FAST RELOCALIZATION FOR VISUAL ODOMETRY USING BINARY FEATURES

J. Straub [1], *S. Hilsenbeck* [2], *G. Schroth* [2], *R. Huitl* [2], *A. Möller* [2], *E. Steinbach* [2]

[1] CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA
[2] Institute for Media Technology, Technische Universität München, Munich, Germany

jstraub@csail.mit.edu, s.hilsenbeck@tum.de

ABSTRACT

State-of-the-art visual odometry algorithms achieve remarkable efficiency and accuracy. Under realistic conditions, however, tracking failures are inevitable and to continue tracking, a recovery strategy is required. In this paper, we propose a relocalization system that enables realtime, 6D pose recovery for wide baselines. Our approach targets specifically resource-constrained hardware such as mobile phones. By exploiting the properties of low-complexity binary feature descriptors, nearest-neighbor search is performed efficiently using Locality Sensitive Hashing. Our method does not require time-consuming offline training of hash tables and it can be applied to any visual odometry system. We provide a thorough evaluation of effectiveness, robustness and runtime on an indoor test sequence with available ground truth poses. We investigate the system parameterization and compare the relocalization performance for the three binary descriptors BRIEF, unscaled BRIEF and ORB. In contrast to previous work on mobile visual odometry, we are able to quickly recover from tracking failures within maps with thousands of 3D feature points.

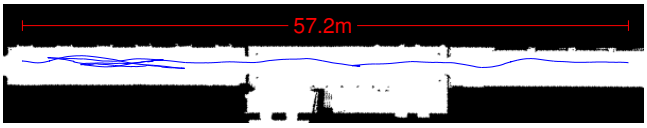
Index Terms— Relocalization, BRIEF, ORB, Locality Sensitive Hashing, Visual Odometry

1. INTRODUCTION

With today's powerful smartphones, image-based localization is a promising approach for indoor positioning without the need for costly hardware installations [1]. To achieve this goal, visual odometry approaches are required, which allow the tracking of a user's path through a building [2]. For feature-rich, static environments and for smooth camera motion, state-of-the-art visual odometry systems achieve remarkable efficiency and accuracy. In real-world applications, however, it is inevitable that these ideal conditions will be violated at some point, causing visual tracking systems to fail (i.e. observations in the current frame cannot be associated with previous ones).

People walking by and occluding the field of view, motion blur during a quick turn, or simply a blank wall with very few salient points are typical causes that may interrupt visual odometry, sometimes for several seconds. To ensure continuous operation over longer periods of time, a complementary relocalization algorithm is required which recovers the current pose and thus allows the visual odometry system to resume tracking.

So far, most approaches that provide real-time relocalization on mobile devices are designed for small (e.g. augmented reality) workspaces, where previous observations can usually be found in a close vicinity to the current pose [3]. This assumption, however, does not hold in the case of mobile device odometry, where the user movement is typically not constrained to a fixed area.



(a) Ground truth path

(b) Start segment (c) Middle segment (d) End segment

Fig. 1: Data and ground truth path used for evaluation. Images are collected using a Samsung Galaxy Tab 10.1 mounted on a trolley. For accurate ground truth data, the trolley is equipped with laser scanners and wheel odometry. The dataset starts in the left segment of the corridor, which is traversed three times. Passing a staircase in the middle segment, the path ends on the right.

The major contribution of this paper is the design of a relocalization system that enables realtime 6D pose recovery even for wide baselines (long periods of tracking failure) on mobile devices. We describe a particularly efficient Locality Sensitive Hashing (LSH) [4] approach based on recently proposed low-complexity binary feature descriptors [5, 6, 7, 8]. This enables fast approximate nearest neighbor search to identify correspondences to previously mapped 3D points. In addition, we restrict the search space according to adaptive visibility constraints which allows us to cope with extensive maps at constant complexity.

For the evaluation of our relocalization system, we use the Parallel Tracking and Mapping algorithm (PTAM) [9] as baseline visual odometry algorithm. Our relocalization method is, however, general and can be integrated into any algorithm that estimates 3D positions of features.

