Sensor Fusion for Sparse SLAM with Descriptor Pooling

Philipp Tiefenbacher(✉), Julian Heuser, Timo Schulze, Mohammadreza Babaee, and Gerhard Rigoll

Institute for Human-Machine Communication, Technical University of Munich, Munich, Germany
{philipp.tiefenbacher,reza.babaee,rigoll}@tum.de, j.heuser@mytum.de, schulzetimo@gmx.net

Abstract. This paper focuses on the advancement of a monocular sparse- SLAM algorithm via two techniques: Local feature maintenance and descriptor-based sensor fusion. We present two techniques that maintain the descriptor of a local feature: *Pooling* and *bestfit*. The maintenance procedure aims at defining more accurate descriptors, increasing matching performance and thereby tracking accuracy. Moreover, sensors besides the camera can be used to improve tracking robustness and accuracy via sensor fusion. State-of-the-art sensor fusion techniques can be divided into two categories. They either use a Kalman filter that includes sensor data in its state vector to conduct a posterior pose update, or they create world-aligned image descriptors with the help of the gyroscope. This paper is the first to compare and combine these two approaches. We release a new evaluation dataset which comprises 21 scenes that include a dense ground truth trajectory, IMU data, and camera data. The results indicate that descriptor pooling significantly improves pose accuracy. Furthermore, we show that descriptor-based sensor fusion outperforms Kalman filter-based approaches (EKF and UKF).

1 Introduction

Handhelds are ubiquitous and are usually equipped with a video camera which enables the integration of simultaneous localization and mapping (SLAM). Handhelds also include additional sensors, the inertial measurement units (IMUs), which can improve the SLAM accuracy [1].

The combination of the video capture and the additional sensor data requires a multi-sensor fusion. This is commonly achieved by Kalman filters [2–4]. Besides the Kalman filter approaches, a vision-based approach exists that improves the image descriptor via the gyroscope data [5]. This work compares the sensor fusion via an unscented Kalman filter (UKF) with the sensor fusion via gravity-aligned feature descriptors (GAFD) [5]. Both approaches are integrated into the parallel tracking and mapping (PTAM) algorithm [6].

Furthermore, we change the patch-based PTAM matching to a descriptor based matching, e.g., SIFT [7]. Image feature detection aims at detecting salient

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part III, LNCS 9915, pp. 698–710, 2016.
DOI: 10.1007/978-3-319-49409-8_58

positions in images at which descriptors are extracted that are robust in terms of scale and rotation. This modification allows us to propose two new descriptor maintenance techniques for an improved matching and tracking accuracy.

In summary, the contributions of this work are the following: *(a)* a new dataset for the evaluation of SLAM algorithms, *(b)* a new descriptor maintenance technique for higher pose accuracy, *(c)* a two-way sensor fusion technique by combining UKF with GAFD.

2 Related Work

The PTAM [6] algorithm belongs to the keyframe-based monocular SLAM methods. It differs from the filtering-based approaches [8]: The knowledge of the system is not represented by a probability distribution but by a subset of images (keyframes) and map points. The PTAM map constitutes a sparse scene representation since only patches of salient image points are incorporated. Sparse SLAM approaches usually allow for faster computation than dense SLAM approaches. Recent works such as ORB-SLAM [9] show that sparse-SLAM techniques can outperform semi-dense ones [10].

Direct visual odometry (VO) techniques utilize the full image information. For instance, dense tracking and mapping (DTAM) [11] is able to reconstruct the map in much more detail than the sparse SLAM techniques but the computational complexity is still too demanding to achieve real-time performance on handhelds. The so called semi-dense techniques [10,12] calculate dense depth maps covering all image regions with non-negligible gradient. Optimized versions of this technique [13] run in real time on handhelds.

Our work integrates IMU output into a sparse SLAM technique. Several works already targeted the integration of inertial sensors into SLAM approaches by using Kalman filters. For example, Omari et al. [14] reviewed an optical flow-based visual system coupled with inertial measurement units. Their unscented Kalman filter (UKF) considered gyroscope and accelerometer measurements. Tiefenbacher et al. [1] used the IMU data as control input for the UKF. Furthermore, a motion model based on the a priori estimate of the UKF was presented. Aksoy and Alatan [2] focused on the uncertainty modeling for a Kalman filter which was combined with a tracking system similar to PTAM.

