Evaluation of Lightweight Local Descriptors for Level Ground Navigation with Monocular SLAM

Weiya Chen, Yulin Wan, Shiqi Ou(✉), and Zhidong Xue(✉)

School of Software Engineering, Huazhong University of Science and Technology, Wuhan, China
{weiya_chen,zdxue}@hust.edu.cn, {wanyulin,oushiqi}@isyslab.org

Abstract. Mobile robots play an important role in Ambient Assisted Living (AAL) by supporting or guiding people with reduced mobility to move in an indoor environment. Visual SLAM algorithms have become an important component of such robots by largely reducing the cost of tracking components. These AAL robots represent a typical situation in which robots move on level ground with merely in-plane navigation tasks. In order to find an optimized configuration of monocular SLAM systems in level ground navigation scenarios, we compared different lightweight local descriptors (LDB, BRIEF and ORB) by evaluating their influence on system performance based on the framework of ORB-SLAM. The results indicate that BRIEF outperforms others in metrics like time and trajectory accuracy, while LDB provides best descriptor matching quality. To conclude, BRIEF would be preferred for indoor level ground navigation with a monocular SLAM system, and LDB can be used instead if matching quality is the primary concern.

Keywords: Monocular SLAM · Level ground navigation
Local descriptor · Evaluation

1 Introduction

Simultaneous Localization and Mapping (SLAM) systems have been widely used for autonomous robot exploration both in indoor and outdoor environments. One major application field of SLAM-based mobile robot is Ambient Assisted Living (AAL) [7]: assistive robots are designed to help disabled individuals or people with reduced mobility to move more easily in daily life. In most cases, these robots provide their service on level ground in an indoor environment (from room to room, or inside a building with corridors) in forms of wheelchair [22], smart walker [33] or robot coach [12]. AAL robots often combine inputs from multiple sensors (e.g. LiDAR, sonars and cameras) to achieve more robust localization capability, which leads to complex hardware integration and excessive cost [6].

With the rapid developments of visual SLAM [35], many SLAM systems are able to track and build the map in real-time from purely visual information. This

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11259, pp. 347–358, 2018.
https://doi.org/10.1007/978-3-030-03341-5_29

kind of visual SLAM has been an active research topic for more than twenty years with contributions coming from Robotics, Computer Vision and other related fields.

The emergence of visual SLAM systems like ORB-SLAM [25] makes it possible to build mobile AAL robots with low cost hardware, e.g. a single camera run on an embedded system. Visual SLAM can also help to build 3D map of the environment, which will provide more useful information of the surrounding for tasks like obstacle avoidance than traditional 2D maps. Since AAL robots only involve in-plane navigation on level ground of indoor environments, visual SLAM systems like ORB-SLAM can be further optimized by reducing from 6DoF tracking to 3DoF. For example, ORB-SLAM is based on ORB feature [27], which is a fast alternative of SIFT [19] or SURF [2]. ORB is composed of a rotation-invariant descriptor - rotated-BRIEF, which is useful for 6DoF tracking (e.g. hand-held camera), but not necessary for in-plane navigation.

Aiming to build a monocular SLAM system for AAL robots that usually run on embedded systems, we want to further optimize the state-of-the-art visual SLAM framework by finding appropriate lightweight descriptors that improve real-time tracking performance and reduce computational cost for in-plane navigation. So in this paper, based on the framework of ORB-SLAM - a milestone of feature-point based SLAM system, we compared different lightweight local descriptors by evaluating their influence on system performance in level ground navigation scenarios.

2 Related Work

2.1 Monocular SLAM

Visual SLAM can be performed with a single monocular camera, which is the simplest and cheapest sensor setup among all choices. This simplicity allows monocular SLAM to run on embedded systems or smartphones with minimal hardware integration effort, which encourages many years of research on this topic. Monocular SLAM algorithms have evolved from filtering to keyframe-based bundle adjustment (BA) algorithms, with many implementations lying in the middle ground between them. Filtering methods create a model based on the information gained over all past frames with a probability distribution, every frame is processed by the filter to jointly estimate the map feature locations and the camera pose [8]. Unlike filtering methods, keyframe-based approaches [23] estimate the map using global bundle adjustment for only a small number of past frames, which remain relatively efficient even processing large number of features from the keyframes. The work of Strasdat et al. [29] demonstrated that keyframe bundle adjustment outperforms filtering in term of accuracy per unit of computing time by measuring entropy reduction and tracking error.

