\mathbf{R} \in \mathbb{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$. Finally, the homography $\mathbf{G}_f$ is used to apply a 2D translation of the face to its specified position within $\mathcal{I}^*$. We assume that the updates $\mathbf{T}(\mathbf{x}), \mathbf{n}_f^*(\mathbf{x})$ of the estimates $\widehat{\mathbf{T}}, \widehat{\mathbf{n}}_f^*$ are reasonably small. They are parameterized in terms of the camera pose and the mesh deformation $\mathbf{x} = (\omega_x, \omega_y, \omega_z, \nu_x, \nu_y, \nu_z, \psi_1, \psi_2, \ldots, \psi_n)$ where the first six parameters represent the update of the pose $\widehat{\mathbf{T}}$ of the camera, represented by the Lie algebra of $\mathbb{SE}(3)$. The remainder of $\mathbf{x}$ represents the update of the inverse depths $\psi_i = 1/z_i^*$ of the movable vertices.

Deformations of the mesh M^* are modeled by moving vertices along their respective rays emanating from the camera center $\mathbf{c}^*$ in the reference view, see Figure 1(b). Every vertex $\mathbf{v}_i^*$ is defined via its 2D coordinates $\mathbf{v}_i^* = (u_i^*, v_i^*, 1)^\top$ in $\mathcal{I}^*$ and its depth z_i^* w.r.t. the camera center $\mathbf{c}^*$. The normal $\mathbf{n}_f^*$ of a face f is computed from its vertices $\{\mathbf{v}_i^*, \mathbf{v}_j^*, \mathbf{v}_k^*\}$ and inverse depths:

$$\mathbf{n}_f^*(\mathbf{x}) = \frac{\mathbf{n}^*}{d^*} = \mathbf{K}^\top \left[\mathbf{v}_i^* \, \mathbf{v}_j^* \, \mathbf{v}_k^*\right]^{-\top} \left[\psi_i \, \psi_j \, \psi_k\right]^\top . \tag{2}$$

This formula was developed by combining the inverted pinhole projection $\mathbf{a} = (x, y, z)^\top = z\mathbf{K}^{-1}(u, v, 1)^\top$ with the plane equation $\mathbf{n}^\top \mathbf{a} = d$. Note that this parameterization of $\mathbf{n}_f^*(\mathbf{x})$ is linear w.r.t. the inverse of the depths.

For the sake of simplicity, we consider only a single face consisting of m pixels and define the $m \times 1$ error vector $\mathbf{y}(\mathbf{x})$ as concatenation of the error measures

$$y_i(\mathbf{x}) = \widehat{\mathcal{I}}^* - \mathcal{I}^* = \mathcal{I}\left(\mathbf{q}_i\right) - \mathcal{I}^*(\mathbf{p}_i^*) \tag{3}$$

$$= \mathcal{I}\left(\mathbf{d}\left(\mathbf{G}\left(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), \mathbf{n}_f^*(\widehat{\mathbf{x}} + \mathbf{x})\right) \mathbf{p}_i^*\right)\right) - \mathcal{I}^*(\mathbf{p}_i^*) \tag{4}$$

where $\mathbf{q}_i$ are pixel coordinates in the input image obtained by back-warping to the reference image and $\mathbf{d}((u, v, w)^\top) = (u/w, v/w, 1)^\top$ represent normalized

homogeneous coordinates. The current estimates of the depths is stored in $\widehat{\mathbf{x}}$, thus the update $\mathbf{n}_f^*(\widehat{\mathbf{x}} + \mathbf{x})$ used in Equation (4) is equivalent to the update $\frac{1}{\hat{z}^*} \leftarrow \frac{1}{\hat{z}^*} + \psi$. To increase numerical stability, we add a regularization term to the cost function via a function $\mathbf{r}(\mathbf{x}) : \mathbb{R}^{6+n} \rightarrow \mathbb{R}^{6+n}$ for n movable vertices in the mesh, discussed in section 3.1. The cost function can be written as