Besides the sensor fusion via Kalman filters, few works incorporated the IMU data directly into the visual descriptors. Kurz and Benhimane [5] proposed the gravity-aligned feature descriptors (GAFD) that align the orientations of local feature descriptors, e.g., SIFT [7] and SURF [15], to the gravitational force obtained from the gyroscope. They showed that GAFD increases the number of successfully matched features since the descriptors become not just invariant to orientation but, more importantly, distinguishable. Guan et al. [16] presented gravity-aligned VLAD features [17] to incorporate the same advantages as in [5]. Our work is the first that combines and evaluates both ways of sensor fusion: Filter- and feature-based fusion.

3 Descriptor-Pooled PTAM with Sensor Fusion

PTAM separates tracking and mapping into two threads. After map initialization, the positions of FAST [18] corners are used to extract patches. These patches are saved into the map and successively tracked. In each new frame, a motion model delivers a prior pose estimate. Then template matching between new patches and the warped patches of the map is applied for those patches which fulfill the epipolar constraint. A pyramid-based [19] matching approach leads to a coarse-to-fine pose estimation and accelerates execution time. The map is updated via keyframes in case of too few successful matches. The following sections present the adaption of PTAM to descriptor-based matching and the two main contributions: Local descriptor update strategies and sensor fusion through adaption of local descriptors or a Kalman filter.

3.1 Tracking and Mapping

The matching over pixel intensities of warped patches is exchanged with scale- and rotation-invariant keypoint descriptors. A guided nearest neighbor search identifies the best keypoints for map initialization, tracking and the mapping process. The number of pyramid levels have been reduced from four to three, since the forth level is too blurred to detect meaningful corner-based (FAST) keypoints. The two-stage coarse-to-fine tracking process is preserved. At the lowest pyramid level, we extract at most 1850 keypoints.

The advanced map stores multiple keypoints, called map points, for every keyframe. The map points hold the mapping to their descriptors. Multiple descriptors for each map point are permitted. We implemented and evaluated the descriptor-based PTAM using the prominent SIFT [7] and efficient ORB [20] descriptors. The 128-/32-dimensional descriptors are matched via euclidean and Hamming distances, respectively.

3.2 Descriptor Update

We investigate on three different descriptor maintenance strategies. The first and most trivial strategy is to keep the descriptor $\boldsymbol{d}$ of a map point $\boldsymbol{p}_m$ fixed to the descriptor of its source keyframe k_{source}. The source keyframe is the keyframe that initially creates the map point. The descriptor is given by

$$\boldsymbol{d}_{p_m} = \boldsymbol{d}_{k_{source}}. \tag{1}$$

The second strategy computes the *bestfit* descriptor $\boldsymbol{d}$ of a map point $\boldsymbol{p}_m$. Since multiple keyframes of the map may contain measurements of the same map point, the best fitting descriptor minimizes the sum of the distances to all other descriptors that are linked to this map point $\boldsymbol{p}_m$. It is given by

$$\boldsymbol{d}_{p_m} = \min_i \sum_{j,j\neq i} \|\boldsymbol{d}_i - \boldsymbol{d}_j\|_{L2 \text{ or Hamming}}; \qquad \boldsymbol{d}_i, \boldsymbol{d}_j \in D \tag{2}$$

with D being the set of descriptors associated with the map point.

The third strategy computes temporally pooled descriptors. *Pooling* describes the combination of feature descriptors at nearby locations with the goal to achieve a joint feature representation "that preserves important information (intrinsic variability) while discarding irrelevant details (nuisance variability)" [21]. Dong and Soatto [22] showed that domain-size pooling of gradient histogram descriptors improves the matching performance significantly. However, this benefit comes with higher computational cost since multiple descriptors with different domain-sizes have to be computed for nearby locations.

We propose a *pooling* over time instead of nearby locations. Thus, a (median) *pooling* occurs over all descriptors that are linked to one map point $\boldsymbol{p}_m$ but captured at different points in time, i.e., different keyframes. For that purpose, we generate a sorted list for each entry of a descriptor and select the middle value, which easily removes outliers.