The most representative keyframe-based system is marked by PTAM [16], which first introduced the idea of splitting camera tracking and mapping into parallel threads. Various systems are proposed in recent years targeting different issues in the front end and back end such as iSAM [14], FrameSLAM [17], etc.

Another type of methods standing out of framework of filtering and keyframe approaches is called direct SLAM, e.g. LSD-SLAM [9]. Direct SLAM method builds large scale semi-dense maps directly upon optimization over image pixel intensities instead of bundle adjustment over features, which offers more potential for related applications.

However, some intrinsic problems of monocular vision systems, e.g. scale drift and failing with pure rotations, still make monocular SLAM difficult to initialize despite simple hardware setup, which lead to the development of stereo and RGB-D vision systems.

2.2 Keypoint Features

Keypoint features are generally salient points (e.g. corners) encoded by information from local image regions that are invariant to viewpoint and lighting condition changes. Many visual SLAM systems use corner detectors in their tracking pipeline, e.g., a machine learning approach called FAST [26] is often used in real-time applications, and its improved version is integrated in other methods like ORB [27]. Besides corner detectors, another popular local descriptor is the Scale Invariant Feature Transform (SIFT) [19], which first achieves scale-invariant keypoint detection using histograms containing main properties of local appearance. However, the high dimension descriptor of SIFT makes it difficult to be used in real-time situations, which leads to different variants such as the Speeded-Up Robust Features (SURF) [2], PCA-SIFT [15] and other types of lightweight local descriptors.

Lightweight local descriptors are mainly designed to be computation-efficient, so the generating and matching of descriptors can run at frame rate. For example, the BRIEF descriptor [5] directly generates bit strings by simple binary tests in a smoothed image patch, and is augmented with rotation invariance by rotated-BRIEF (ORB). Unlike BRIEF, BRISK [18] and its successor FREAK [1] use a circular sampling pattern to compute intensity comparisons between point pairs. Another descriptor named LDB [34] computes a binary string for an image patch using simple intensity and gradient difference tests on pairwise grid cells, which is demonstrated to achieve greater accuracy and faster speed for tracking tasks than state-of-the-art algorithms.

Lots of evaluations and comparisons of keypoint detectors and descriptors have been done to help us choose among enormous options for a given application. Some surveys compare a special group of algorithms like Juan & Gwun's work [13] on SIFT-related methods, while others include more detectors and descriptors to compare with [20,32]. In the field of visual SLAM, there are also many existing work on the performance comparison of interest point detectors and descriptors [3,10,24]. The common conclusion that we can draw from these surveys is that there is a trade-off between accuracy and computation cost. SIFT and related methods offer better matching performance with high computational cost, while lightweight descriptors provide less precise matching at a much higher speed [21].

The aforementioned evaluations have covered a wide range of detectors and descriptors, but some recent advances like LDB haven't been compared altogether. Moreover, these studies mostly target at general 6DoF tracking scenarios, cases for 3DoF in-plane level ground navigation haven't been addressed yet.

3 Experiment

In order to find an optimized configuration of monocular SLAM systems in level ground navigation scenarios, we compared different lightweight local descriptors by evaluating their influence on system performance based on the framework of ORB-SLAM. The descriptor used in ORB-SLAM is rotated-BRIEF (or rBRIEF), which is BRIEF enhanced with rotation-invariance. Since we only have yaw rotation in level ground scenarios, BRIEF is already sufficient and we expect more efficient tracking with BRIEF as rotation is not considered. Another lightweight, and claimed to be ultra-fast descriptor that we included in the evaluation is LDB [34]. As mentioned in Sect. 2, LDB is an efficient binary descriptor that has the same length as BRIEF (32 bits), and is much shorter than BRISK and SURF (both have 64 bits). Other popular descriptors exceeding a length of 64 bits are excluded from comparison.