$$\phi(\mathbf{x}) = \frac{1}{2}\left(||\mathbf{y}(\mathbf{x})||^2 + \lambda\,||\mathbf{r}(\mathbf{x})||^2\right) \tag{5}$$

where the scalar λ is used to balance the squared norms of $\mathbf{y}(\mathbf{x})$ and $\mathbf{r}(\mathbf{x})$. The update $\mathbf{x}$ is computed by linearizing the quadratic cost function and therefore solving the linear system

$$\left(\mathbf{J}_\mathbf{y}^\top\mathbf{J}_\mathbf{y} + \lambda\mathbf{J}_\mathbf{r}^\top\mathbf{J}_\mathbf{r}\right)\mathbf{x} = -\left(\mathbf{J}_\mathbf{y}^\top\mathbf{y}(\mathbf{0}) + \lambda\mathbf{J}_\mathbf{r}^\top\mathbf{r}(\mathbf{0})\right) \tag{6}$$

where $\mathbf{J}_\mathbf{y}$ and $\mathbf{J}_\mathbf{r}$ are Jacobians of the data and the regularization terms. This system is solved iteratively for $\mathbf{x}$ using *e.g.* its pseudo-inverse or Cholesky decomposition. The Jacobian $\mathbf{J}_\mathbf{y}$ can be written as the product $\mathbf{J}_\mathbf{y} = \mathbf{J}_{\widehat{\mathcal{I}}^*}\mathbf{J}_\mathbf{d}\mathbf{J}_\mathbf{G}$ where $\mathbf{J}_{\widehat{\mathcal{I}}^*}$ is the gradient of the estimated reference image, $\mathbf{J}_\mathbf{d}$ and $\mathbf{J}_\mathbf{G}$ are the Jacobians of the projection and the homography. In the spirit of [4], this first order linearization can be approximated to second order as $\mathbf{J}_\mathbf{y} = \frac{1}{2}\left(\mathbf{J}_{\widehat{\mathcal{I}}^*} + \mathbf{J}_{\mathcal{I}^*}\right)\mathbf{J}_\mathbf{d}\mathbf{J}_\mathbf{G}$ by including the gradient of the reference image $\mathbf{J}_{\mathcal{I}^*}$. As shown in the evaluation, this in general increases the convergence frequency of the Gauss-Newton optimization with low additional costs. The convergence area is increased by employing multiple levels of an image pyramid.

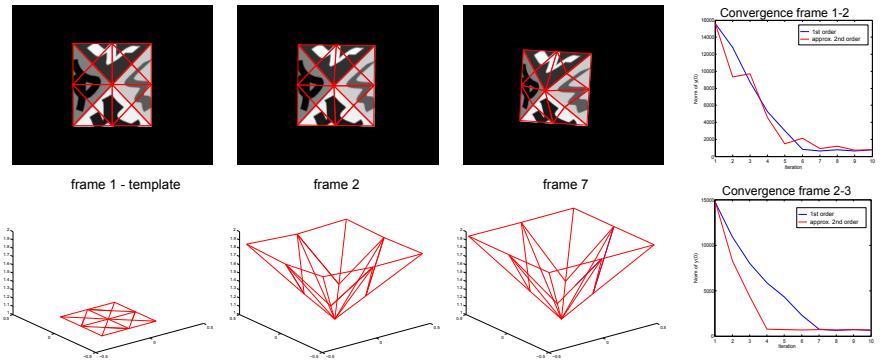
3.1 Regularization

In case the camera is close to the reference camera, the matrix $\mathbf{J}_\mathbf{y}^\top\mathbf{J}_\mathbf{y}$ becomes increasingly ill-conditioned, *i.e.* tiny changes in $\mathbf{y}(\mathbf{0})$ may provoke huge changes in $\mathbf{x}$. This is because the projection rays of the current camera are approximately aligned with those of the reference camera (depicted in Figure 1(b)). In this degenerate configuration, arbitrary movements of the vertices, respectively their inverse depth ψ_i, result in almost identical unwarped reference images $\widehat{\mathcal{I}}^*$.