The descriptor of a particular map point is updated whenever a new keyframe is added that contains a measurement of the map point. In consequence, only a small computational overhead is introduced by this new descriptor update technique. Furthermore, the mapping process tries to refine map points in other keyframes that have not been searched before or have been rated as a bad measurement. If the mapping process is successful in this search, the descriptor of the corresponding map point will be updated. The entire descriptor update is handled by the mapping thread.

3.3 Sensor Fusion

Kalman Filter-Based Fusion. In order to project map points into the current frame, a motion model has to predict a prior pose. The precision and stability of the motion model has a direct influence on tracking performance. For our experiments, we make use of the PTAM motion model (PMM) and the findings in [1].

PMM consists of a decaying velocity model. It can be formalized using exponential coordinates

$$P_i = \exp(\boldsymbol{\tau_i}) \cdot P_{i-1} \tag{3}$$

with the decaying linear and angular velocity being

$$\boldsymbol{\tau}_i = 0.9 \cdot (0.5 \cdot \boldsymbol{\tau_{i-1}} + 0.5 \cdot \ln(P_{i-1} \cdot P_{i-2}^{-1})). \tag{4}$$

This motion model is used for both PTAM and the PTAM+ [1] variants. For the latter, an unscented Kalman filter (UKF) additionally updates the final PTAM pose leading to a posterior pose estimate (similar to [1]). The state vector is identical to [1]. It has 26 dimensions and contains, among others, the IMU-to-world attitude, the position and velocity vectors as well as the gravity vector. The employed UKF update was most stable in estimating the attitude [1], since the UKF keeps track of the gravity vector. We do not make use of the accelerometer measurements as those tend to corrupt the position estimate of SLAM quadratically.

Gravity-Aligned Feature Descriptors (GAFD). We utilize the IMU data in the tracking and mapping process by aligning the descriptors to the gravity vector [23] prior to matching. This creates distinguishable descriptors for congruent features that would not be distinguishable with the basic algorithms, e.g., SIFT. We track the gravity vector in the fixed world frame with a UKF. For every frame, the gravity vector $\boldsymbol{g}_C$ in the moving camera frame is updated via

$$\boldsymbol{g}_C = {}^C\boldsymbol{R}_W \boldsymbol{g}_W \tag{5}$$

with the gravity vector $\boldsymbol{g}_W$ in the world frame and the world to camera rotation matrix ${}^C\boldsymbol{R}_W$. The gravity vector is projected onto the image plane by applying the camera model and the intrinsic camera calibration matrix $\boldsymbol{K}$. Its $2D$-orientation in the image plane is computed with respect to the location of the keypoint. The final $2D$-projection of the gravity vector $\boldsymbol{d} = [d_u, d_v, 0]^T$ at a pixel $\boldsymbol{p} = [u, v, 1]^T$ is computed by

$$\boldsymbol{d} = \boldsymbol{p}' - \boldsymbol{p} \tag{6}$$

with

$$\boldsymbol{p}' = [u', v', 1]^T = \frac{1}{1 + g_{C_z}} (\boldsymbol{p} + \boldsymbol{K}\boldsymbol{g}_C). \tag{7}$$

Finally, the orientation angle of the gravity-aligned descriptor is given by

$$\theta = \arctan \frac{d_v}{d_u}. \tag{8}$$

4 Dataset

Publicly available datasets lack either gyroscope and accelerometer data [24,25] or a dense ground truth trajectory [26]. Hence, we recorded new scenes including these data. Furthermore, we focused on recording fast movements and pure rotations which are very challenging for most visual SLAM approaches. The new dataset consists of 21 scenes that we make available at http://www.mmk.ei.tum.de/sensorintegrationslam/. A professional external tracking system by *ART* provides the ground truth of 21 motion trajectories. The tracking system includes