2. RELATED WORK

A comparison of several relocalization techniques for visual SLAM is provided by Williams et al. [10] where three basic types of approaches are distinguished: map-to-map (matching two 3D point clouds), image-to-map (matching observations in an image to a 3D point cloud), and image-to-image (matching observations in one image to observations in another image).

In the image-to-image category, a common approach for visual location recognition is the Bag of Words model [11] as, for example, employed by Eade et al. [12] in conjunction with SIFT [13] features and Cummins et al. [14] using SURF [15] features. A third method

in this vein proposed by Klein and Murray [3] is the relocalization algorithm originally used in the PTAM implementation. Based on down-sampled versions of keyframes (collected about every 50 to 80 frames), the one that is most similar to the current image is identified by computing the sum of squared differences.

While all three methods are extremely fast (1.5 ms for pose estimation within 250 keyframes in the case of Klein and Murray), it is only possible to localize to previously visited places where keyframes were added to the map. These conditions are valid, for instance, in the small desktop workspace scenario that PTAM was originally developed for. In the case of mobile device odometry, however, this is a clear limitation since the user is expected to walk into previously unseen areas where no associated keyframes exist.

Map-to-map techniques, on the other hand, involve matching of large unstructured point clouds in 3D space, a process of significant complexity. Not only are approaches of this kind usually too slow, but they also lack accuracy unless the data were collected using, e.g., high precision laser scanners. Consequently, in [16] Williams et al. come to the conclusion that, regarding visual SLAM, image-to-map techniques work best because they provide the accuracy as well as the speed required for restarting visual tracking. They propose a system that employs randomized trees for classification to identify correspondences between image features and map points. While being very fast, the approach is designed for maps rarely exceeding 150 features. It cannot handle maps of several thousand (about 15k) features which are typical in visual odometry applications.

The approach conceptually most similar to ours is proposed by Arth et al. [17] who present a feature-based localization system for smartphones. Based on a modified SURF descriptor, image-to-map matching is performed by exhaustive nearest neighbor search. The 3-point pose algorithm and RANSAC are used to obtain a pose estimate which is further refined using robust Gauss-Newton optimization. In order to reduce the search complexity, feature matching has to be restricted to so called *potentially visible sets* (PVSs). To identify PVSs, the 3D point cloud of the environment of interest needs to be collected in advance and discretized into view cells by precomputing the cell-to-cell visibility. The assumption of map data that are available a priori does not hold for our application. Further, as the continuous extraction of SURF features requires approximately 700 ms per keyframe on a state-of-the-art mobile device, the computational load is prohibitive for our scenario.

With the emergence of fast-to-extract binary descriptors, however, feature-based relocalization becomes feasible on mobile devices if combined with an efficient nearest neighbor search. For instance, the BRIEF descriptor proposed by Calonder et al. [5] performs a fixed number of intensity comparisons to compute the descriptor. The predefined locations of these intensity tests are chosen to minimize correlation. For increased robustness, the patch is initially blurred using a Gaussian kernel. As a result, the extraction of BRIEF descriptors is about 40 times faster than for SURF while generating only half the memory footprint. The binary descriptor can be matched using the Hamming distance. In addition, the bitwise independence naturally enables approximate nearest neighbor search using Locality Sensitive Hashing with almost no overhead compared to other methods. Besides BRIEF, our approach can also be based on more advanced binary descriptors such as ORB [6], BRISK [7] or FREAK [8].

3. PARALLEL TRACKING AND MAPPING

Parallel Tracking and Mapping (PTAM) [9] is an efficient algorithm to perform monocular SLAM. In this paper, it is used as the base system to evaluate our relocalization method. PTAM splits up the tasks of camera pose tracking and map building into two threads: the tracking and the mapping thread.

The mapping thread estimates the 3D structure of the environment using the information about 2D-3D point correspondences supplied by the tracking thread. The map is represented as a point cloud of 3D points, where each 3D point is associated with its observations in the set of keyframes. The tracking thread in turn uses the knowledge of 3D feature positions to find 2D-3D point correspondences which then are used to update the camera pose. This separation allows for a continuous refinement of the map estimate as computation time is available.