However, this configuration can be easily mitigated by adding a regularization term to the cost function that restrains the vertices in that case. We define $\mathbf{r}(\mathbf{x})$ as $\mathbf{r}(\mathbf{x}) = (\mathbf{0}_{1\times 6}, r_1(\mathbf{x}), r_2(\mathbf{x}), \ldots, r_n(\mathbf{x}))^\top$ which currently only operates on the n movable vertices. We compute $\forall i \in 1, 2, \ldots, n$:

$$r_i(\mathbf{x}) = \left(1 + \lambda_s e^{-\lambda_r||\widehat{\mathbf{t}}||^2}\right)\left(\frac{1}{\widehat{\psi}_i + \psi_i} - \mu_i\right). \tag{7}$$

The first part of the regularization term is a weighting factor that penalizes the degenerate configuration just discussed. The scalars λ_s and λ_r determine the scale and range of the penalty concerning the baseline, empirically $\lambda_s = \lambda_r = 10$ gave good results. The second part of Equation (7) is responsible for damping the deformations and moving them towards their most likely true value. It penalizes changes of the depths with respect to a reference depth μ_i of the vertex.



(a) Convergence of shape (b) 1st vs. 2nd order

Fig. 2. Evaluation on synthetic data. (a) Sequence of synthetic pyramid, 16 faces and 12 moving vertices were used. Note that the shape quickly converges towards true shape from the template image (frame 1) to frame 2. (b) The proposed second order approximation of $\mathbf{J_y}$ converges 2-4 iterations earlier in case of slight deformations.

A naïve way of determining μ_i may consist in computing it as running average, *e.g.* updated after every image as $\mu_i \leftarrow 0.9\,\mu_i + 0.1/\widehat{\psi}_i$. This method is simple yet effective in case of a continuously moving camera. However, when the camera becomes stationary, μ_i will converge towards the value optimal for only this local configuration and information from distant successful registrations will be lost over time.

An improved version of determining μ_i tries to preserve previous knowledge about the camera motion. For this, we spatially sample height estimates of the proposed method on a hemisphere around each vertex using the geometry of the camera ray of the vertex in $\mathcal{I}^*$ and the current camera ray in $\mathcal{I}$. The samples are weighted using the angle between the rays, small angles are down-weighted as they represent (near-) aligned camera rays and thus lead to the degenerate configuration just discussed. Further we include into the weight the normalized cross-correlation of the adjacent faces of the vertex in both $\mathcal{I}^*$ and $\widehat{\mathcal{I}}^*$ to mitigate the influence of severely incorrect estimations of the camera pose or vertex heights. Typically, the value of μ_i changes rapidly in the beginning as the shape transforms from the initial estimate towards a more likely shape, but after that becomes relatively stable given sufficient camera movements.

4 Evaluation

The proposed method was quantitatively evaluated both on synthetic and real video sequences for which ground truth of the camera pose was available; in case of the synthetic sequence also the estimate of the shape was evaluated. Further, we evaluated the method qualitatively on smooth objects and on objects with creases, using a moving camera. Equally we tested our method on a smoothly deforming object with a fixed camera. Comparison against PTAM [10] was conducted in presence of several levels of blur. Videos of the evaluations can be found in the supplementary material.