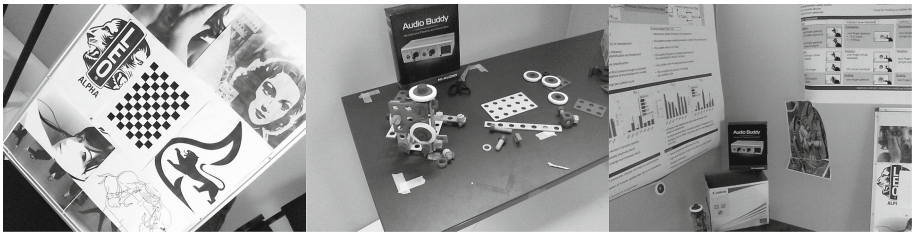

Fig. 1. Camera capture of scene 1 (left), scene 15 (center) and scene 18 (right). These captures are picked to illustrate the environment. The other scenes are recorded in the same environment.

Table 1. Absolute values of each low-level feature for each scene and the corresponding cluster assignment.

	Transl. distance [m]	Rot. distance [deg]	Transl. vel. [m/s]	Rot. vel. [deg/s]
Scene 1	23.823 (*Low*)	3948 (Mod.)	0.2362	39.1384
Scene 2	27.549 (*Low*)	6318 (**High**)	0.2908	66.7306
Scene 3	22.319 (*Low*)	4522 (Mod.)	0.2205	44.6794
Scene 4	46.718 (Mod.)	2967 (*Low*)	0.4590	29.1496
Scene 5	46.33 (Mod.)	4672 (Mod.)	0.5331	53.7486
Scene 6	59.747 (**High**)	2730 (*Low*)	0.6453	29.4867
Scene 7	56.13 (**High**)	2560 (*Low*)	0.6219	28.4292
Scene 8	54.74 (**High**)	6245 (**High**)	0.5968	68.0944
Scene 9	46.144 (Mod.)	4296 (Mod.)	0.6090	56.6942
Scene 10	63.447 (**High**)	4609 (Mod.)	0.7079	51.4246
Scene 11	33.078 (*Low*)	2235 (Low)	0.4229	28.5781
Scene 12	41.227 (Mod.)	2470 (*Low*)	0.4019	24.0790
Scene 13	46.492 (Mod.)	3748 (Mod.)	0.4795	38.6584
Scene 14	49.472 (Mod.)	4573 (Mod.)	0.5683	52.5257
Scene 15	18.942 (*Low*)	1662 (*Low*)	0.4083	35.8284
Scene 16	25.225 (*Low*)	3442 (Mod.)	0.3314	45.2329
Scene 17	39.73 (Mod.)	3379 (Mod.)	0.3496	29.7341
Scene 18	23.54 (*Low*)	1866 (*Low*)	0.3153	24.9886
Scene 19	10.534 (*Low*)	963.4 (*Low*)	0.2352	21.5068
Scene 20	38.262 (Mod.)	5101 (Mod.)	0.3475	46.3317
Scene 21	50.414 (Mod.)	5998 (**High**)	0.4563	54.2867

five high-resolution cameras that record the trajectories at a sampling rate of 60 Hz. The mobile device is equipped with passive markers. In order to validly compare the SLAM-based tracking results with the external tracking system, a hand-eye calibration between markers and the camera has been applied [27]. Besides the ground truth trajectory, each scene consists of a 20 Hz grayscale camera capture with a resolution of 640 × 480 and 60 Hz IMU readings (gyroscope and accelerometer) obtained from a "Microsoft Surface 2 Pro". Figure 1 illustrates camera captures of three different scenes.

In order to characterize each scene in more detail, we extract scene properties of the provided ground truth motion trajectory. The first and last 10 % of the frames are discarded for every scene, since they are either needed for the map initialization or the tracking already failed. Then we perform k-means clustering on the euclidean distances of each scene property (1D) with three cluster centers (low, moderate and high). The chosen scene properties are the overall translation distance (TD) and the overall rotation distance (RD). The velocities are proportional to the corresponding distances since each scene has a duration of 100 to 120 s.

Table 1 depicts the absolute values of the low-level features of each scene as well as the corresponding clusters. It is important to note that the number of scenes per cluster is not equally distributed. For instance, in RD only three of the 21 scenes belong to the *High* cluster, while 8 scenes are matched to the *Low* cluster. The cluster *TD High* starts at 54.74 m until 63.45 m, while the range of the *TD Low* cluster lies between 10.53 m and 33.08 m. The average translation velocities range from 0.22 m/s to 0.71 m/s, while the average rotation velocities range from 21.51 deg/s to 68.08 deg/s.