So in this experiment, we choose to compare three lightweight descriptors: BRIEF, ORB (rotated-BRIEF) and LDB (without rotation invariance).

3.1 Dataset

Existing Datasets. We first considered existing public visual SLAM datasets for the evaluation task undertaken. The datasets that satisfy our testing requirements should only involve yaw rotation and in-plane translation (3DoF), which excludes most hand-held sequences such as the TUM RGB-D benchmark [30] and NYU Depth dataset [28]. Moreover, we prefer video recordings of indoor environment as AAL robots are mostly designed for indoor service, which again filters out datasets for large-scale outdoor environments, e.g. KITTI dataset [11] for car driving and the EuRoC dataset [4] for aerial vehicle navigation.

Finally we selected two sequences from the TUM RGB-D dataset (we use only the color images) that are designed for testing and debugging purpose - fr1/xyz and fr2/xyz. These two sequences only contain translation movements within a small movement range, which is not strictly "in-plane", but no rotation is involved. The TUM dataset also provides a tool that implements two methods for calculating the error between the estimated trajectory and the real one, namely Absolute Trajectory Error (ATE) and Relative Pose Error (RPE), both are useful for comparison of tracking performance.

Level Ground Sequences. Since we found little existing datasets for level ground indoor navigation, we decided to make some recordings that satisfy the requirements mentioned above. We mounted a monocular camera on a robotic walker - a standard four-wheel (no motor control) assistive walker combined

with different sensors. The user stands behind the walker and walks forward while pushing the walker by holding the handles. A laptop computer running the SLAM algorithm is put on the robotic walker and connected to the camera mounted in front of the walker via a USB cable.

We choose three types of trajectories to be tested, including straight line, zigzag and octagon paths (Fig. 1). These segments have increased complexity and their combination can represent most use case that we encounter for level ground navigation. The length of each segment for these trajectories is chosen arbitrarily according to the room size.

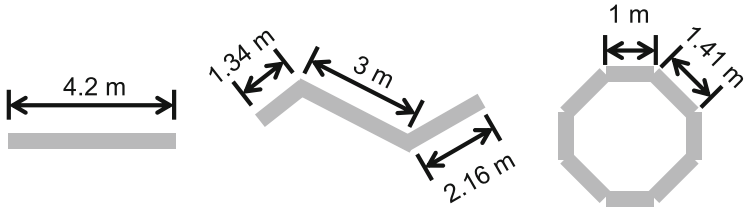

Fig. 1. Trajectories of the level ground video sequences, from left to right: line, zigzag and octagon.

The level ground sequences used in this experiment were captured by a Logitech C525 camera, with the auto-focus function turned off. The intrinsic parameters of the camera are: focal lengths - $f_x = 820.2028$ and $f_y = 819.9700$, the principal point (u,v) = (255.4357, 222.3254), and the radial distortion - $K_1 = 0.0378$ and $K_2 = -0.3324$. The three sequences that we recorded are stored as 640 * 480 images with a frame rate of 30 fps. The line, zigzag and octagon sequences last respectively 33, 54 and 110 s, and are saved as 803, 1291 and 2647 images.

3.2 Performance Metrics

Time and accuracy are two fundamental aspects that represent the real-time responsiveness and quality of a SLAM system. The performance metrics that we use to evaluate the influence of different descriptors are thus divided into the following groups:

Time: We logged time used for descriptor generation and matching since they directly reflect a descriptor's time efficiency. We also want to see the impact of changing keypoint descriptor on system performance, so we measured the execution time of the whole SLAM process along with the time for different states - initialization, tracking and relocalization. A good SLAM system should spend less time to initialize and relocate, leaving more time for tracking.