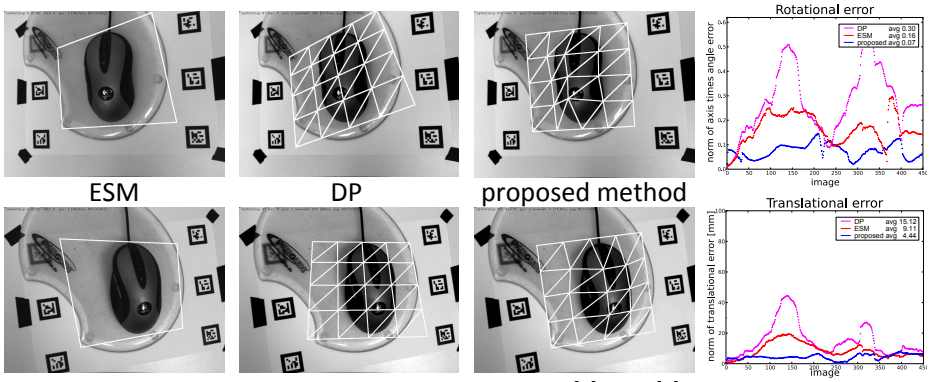
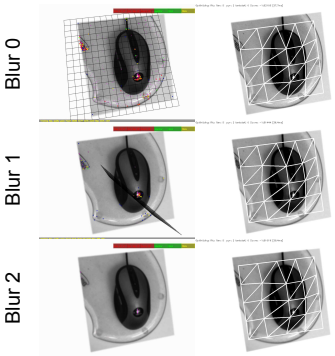


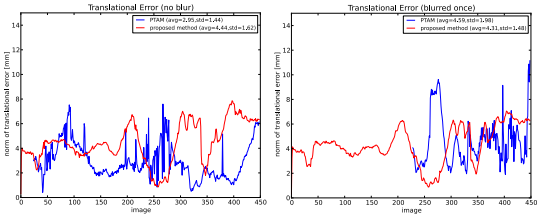
Fig. 3. Evaluation on real data. Comparison of ESM [3], DP [4] and proposed method. Poses were compared to ground truth from a mechanical measurement device.

4.1 Quantitative evaluation

Synthetic sequence A synthetic pyramid was created first seen from the top, then moving towards the lower left corner of the image while rotating. We used a mesh of 16 faces and 13 vertices from which only the central vertex was fixed. No regularization was employed as neither noise nor degenerate configurations are present and only a maximum of five iterations per frame on pyramid level 0, *i.e.* on the original resolution, were allowed. The method shows low errors in both pose and shape of the object. The synthetic evaluation is illustrated in Figure 2 and in the supplementary material. When comparing the first order linearization of $\mathbf{J_y}$ with the presented approximated second order linearization, we observed that they have similar convergence rate when there is strong motion in the depths like in frames 1-2 in Figure 2(b) . However, when the estimation of the structure is changing just slightly like in frames 2-3 shown on the bottom in Figure 2(b), 2 to 4 iterations may be saved and our results match those of Benhimane and Malis [4] in terms of convergence.

Real sequence To perform a quantitative evaluation with real camera images, we have created a sequence using a real camera mounted on a mechanical measurement device that provided a ground truth pose of the camera computed similarly to Lieberknecht *et al.* [11]. We made a sequence for tracking low textured target, a computer mouse on a mouse pad. Similar to the synthetic sequence, this sequence starts with an almost fronto-planar view such that we can create a reasonable reference image from it by rectifying the first image given the ground truth pose. The sequence was used to evaluate our method, ESM [3] and the calibrated multi-planar tracking method [4] referred to as DP. The algorithms were given identical parameters, *i.e.* 2 pyramid levels and 5 iterations per level. Poses were computed from the 2D–3D correspondences of the corners of the templates. As can be seen in Figure 3, our method outperforms planar methods in terms of accuracy on the pose of the camera. Furthermore, we evaluated the robustness of the proposed method with respect to blur introduced by consecu-