5 Results

The evaluations consider the first 350 frames of every scene due to tracking failures. The patch versus descriptor matching as well as the sensor fusion techniques are only evaluated via the absolute trajectory error (ATE), since the results are already very distinctive. Both ATE and rotation error are computed for the descriptor update techniques. We calculate the quaternion-based rotation error of Gramkow [28] for every frame. Afterwards, we averaged the rotation error per scene. The clusters results displayed in the figures depict the median of the averaged rotation errors. The ATE is computed via the publicly available script of [25].

5.1 Patch Versus Descriptor Matching

In Fig. 2, we compare the ATE results of the scenes that were trackable by all approaches. It can be seen that the patch-based PTAM performs similarly to ORB-PTAM with a slight advantage in scene 14. The gravity-aligned (GA)-ORB-PTAM performs better than the standard ORB-PTAM in 6 of 7 scenes. Moreover, the gravity-aligned features mostly improve tracking in cases of pure rotation trajectories (see *RD High* in Fig. 3). The SIFT-PTAM algorithm completely fails to track in those cases. That is why only this small subset could be evaluated here. Furthermore, SIFT-PTAM leads to higher ATE than ORB-PTAM in every tested scene.

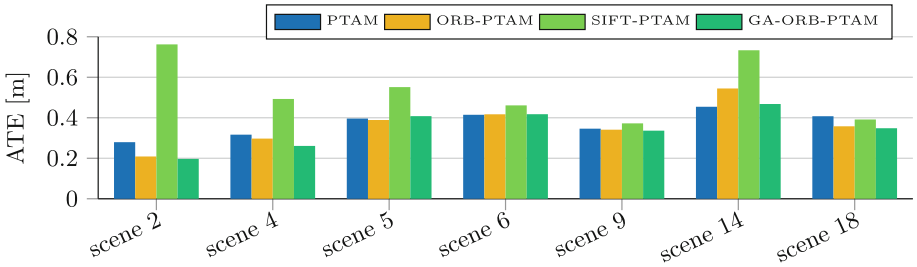

Fig. 2. The ATE [m] of patch-based versus descriptor-based matching.

5.2 Descriptor Update Techniques

For the evaluation of the descriptor update techniques, the original PTAM algorithm is adapted to ORB-PTAM with GAFD and without. No Kalman filter for a posterior update is used. Moreover, the size of set D of the associated descriptors is not limited for *pooling* and *bestfit* (see Eq. (2)). Figure 3 consists of 6 different trajectory clusters, however, the scenes 2, 7, 11, 15, 17, and 22 are excluded due to tracking errors in the *source* descriptor technique. It can be seen that the ATEs do not differ much except for GAFD and non-GAFD descriptor updates. GAFD produces the smallest error in every cluster. Considering only the GAFD versions, the *pooling* reaches slightly lower errors than the other descriptor update strategies. Moreover, the new descriptor techniques perform only worse than the *source* technique for the *RD High*. For the non-GAFD versions, there is no clear difference and source even performs best in *RD Low*.

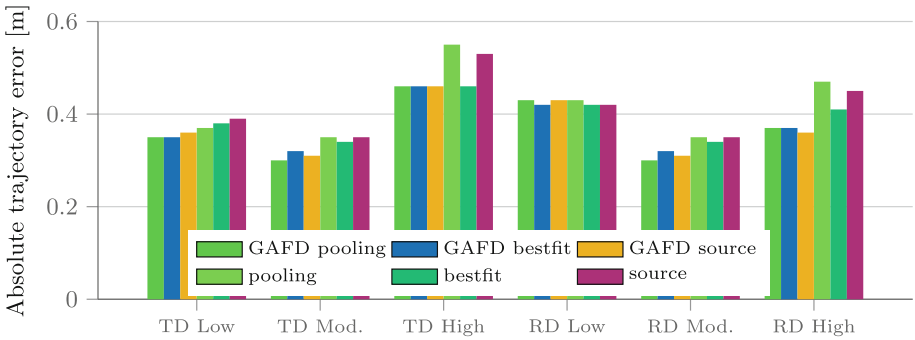

Fig. 3. The ATE [m] of the descriptor maintenance techniques for GAFD and non-GAFD extensions. Only the ORB descriptor is used for this comparison.