Matching Accuracy: Keypoint matching between frames is used to recover the camera's change of pose. We counted the number of matched keypoints as

more correct matches generally lead to more accurate recovered pose. When regarding descriptor matching as a classification problem and each keypoint to be an individual class, we can use J3 (Eq. 1) to quantify class separability which is based on within and between class scatter matrix: S_w and S_b (Eq. 2) [31].

$$J_3 = trace\{S_w^{-1} S_m\} \tag{1}$$

$$S_m = S_w + S_b = \sum_{i=1}^{M} p_i s_i + \sum_{i=1}^{M} p_i (\mu_i - \mu_0)(\mu_i - \mu_0)^T \tag{2}$$

where S_m is the global covariance matrix. To compute S_w, p_i and s_i are the probability and covariance matrix of class i. For S_b, μ_i is the average feature vector for class i and μ_0 is the average vector for all classes. Higher J3 value computed from all the binary strings of a descriptor indicates better matching capability. To compute J3, we selected 50 images at the end of each sequence and collected all descriptor binaries for keypoints extracted from the very first image. Finally, only keypoints that have more than 30 binary strings for all three descriptors are included.

Tracking Accuracy: Since we use part of the TUM RGB-D dataset, we can make use of some useful tools provided by the authors. Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) are two methods well-suited for measuring the performance of visual SLAM systems when ground-truth trajectory is available. In this experiment, our level ground recordings don't have ground-truth data that are compatible with TUM tools, so these two methods are only applied to TUM dataset.

4 Results

We performed our tests on a laptop computer with Intel (R) Core (TM) i7-5700HQ CPU @ 2.70 GHz with 8G RAM, running Ubuntu 16.04 LTS. For each video sequence, we run the SLAM system under each testing condition for 10 times to see their averaged performance. Hereafter we name each testing condition by the name of the descriptor in use, i.e. LDB, BRIEF and ORB condition.

4.1 Time

Figure 2 shows the time performance for the whole video sequences under each condition for two descriptor-related tasks: descriptor generation and matching. The results show that, LDB is slightly quicker for keypoint matching, but takes more time to generate the binary code than two other methods, and the result is almost consistent across different video sequences.

In addition to absolute time duration for a task, we also computed the proportion of that task in the total time of the whole SLAM process since the total time differs under each condition. On average, LDB has the highest time rate

for descriptor generation (57.9%) and lowest time rate for keypoint matching (5.9%). Regarding ORB, it has the highest matching time cost rate (7.6%), but has similar performance in descriptor generation (52.3%) with BRIEF (51.5%).

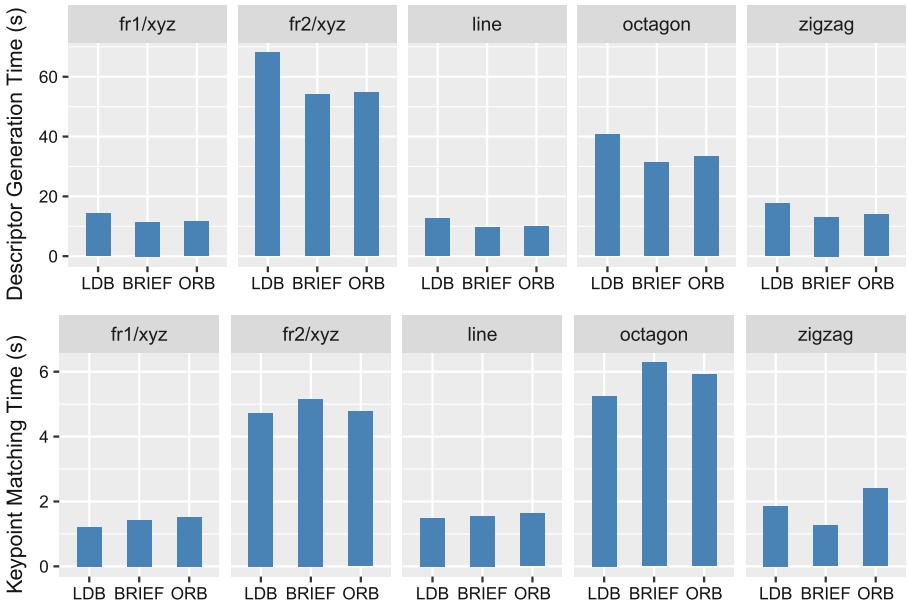

Fig. 2. Descriptor-related time performance for different video sequences (in seconds)

As mentioned in previous section, we collected the execution time for the whole process as well as for each system state. As shown in Fig. 3, all conditions have good performance in fr2/xyz with most time spent on tracking (from 98.8% to 99.3%), while with other sequences all conditions take more time to initialize, among which ORB suffers a steeper increase (up to 12.6%).