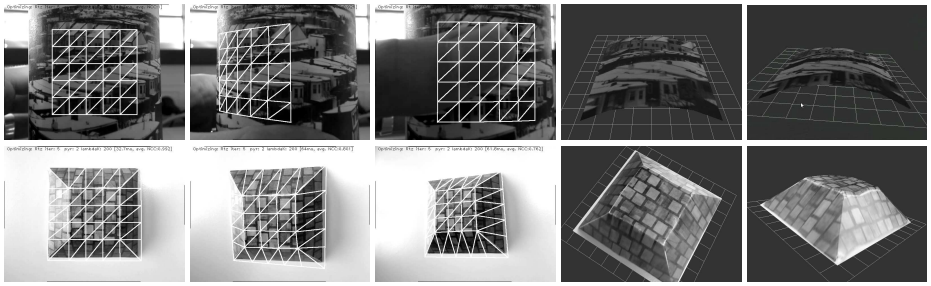
		level of blur				
		0	1	2	3	4
proposed	full	4.44, 1.62	4.32, 1.49	4.17, 1.43	4.04, 1.39	3.39, 1.38
	cropped	4.44, 1.62	4.31, 1.48	4.17, 1.43	4.03, 1.39	3.39, 1.38
PTAM	full	2.49, 0.70	1.70, 0.58	2.00, 0.73	1.70, 0.60	1.61, 0.62
	cropped	2.95, 1.44	4.59, 1.98			

Fig. 4. Quantitative evaluation against blur. A (5×5) mean filter was applied consecutively 0–4 times to evaluate the robustness of the method to blur. *Left:* Frame of the blurred image sequences given to PTAM (left) and proposed method (right). In case of PTAM the plane indicates the ground plane PTAM fits to available features after initialization. In our case we show the deformed 3D mesh model. *Right:* Error in translation for cropped images and blur levels 0 and 1, below a table displaying the mean error and standard deviation of the methods.

tively applying a (5×5) mean filter. This kind of blur can be found in real data when the object is out-of-focus given a fixed-focus camera. We observed that the accuracy of the method did increase slightly as the blur increased. The same sequences were given to PTAM. As poses of PTAM are defined in an rather arbitrary coordinate system, we aligned them by minimizing the sum-of-squared distance to the ground truth, solving for a 6-DOF transformation and 1-DOF scale. In order to make fair comparisons, we focused only on the area belonging to the object and cut the markers out. To avoid synthetic stable features, like those on the edges of the cut, we slightly randomized the borders of the mask. PTAM could not successfully initialize starting from the second level of blur, as depicted in Figure 4, since there were very few features on the lowest image pyramid level to be tracked. However the accuracy of PTAM is superior when using the full image as shown in Figure 4. The proposed method is giving the same results both for full and cropped image.

4.2 Qualitative evaluation

To analyze how the method works in case of a smooth object and in case of object with creases, we evaluated it by tracking a cup and a truncated pyramid. The method was able to track both objects well and approximated the shapes reasonably. As noted in [5], best results are obtained when the structure of the mesh is able to express the structure of the underlying object. Furthermore, we evaluated the robustness of the method when tracking deformed objects. Although this violates the rigidity assumption, the method copes well with slight deformations as shown in Figure 5. In the cup sequence, after estimating the shape we manually disabled the estimation of the depths and used the method only for tracking the pose. We show that the pose is well estimated even under severe occlusion of up to 50% of the mesh. On a 2.5 GHz dual core notebook,



(a) Evaluation for rigid objects.

(b) Evaluation for a deforming object.

Fig. 5. Qualitative evaluation of recovered shapes. The first frame of the sequence (shown left-most) is used as template. (a) Shape recovery of the rigid objects from moving camera where also the camera motion is estimated. (b) Recovering shape of the deforming object where the camera is not moving. Although the method was not designed for such situations, we still managed to apply it to recover moderate object deformations.

the speed is typically 10–30 ms per frame when estimating the camera pose and around 40–60 ms when additionally estimating the deformations. The timings were obtained using pyramid levels 3 and 2, at most 5 iterations per level and a mesh of approximately 200×200 pixels on level 0. Most of the time is spent in the direct computation of $\mathbf{J}_\mathbf{y}^\top \mathbf{J}_\mathbf{y}$.