4. RELOCALIZATION BASED ON BINARY FEATURES

Our relocalization method robustly estimates a 6DOF pose from 2D image to 3D world coordinate correspondences. This association is established via approximate nearest neighbor search in the feature Hamming space. During normal tracking of PTAM, the respective binary feature descriptors for each map point are extracted whenever a new keyframe is inserted into the map. Conventional descriptors such as SIFT and SURF possess larger invariance against perspective transformations than binary descriptors. Therefore, they are well suited for matching images along wider baselines. This advantage, however, comes at the cost of greatly increased amounts of time spent for descriptor extraction. Moreover, the baselines to cope with can be significantly shortened by exploiting the fact that a visual tracking system continuously provides updated observations of known landmarks. Hence, for each map point, only the most recent descriptor is retained.

In the event of a tracking failure, our relocalization method obtains an estimate of the camera pose within PTAM's map as follows:

- First, binary descriptors are extracted at the positions of FAST [18] corners in the current image.
- Next, binary descriptors in the current image are matched to descriptors (associated with 3D points) stored in the map. This search is performed rapidly using Locality Sensitive Hashing.
- Then, the camera pose is robustly estimated from the 2D-3D point correspondences using the 3-point pose method [19] and Progressive Sample Consensus (PROSAC) [20].
- Finally, this pose estimate is refined using Levenberg Marquardt optimization (M-Estimator).

This method is repeated for each incoming frame until the pose estimate allows PTAM to continue tracking.

To reduce the search space, the map points used for matching are filtered according to their visibility. We select all keyframes that lie within a certain radius around the last known position of the camera. Based on typical walking speeds, we found that a distance of 5 m is sufficient, although this choice can be adapted according to the time elapsed since the tracking failure. By accumulating the observations stored with those keyframes, we can derive a good estimate of the set of features that are likely to be visible from the current position.

4.1. Locality Sensitive Hashing for Binary Features

Locality Sensitive Hashing (LSH) [4] is a hashing-based technique to perform approximate nearest neighbor search in high dimensions. The algorithm employs a hash function that generates the same hash code for close-by feature descriptors with high probability. Shahbazi et al. [21] propose to use Locality Sensitive Hashing for large scale feature matching. They extract SIFT descriptors, which have to be projected onto random vectors to obtain hash values. In contrast to that, we exploit the inherent independence among bits in the binary

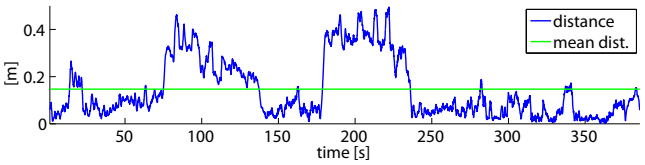


Fig. 2: Distance between PTAM's trajectory and ground truth.

feature vector to perform rapid hash value generation without having to perform a costly projection.

To be more precise, the hash function generates the same hash codes for two binary descriptors with a higher probability if their Hamming distance is less than some r than if their Hamming distance is greater than r. In order to increase the probability of finding the true nearest neighbor, N_T hash tables are used in parallel. We choose the hash function g_i to compose the hash code from N_B randomly selected bits in the binary descriptor bit strings $\mathrm{BD}_j = [\mathrm{bit}_1, \mathrm{bit}_2, \ldots, \mathrm{bit}_M]$:

$$g_i(\mathrm{BD}_j) = \left[\mathrm{bit}_{r_1}, \mathrm{bit}_{r_2}, \ldots, \mathrm{bit}_{r_{N_B}}\right],$$

where the indices r_i are drawn uniformly and without repetition from $\{1, \ldots, M\}$ once at instantiation of the hash function.

In combination with LSH, feature matching can be performed about 23 times faster than exhaustive search. Hash tables for binary descriptors can be built very efficiently in batch or incrementally as, with randomly selected hash functions, no time consuming training is required. This is an important property to restrict the search space according to visibility constraints at very little overhead.

5. RESULTS

Running PTAM as the baseline visual odometry system[1], we extract BRIEF, usBRIEF, and ORB binary features (each 256 bits) whenever a keyframe is inserted. We use the respective OpenCV implementations. By usBRIEF we denote the unscaled BRIEF descriptor, i.e., the descriptors are all extracted at the original image resolution. In contrast to that, BRIEF and ORB descriptors are extracted at the image scale level of the respective FAST keypoint which distinguishes four resolution layers (each down-scaled by a factor of 2). For all experiments, we run PROSAC with maximally 100 iterations.