Both zigzag and octagon sequences include yaw rotations, relocalization occurred under all conditions on these two sequences. In the zigzag sequence, initialization remains acceptable for LDB and BRIEF (6.9% and 7.4%), whereas ORB increases rapidly (41.3%). LDB spends most time for tracking (85.7%) and least for relocalization (7.4%), BRIEF (52.9%) has similar tracking time as ORB (44.2%), but much more time for relocalization (39.7%). Octagon sequence contains multiple in-place rotations with relatively short transition, as a consequence, all conditions have bad performance. The best condition in this case - BRIEF is able to run tracking for half of the total time (54.8%), while the others have to relocate from time to time.

If we take the sum for all video sequences, ORB condition spends more time for initialization than LDB and BRIEF, while BRIEF condition outperforms others in tracking, relocalization and total time with a slight advantage.

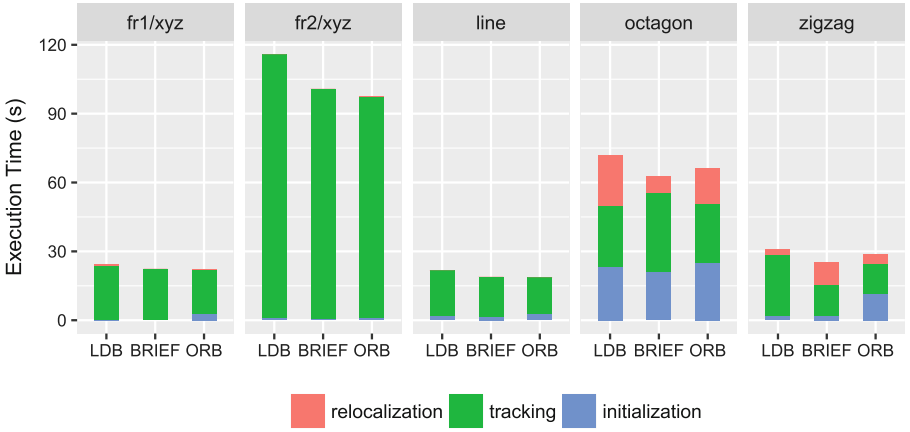

Fig. 3. Total and state-wise execution time with different video sequences (in seconds).

Table 1. Average number of matched keypoints per frame and J3 score for each condition.

Sequence	Matched keypoint number			J3 score		
	LDB	BRIEF	ORB	LDB	BRIEF	ORB
fr1/xyz	261	**295**	198	**98.67**	89.46	56.39
fr2/xyz	201	**219**	169	**88.63**	84.71	58.75
line	276	**280**	237	**102.36**	91.82	57.85
zigzag	225	**244**	204	**84.05**	72.27	61.99
octagon	210	**214**	189	**97.79**	95.02	60.82
mean	235	**250**	199	**94.30**	86.66	59.16
sd	32.4	36.4	25.0	7.64	8.88	2.25

4.2 Matching Accuracy

Table 1 shows the average number of matched keypoints per frame and J3 score for the whole sequences. We can see that for the number of matched keypoints, LDB (mean = 235) has similar performance with BRIEF (mean = 250), while ORB (mean = 199) has much lower number than both of them. Regarding J3 score, from the frames we choose (all three conditions run tracking during this period), we find that LDB has the highest score in all sequences (mean = 94.30), and the mean score of ORB (mean = 59.16) is far lower than the other two.

4.3 Tracking Accuracy

We use ATE and RPE to compute the trajectory error of fr1/xyz and fr2/xyz. As shown in Table 2 and Fig. 4, all conditions have similar performance for ATE ($0.3625 \sim 0.3666$) in sequence fr2/xyz, however, BRIEF gets much smaller error

than the other two in fr1/xyz with an error of 0.057 m. For RPE, we sum translation and rotation error separately. Same as ATE, the performance for all conditions are close in fr2/xyz, but BRIEF still outperforms the others in fr1/xyz.