Figure 4 presents the rotational pose error (RPE). The rotational pose errors differ more significantly between the techniques. Here, *pooling* leads to a rotation error reduction compared to the *bestfit* and the *source* techniques except for *RD High*. There, the *bestfit* technique produces a lower RPE than *pooling*, whereas the *pooling* technique performs best in 5 of 6 cluster. In two clusters (*TD Mod* and *RD Low*) the *bestfit* approach leads to worse results than the *source* technique.

The local descriptors without the GAFD extension never state the best results. Additionally, the error variations between the descriptor maintenance techniques are much larger without GAFD than with GAFD. This shows that GAFD stabilizes the local features on the one hand, but on the other hand, outlines the importance of the descriptor maintenance technique, i.e., pooling, in case of non-GAFD. There is no relation between the clusters and the descriptor maintenance techniques.

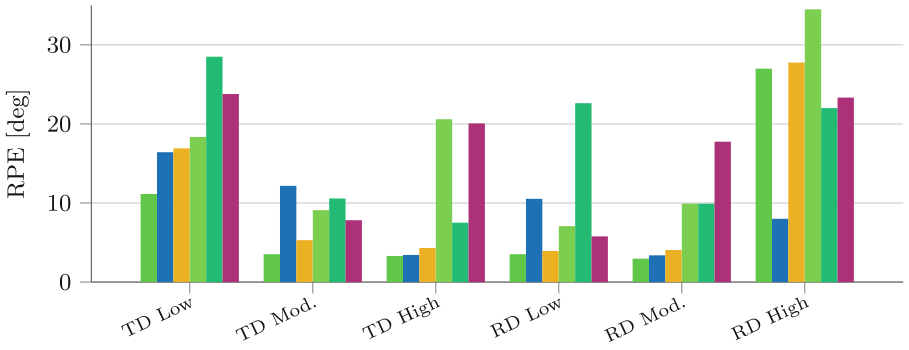

Fig. 4. Rotational pose errors (RPE) [deg] of the GAFD- and non-GAFD ORB features for different descriptor maintenance techniques. Same color scheme applies as in Fig. 3.

5.3 Sensor Integration: GAFD Versus GAFD and Kalman Filter

In this section, we evaluate whether a combination of GAFD and a Kalman filter can further improve the tracking accuracy. Every descriptor-based tracking algorithm incorporates *pooling* as descriptor maintenance technique. Furthermore, 17 of 21 scenes have been used for the evaluation. The scenes 7, 11, 17, and 19 either failed to initialize the map or the tracking failed after a certain time. SLAM algorithms that include a posterior pose update via UKF are marked with a plus (+). Besides the original PTAM, also PTAM+ [1] as well as EKF-SLAM [29] are included in our comparisons. Figure 5 illustrates the ATE of all clusters and SLAM techniques.

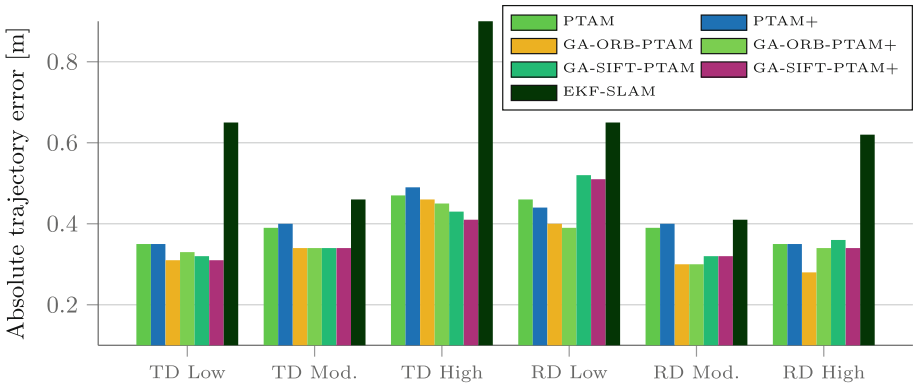

Fig. 5. Absolute trajectory errors [m] of GAFD versus GAFD and UKF (+).