4.3 Discussion and future work

During the evaluation, we observed that the main source of error originated from fast translational camera motion as this violates our assumption of small motion considerably. However, we believe that this could be mitigated by using active search, *e.g.* by employing a motion model. To further increase robustness, we plan to investigate in a regularization term that penalizes deformation caused by errors in camera tracking. In addition, we plan to add the possibility of dynamically extending the deformed template as camera moves around.

5 Conclusion

We presented a real-time method for simultaneous tracking and reconstruction of non-planar templates. While we remove the planarity constraint inherent to classical template tracking, we still benefit from all available pixels of the template when building our objective function. We do not impose any constraints on the model deformation, therefore we can equally reconstruct and track templates that are smooth or have creases. The tracking precision of our method is very good compared to the ground truth. This proves that even with only an

approximate shape of the template recovered on-line, the tracking is more stable compared to planar template tracking methods. Furthermore, and in contrast to SfM and SLAM methods, the proposed algorithm still works well for low textured objects and in presence of strong blur.

References

1. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. IJCV 56(3), 221–255 (2004)
2. Bartoli, A., Zisserman, A.: Direct estimation of non-rigid registrations. In: BMVC (2004)
3. Benhimane, S., Malis, E.: Homography-based 2d visual tracking and servoing. Special Joint Issue IJCV/IJRR on Robot and Vision. Published in The International Journal of Robotics Research 26(7), 661–676 (July 2007)
4. Benhimane, S., Malis, E.: Integration of euclidean constraints in template based visual tracking of piecewise-planar scenes. In: IROS (2006)
5. Datta, A., Sheikh, Y., Kanade, T.: Linear motion estimation for systems of articulated planes. In: CVPR (2008)
6. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. PAMI 26(6), 1052–1067 (2007)
7. Gay-Bellile, V., Bartoli, A., Sayd, P.: Direct estimation of non-rigid registrations with image-based self-occlusion reasoning. PAMI 32, 87–104 (2009)
8. Hilsmann, A., Schneider, D., Eisert, P.: Realistic cloth augmentation in single view under occlusion. Computers & Graphics (2010)
9. Jurie, F., Dhome, M.: Hyperplane approximation for template matching. PAMI 24, 996–1000 (2002)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: ISMAR (2007)
11. Lieberknecht, S., Benhimane, S., Meier, P., Navab, N.: A dataset and evaluation methodology for template-based tracking algorithms. In: ISMAR (2009)
12. Lucas, B., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: IJCAI (1981)
13. Newcombe, R., Davison, A.: Live dense reconstruction with a single moving camera. In: CVPR (2010)
14. Pilet, J., Lepetit, V., Fua, P.: Fast non-rigid surface detection, registration and realistic augmentation. IJCV 76(2), 109–112 (2008)
15. Salzmann, M., Fua, P.: Reconstructing sharply folding surfaces: A convex formulation. In: CVPR (2009)
16. Salzmann, M., Urtasun, R., Fua, P.: Local deformation models for monocular 3d shape recovery. In: CVPR (2008)
17. Silveira, G., Malis, E.: Unified direct visual tracking of rigid and deformable surfaces under generic illumination changes in grayscale and color images. IJCV 89(1), 84–105 (2010)
18. Triggs, B., Mclauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment – a modern synthesis. In: Proceedings of the International Workshop on Vision Algorithms: Theory and Practice (2000)
19. Varol, A., Salzmann, M., Tola, E., Fua, P.: Template-free monocular reconstruction of deformable surfaces. In: ICCV (2009)
20. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: Statistical and Geometrical Approaches to Visual Motion Analysis, chap. An Improved Algorithm for TV-L1 Optical Flow, pp. 23–45. Springer-Verlag (2009)