Table 2. Measurement of tracking accuracy for each condition (in meter and degree).

Sequence		ATE			RPE-T error			RPE-R error		
		LDB	BRIEF	ORB	LDB	BRIEF	ORB	LDB	BRIEF	ORB
fr1/xyz	mean	0.1039	**0.0570**	0.1009	0.1251	**0.0634**	0.0922	5.2664	**2.5152**	5.7203
	sd	0.0496	0.0500	0.0800	0.0490	0.0539	0.0863	3.7171	3.2312	4.6535
fr2/xyz	mean	0.3666	0.3625	0.3651	0.0637	0.0635	0.0635	2.1105	1.7682	2.0970
	sd	0.0014	0.0042	0.0022	0.0002	0.0007	0.0004	0.7405	0.0461	0.6862

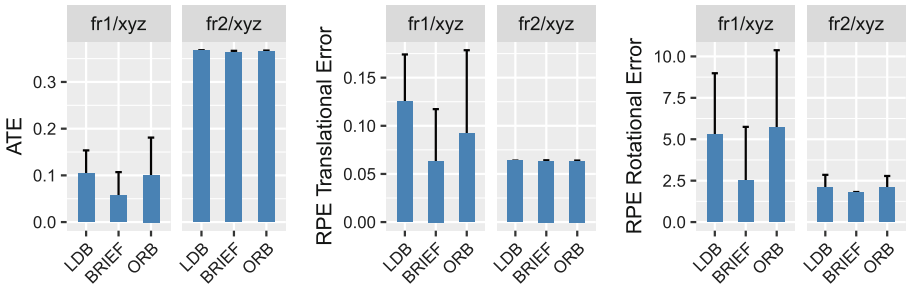

Fig. 4. Measurement of tracking accuracy for each video sequence.

5 Discussion

From the above results, we can see that descriptor generation is still the most time-consuming task for local feature based SLAM system that takes more than half of the total system running time. The use of BRIEF provides faster binary code generation that allows more time for tracking and less total time than with the other two descriptors.

The ORB descriptor is indeed rotated-BRIEF, with additional rotation-invariance ability compared to BRIEF, however, according to our tests, this augmentation largely reduced the number of matched keypoints per frame, which hinders not only the system time efficiency, but also matching ability (as more matched keypoints lead to better tracking result). This reduction is mainly due to the additional angular constraints during keypoint matching. Since rotation invariance is not required in level ground navigation, ORB descriptor is not recommended for this type of application.

Through all the tests with various sequences, we find the performance of different descriptors tends to diverge as the camera motion becomes more complicated (from line to octagon), and remains at the same level with very smooth and slow motion (e.g. in fr2/xyz). Globally, BRIEF retains robust tracking performance in difficult situations, although more sequences should be included to further confirm this observation on trajectory estimation quality.

In fact, when running pilot test for our robotic walker with ORB-SLAM, we found that the system struggled to initialize in indoor environment with many white walls around. The keypoints that the system can extract at runtime are too few to support functional tracking. We had to paste some texture-rich pictures on the walls to facilitate keypoints extraction. On the contrary, if tests were taken in an outdoor environment, the number of keypoints should no longer be a problem. In this case, LDB would be an appropriate choice since it has highest J3 score among our tested descriptors.

6 Conclusion

In this work, we conducted an experiment to test the influence of different lightweight local descriptors on the performance of monocular SLAM system, in aim to find the best choice among LDB, BRIEF and ORB for level ground indoor navigation. The results indicate that BRIEF outperforms the others both in terms of time and trajectory accuracy, though it provides slightly lower matching quality than LDB. To conclude, BRIEF would be a preferred component of monocular SLAM systems designed for indoor level ground navigation.

In the future, with advances from the computer vision community, more lightweight descriptors can be included for comparison and we can also evaluate the impact of keypoint extraction methods. To further improve the usability of SLAM systems for robotic walker as monocular SLAM systems are sometimes delicate to initialize, we can take stereo, RGB-D and even inertial sensors into consideration.