5.1. Evaluation Dataset

For evaluation of the accuracy of PTAM and the relocalization mechanisms, a tablet[2] was mounted on a trolley equipped with laser scanners and wheel odometry. Running a particle filter localization algorithm, it provides ground truth with an average accuracy of 2 cm [22]. Video and poses of the trolley are synchronized by hand. As can be seen in Fig. 1, the dataset was recorded in a well lit but sparsely textured corridor. The corridor's left side shows a series of windows, whereas the right side is blank except for several display cases. In the middle part, the hallway opens up for a large staircase.

Fig. 1a shows the ground truth path. The run starts in the left segment of the hallway which is traversed three times before entering the middle segment. The dataset has a length of seven minutes and the trajectory is about 100 m long. All computation results and timings were collected on an Intel i5 laptop. In its original form, this dataset does not cause any tracking failures. Fig. 2 depicts the deviation of PTAM's undisturbed trajectory estimate from the ground truth which, on average, stays below 20 cm. Evidently, PTAM leads to estimation errors of up to 40 cm for this dataset.

[1]Source code available from `www.robots.ox.ac.uk/~gk/PTAM/`

[2]Samsung Galaxy Tab 10.1

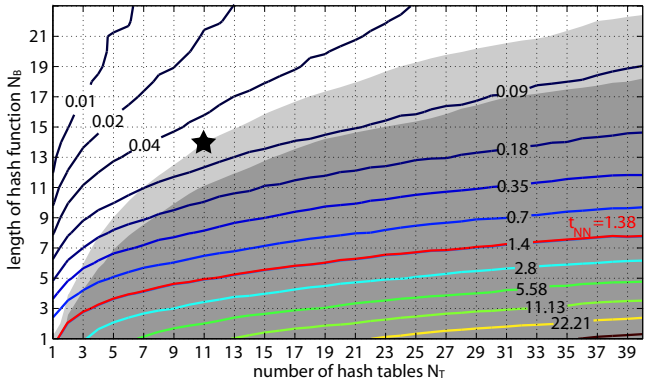


Fig. 3: Timing in ms of a single feature LSH query on a database of 15k binary features. The gray areas depict the regions of more than 90% (light gray) and more than 95% (darker gray) chance of finding the true nearest neighbor. For comparison, the red line depicts the time required for exhaustive exact nearest neighbor search. The solid star denotes the parameter choice for our experiments.

5.2. LSH Parameter Selection

In order to find a good set of parameters for LSH, we evaluate the query time and the probability of finding the exact nearest neighbor on a database of 15k binary features. This experiment was conducted offline using the database of features collected during a complete run of PTAM on our dataset described in the previous section. In our experiments, the average database comprises 15k binary features, after having filtered for features that are visible in a radius of approximately 5 m around the current location.

The results of this experiment are shown in Fig. 3. For the following evaluations, we use 11 hash tables and 14 hash bits (indicated by a solid star in Fig. 3). This configuration gives a 23 times speedup over exhaustive nearest neighbor search at a 90% probability of finding the true nearest neighbor. Additionally, it uses only a small number of hash tables, which is important for deployment on a mobile device where memory is scarce.

5.3. Relocalization Timings

As described in Sec. 4, the relocalization algorithm first extracts descriptors from the current frame, finds approximate nearest neighbors of those in the map, and finally robustly estimates the camera pose from the correspondences.

Fig. 4 shows a breakdown of the time required for relocalization using either BRIEF, usBRIEF, or ORB. Notice that the LSH preparation, i.e., selecting binary feature descriptors from the current map and sorting them into hash tables is performed as a batch process. This can also be done incrementally while descriptors are extracted from keyframes. Since we expect tracking failures to occur infrequently, in-batch preparation of hash tables is more efficient.

Feature-based relocalization requires the extraction of features whenever PTAM adds observations. Table 1 lists the timings for descriptor computation as measured on our system. As can be seen, this adds an overhead at each keyframe. However, due to the use of binary features, this additional time spent for computation remains with 3 to 6 ms very limited compared to several hundred milliseconds for SURF descriptors.