The EKF-SLAM algorithm achieves the worst performance in all clusters. The GA(FD)-ORB-PTAM tracker shows the lowest error in 4 out of 6 clusters. Thus, GA-ORB-PTAM is the best algorithm in this comparison. The additional

UKF (+) for a posterior pose update does improve accuracy only for the clusters *TD High* and *RD Low*, whereas the ATE increases in the cluster with large rotations *RD High*. The GA-ORB-PTAM presents a recognizable smaller error than GA-ORB-PTAM+, which might be caused due to the accumulation of the error in the Kalman filter. This is expected since the gravity-aligned features stabilize the matching in these cases, whereas algorithms such as PTAM and also ORB-SLAM [9] struggle to identify the correct trajectories. The original PTAM algorithm is outperformed in every cluster.

The SIFT descriptor produces higher trajectory errors than the ORB descriptor, especially in *RD Low*. This is related to the fact that SIFT was unable to detect enough keypoints, while FAST produced enough keypoints for the ORB descriptor extraction and matching. It is of interest that for the GA-SIFT implementations an additional UKF (+) improved the tracking accuracy. Thus, a posterior pose update based on the sensor data is more beneficial in case of larger trajectory errors.

6 Computational Costs

The functional runtimes of every frame have been calculated and averaged over all sets. Table 2 depicts the function runtimes of keypoint extraction (KE), descriptor extraction (DE), tracking (T), relocalization (R) and motion model (MM). The experiments have been carried out on an Intel Core i7-4600U (2 cores @ 2.1 GHz).

Table 2. Mean runtimes of different PTAM approaches [ms].

	PTAM	PTAM+	ORB PTAM+	GA-ORB PTAM+	GA-SIFT PTAM+
KE	3.99	4.36	9.83	5.33	488.10
DE	-	-	26.11	16.78	646.08
T	18.21	19.10	13.37	10.07	7.03
R	1.18	1.69	2.17	0.72	2.19
MM	2.09	1.16	1.1	0.64	1.38
Total	25.47	26.30	52.60	33.54	1144.79

It can be seen that the descriptor-based versions perform slightly faster in the tracking stage (T) than their patch-based counterparts. The patch-based PTAM variants create warped patches on four pyramid levels, while the descriptor-based PTAM considers only three pyramid levels. The descriptor-based versions involve a computationally expensive descriptor extraction (DE) stage, which can be completely omitted in the patch-based versions.

Interestingly, ORB-PTAM+ is more time consuming than GA-ORB-PTAM+. The GAFD version allows to detect duplicated keypoints, which are erased prior to the descriptor extraction, thus saving time. The SIFT versions

have been parametrized with a smaller number of keypoints to extract (∼1000 keypoints), therefore, the matching process is faster in the tracking stage compared to ORB. The total runtimes show that GAFD-ORB-PTAM is around 6 ms slower than the original PTAM algorithm which runs the fastest (25.47 ms), but GAFD-ORB-PTAM is still real-time capable (∼32 ms). The SIFT variant is not of practical use even though the number of keypoints has been limited.

7 Conclusion

We proposed two new descriptor maintenance techniques, temporal *pooling* and *bestfit*. The (median) *pooling* technique produced the highest rotational pose accuracy and the lowest absolute trajectory error. Moreover, the new techniques also increased tracking accuracy in combination with gravity-aligned (GA) features, but were even more beneficial when GA features were missing as in cases that lack gyroscope data. The local descriptors without gravity alignment varied much more revealing potential for further improvements. For the sensor fusion, we revealed that a posterior update with a UKF did not improve tracking accuracy if the sensor information is already included into the local features. While still real-time capable, the GA-ORB-PTAM algorithm performed better than the original PTAM, PTAM+ (UKF) as well as EKF-SLAM. Thus, we recommend to use gravity-aligned descriptors instead of incorporating the gyroscope data in a Kalman filter.