Acknowledgement. This work was funded by the Chinese Universities Scientific Fund (2017KFYXJJ225) and Science and Technology Program of Guangzhou (201803010067).

References

1. Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: fast retina keypoint. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510–517. IEEE (2012)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_32
3. Bayraktar, E., Boyraz, P.: Analysis of feature detector and descriptor combinations with a localization experiment for various performance metrics. arXiv preprint arXiv:1710.06232 (2017)

4. Burri, M., et al.: The EuRoC micro aerial vehicle datasets. Int. J. Robot. Res. (2016). https://doi.org/10.1177/0278364915620033
5. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_56
6. Cavanini, L., Benetazzo, F., Freddi, A., Longhi, S., Monteriu, A.: Slam-based autonomous wheelchair navigation system for AAL scenarios. In: 2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA), pp. 1–5. IEEE (2014)
7. Costa, R., et al.: Ambient assisted living. In: Corchado, J.M., Tapia, D.I., Bravo, J., et al. (eds.) 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008, pp. 86–94. Springer, Heidelberg (2008)
8. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: real-time single camera slam. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)
9. 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, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
10. Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of interest point detectors and feature descriptors for visual tracking. Int. J. Comput. Vis. **94**(3), 335 (2011)
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
12. Gross, H.M., et al.: Roreas: robot coach for walking and orientation training in clinical post-stroke rehabilitation-prototype implementation and evaluation in field trials. Auton. Robot. **41**(3), 679–698 (2017)
13. Juan, L., Gwun, O.: A comparison of sift, pca-sift and surf. Int. J. Image Process. (IJIP) **3**(4), 143–152 (2009)
14. Kaess, M., Ranganathan, A., Dellaert, F.: iSAM: incremental smoothing and mapping. IEEE Trans. Robot. **24**(6), 1365–1378 (2008)
15. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 2, p. II. IEEE (2004)
16. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007, pp. 225–234. IEEE (2007)
17. Konolige, K., Agrawal, M.: FrameSLAM: from bundle adjustment to real-time visual mapping. IEEE Trans. Robot. **24**(5), 1066–1077 (2008)
18. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: binary robust invariant scalable keypoints. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2548–2555. IEEE (2011)
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
20. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
21. Miksik, O., Mikolajczyk, K.: Evaluation of local detectors and descriptors for fast feature matching. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 2681–2684. IEEE (2012)

22. Morales, Y., Kallakuri, N., Shinozawa, K., Miyashita, T., Hagita, N.: Human-comfortable navigation for an autonomous robotic wheelchair. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2737–2743. IEEE (2013)
23. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Real time localization and 3d reconstruction. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 363–370. IEEE (2006)
24. Mozos, Ó.M., Gil, A., Ballesta, M., Reinoso, O.: Interest point detectors for visual SLAM. In: Borrajo, D., Castillo, L., Corchado, J.M. (eds.) CAEPIA 2007. LNCS (LNAI), vol. 4788, pp. 170–179. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75271-4_18
25. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015)
26. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_34
27. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571. IEEE (2011)
28. Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition (2011)
29. Strasdat, H., Montiel, J.M., Davison, A.J.: Visual SLAM: why filter? Image Vis. Comput. **30**(2), 65–77 (2012)
30. 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 (IROS), October 2012
31. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic Press, Boston (2009)
32. Tuytelaars, T., Mikolajczyk, K., et al.: Local invariant feature detectors: a survey. Found. trends® Comput. Graph. Vis. **3**(3), 177–280 (2008)
33. Wachaja, A., Agarwal, P., Zink, M., Adame, M.R., Möller, K., Burgard, W.: Navigating blind people with a smart walker. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6014–6019. IEEE (2015)
34. Yang, X., Cheng, K.T.: Local difference binary for ultrafast and distinctive feature description. IEEE Trans. Pattern Anal. Mach. Intell. **36**(1), 188–194 (2014)
35. Yousif, K., Bab-Hadiashar, A., Hoseinnezhad, R.: An overview to visual odometry and visual SLAM: applications to mobile robotics. Intell. Ind. Syst. **1**(4), 289–311 (2015)