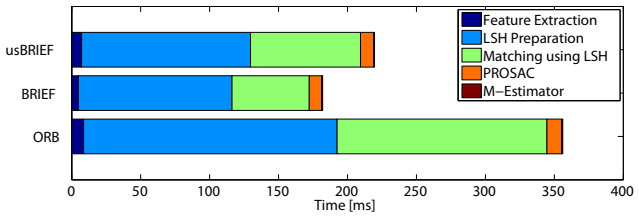


Fig. 4: Relocalization timings for the three binary descriptors split into the different parts of the recovery procedure. These steps are necessary only after a tracking loss. ORB extracts around 50% more features than the BRIEF variants, increasing the time spent for LSH.

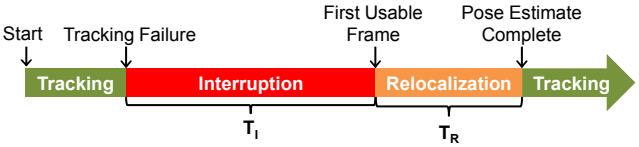


Fig. 5: After a tracking failure, two phases are important to distinguish: The duration of the tracking interruption, T_I, and, once the first usable frame arrives, the time spent to estimate the current pose, T_R. While T_I is for the most part determined by external influences, T_R is mainly defined by the relocalization procedure (see Fig. 4).

5.4. Relocalization Precision and Robustness

As depicted in Fig. 5, we need to distinguish two phases that follow a tracking failure: The first is the duration of the tracking interruption, T_I. The second is the time span, T_R, that is spent for relocalization upon arrival of the first usable frame. T_R ends with the completion of a pose estimate that is accurate enough to restart tracking.

Next, we evaluate the robustness of our binary-feature-based relocalization algorithm with respect to increasing T_I. We artificially introduce a tracking failure at approximately every 80 frames in the dataset (over a series of experiments). At each position, tracking interruptions of increasing durations, T_I, are simulated, after which we record whether our recovery approach allows us to resume tracking.

Fig. 6 shows the fraction of successful relocalizations for the three different binary descriptors as a function of the distance between the point of tracking loss and the point of relocalization. Regarding precision, the poses of successful relocalization attempts are on average 40 cm away from ground truth with maximum distances of up to 1.5 m. For short interruptions of tracking, the version of BRIEF that is not scale invariant (usBRIEF) gives almost 100% of successful recoveries. As the duration of tracking failure increases, however, the performance drops quickly. ORB and BRIEF, in contrast, remain well above 90% even for considerable distances. Due to its additional rotation invariance, ORB outperforms the other two descriptors for long periods of tracking loss. The superior robustness of ORB comes with the slight drawback of an extraction time about twice as long as for BRIEF (see Table 1).

As the relocalization process requires a certain amount of time (T_R), the returned pose estimate is deviating from the actual pose in the case of camera motion. Thus, in a final experiment, we determine the maximum deviation for which tracking is able to resume. We interrupt tracking for a fixed duration, $T_I = 10\ s$, and subsequently let the recovery procedure compute a pose estimate. But instead of feeding it immediately to PTAM, we introduce an additional delay before the location estimate is forwarded and record whether PTAM is able to resume tracking. With increasing delay (over multiple runs of the experiment), the deviation between the pose estimate of our relocalization algorithm and the actual pose grows, as the camera continues along its trajectory (including translational as well as rotational motion).

For this evaluation, we pick two different spots in the trajectory: the first within an already explored area, and the second in the middle of a long stretch of pure forward motion into unexplored parts of the corridor. We find that PTAM can restart tracking despite a deviation of up to 1.8 m in the easier scenario in already explored space and up to 0.8 m in the difficult scenario of purely forward motion. As mentioned above, our recovery procedure requires 180 to 350 ms to compute a pose estimate. Thus, for tracking to be able to resume, the camera motion may not exceed 5.14 m/s in the first scenario and 2.28 m/s in the more difficult second scenario. Assuming a typical walking speed of 1.5 m/s, the proposed relocalization system can cope with realistic scenarios.

	avg. extraction time	avg. number of features
usBRIEF	3.07 ms	222
BRIEF	2.92 ms	216
ORB	5.43 ms	348

Table 1: Extraction time for the three different binary feature descriptors which are extracted at each keyframe. The timings include feature detection using FAST on four scale octaves (0.9 ms).