Future work should try to find even better maintenance techniques for the feature descriptors. For example, the descriptors could be selected based on a trade-off between the current velocity (motion blur) and viewing angle (texture quality).

References

1. Tiefenbacher, P., Schulze, T., Rigoll, G.: Off-the-shelf sensor integration for mono-SLAM on smart devices. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, pp. 15–20. IEEE (2015)
2. Aksoy, Y., Alatan, A.A.: Uncertainty modeling for efficient visual odometry via inertial sensors on mobile devices. In: Proceedings of the International Conference on Image Processing. IEEE (2014)
3. Julier, S.J., Uhlmann, J.K.: A new extension of the Kalman filter to nonlinear systems. In: Proceedings of Signal Processing, Sensor Fusion, and Target Recognition, vol. 3068, pp. 182–193 (1997)
4. Servant, F., Houlier, P., Marchand, E.: Improving monocular plane-based SLAM with inertial measures. In: Proceedings of the International Conference on Intelligent Robots and Systems, pp. 3810–3815. IEEE (2010)
5. Kurz, D., Benhimane, S.: Inertial sensor-aligned visual feature descriptors. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 161–166. IEEE (2011)
6. Klein, G., Murray, D.: Improving the agility of keyframe-based SLAM. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5303, pp. 802–815. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88688-4_59

7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Springer Int. J. Comput. Vis. **60**(2), 91–110 (2004)
8. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1151–1156. IEEE (2003)
9. Mur-Artal, R., Montiel, J., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015)
10. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: Proceedings of the International Conference on Computer Vision, pp. 1449–1456. IEEE (2013)
11. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: Proceedings of the International Conference on Computer Vision, pp. 2320–2327. IEEE (2011)
12. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10605-2_54
13. Schöps, T., Engel, J., Cremers, D.: Semi-dense visual odometry for AR on a smartphone. In: Proceedings of the International Symposium on Mixed and Augmented Reality, pp. 145–150. IEEE (2014)
14. Omari, S., Ducard, G.: Metric visual-inertial navigation system using single optical flow feature. In: Proceedings of the European Control Conference, pp. 1310–1316. Springer (2013)
15. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Elsevier Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
16. Guan, T., He, Y., Gao, J., Yang, J., Yu, J.: On-device mobile visual location recognition by integrating vision and inertial sensors. IEEE Trans. Multimed. **15**(7), 1688–1699 (2013)
17. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 3304–3311. IEEE (2010)
18. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: Proceedings of the International Conference on Computer Vision, vol. 2, pp. 1508–1515. IEEE (2005)
19. Burt, P., Adelson, E.: The Laplacian pyramid as a compact image code. IEEE Trans. Commun. **31**(4), 532–540 (1983)
20. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of the International Conference on Computer Vision, pp. 2564–2571. IEEE (2011)
21. Boureau, Y.L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the International Conference on Machine Learning, pp. 111–118 (2010)
22. Dong, J., Soatto, S.: Domain-size pooling in local descriptors: DSP-SIFT. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 5097–5106. IEEE (2015)
23. Kurz, D., Benhimane, S.: Handheld augmented reality involving gravity measurements. Elsevier Comput. Graph. **36**(7), 866–883 (2012)
24. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 3354–3361. IEEE (2012)

25. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: Proceedings of the International Conference on Intelligent Robot Systems, pp. 573–580. IEEE (2012)
26. Ovrén, H., Forssén, P.E.: Gyroscope-based video stabilisation with auto-calibration. In: Proceedings of the International Conference on Robotics and Automation, pp. 2090–2097. IEEE (2015)
27. Bianchi, G., Wengert, C., Harders, M., Cattin, P., Szkely, G.: Camera-marker alignment framework and comparison with hand-eye calibration for augmented reality applications. In: Proceedings of the ISMAR, pp. 188–189 (2005)
28. Gramkow, C.: On averaging rotations. Springer J. Math. Imaging Vis. **15**(1–2), 7–16 (2001)
29. Civera, J., Grasa, O.G., Davison, A.J., Montiel, J.M.M.: 1-Point RANSAC for extended Kalman filtering: application to real-time structure from motion and visual odometry. ACM J. Field Robot. **27**(5), 609–631 (2010)