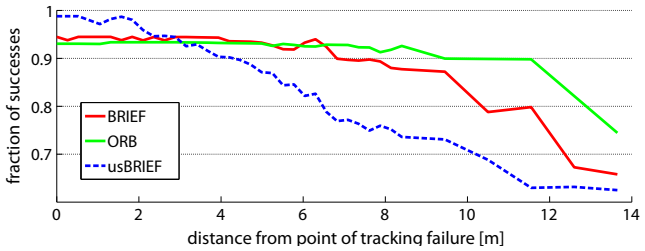


Fig. 6: Fractions of successful relocalization attempts over increasing durations of tracking failure, T_I. While the tracking loss persists, the camera moves away from the last known position. Results for three different binary feature descriptors. The average distance from the true position among successful relocalizations is 40 cm.

6. CONCLUSION

From our results we conclude that binary features combined with LSH for approximate nearest neighbor search yield a fast relocalization technique, which allows visual odometry systems to reliably recover after prolonged periods of tracking failure. In comparison to the widely used SURF features, binary features are up to 40 times faster to extract and Hamming distance computations can be performed very efficiently. Additionally, they require only half the memory of standard SURF. In combination with LSH, feature matching is performed about 23 times faster than exhaustive search. Here, we exploit the structure of binary features which is particularly suited for hashing. Restricting the search space according to adaptive visibility constraints allows us to cope with extensive maps at constant complexity. The algorithm may be employed in any camera tracking system that provides 3D positions for feature descriptors.

7. ACKNOWLEDGMENT

This research project has been supported by the space agency of the German Aerospace Center with funds from the Federal Ministry of Economics and Technology on the basis of a resolution of the German Bundestag under the references 50NA1107 and 50NA1307.

8. REFERENCES

[1] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach, "Mobile visual location recognition," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 77–89, 2011.

[2] S. Hilsenbeck, A. Möller, R. Huitl, G. Schroth, M. Kranz, and E. Steinbach, "Scale-preserving long-term visual odometry for indoor navigation," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sydney, Australia, Nov. 2012.

[3] G. Klein and D. Murray, "Improving the agility of keyframe-based SLAM," *European Conference on Computer Vision (ECCV)*, pp. 802–815, 2008.

[4] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *International Conference on Very Large Data Bases (VLDB)*, Edinburgh, Scotland, UK, 1999, pp. 518–529.

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," *European Conference on Computer Vision (ECCV)*, pp. 778–792, 2010.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011, IEEE, pp. 2564–2571.

[7] S. Leutenegger, M. Chli, and R.Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011, IEEE, pp. 2548–2555.

[8] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, 2012, IEEE.

[9] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan, 2007, IEEE Computer Society, pp. 1–10.

[10] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular slam," *Robotics and Autonomous Systems (RAS)*, vol. 57, no. 12, pp. 1188–1197, 2009.

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision (ICCV)*, Nice, France, 2003, IEEE, pp. 1470–1477.

[12] E. Eade and T.W. Drummond, "Unified loop closing and recovery for realtime monocular slam," in *British Conference on Machine Vision (BMVC)*, Leeds, UK, 2008, BMVA.

[13] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[14] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *International Journal of Robotics Research (IJRR)*, vol. 27, no. 6, pp. 647–665, 2008.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[16] B. Williams, G. Klein, and I. Reid, "Real-time SLAM relocalisation," in *IEEE International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, 2007, IEEE, pp. 1–8.

[17] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg, "Wide area localization on mobile phones," in *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Orlando, FL, USA, 2009, IEEE, pp. 73–82.

[18] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *European Conference on Computer Vision (ECCV)*, pp. 430–443, 2006.

[19] M.A. Fischler and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[20] O. Chum and J. Matas, "Matching with prosac-progressive sample consensus," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, 2005, IEEE, vol. 1, pp. 220–226.

[21] H. Shahbazi and H. Zhang, "Application of locality sensitive hashing to realtime loop closure detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 1228–1233.

[22] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, "TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping," in *IEEE International Conference on Image Processing (ICIP)*, Orlando, FL, USA, Sept. 2012, IEEE, Dataset available at `http://navvis.de/dataset`.