2013 European Conference on Mobile Robots (ECMR)
Barcelona, Spain, September 25-27, 2013

A Comparison of Feature Descriptors for Visual SLAM

Jan Hartmann, Jan Helge Klüssendorff, and Erik Maehle

Abstract— **Feature detection and feature description plays an important part in Visual Simultaneous Localization and Mapping (VSLAM). Visual features are commonly used to efficiently estimate the motion of the camera (visual odometry) and link the current image to previously visited parts of the environment (place recognition, loop closure). Gradient histogram-based feature descriptors, like SIFT and SURF, are frequently used for this task. Recently introduced binary descriptors, as BRIEF or BRISK, claim to offer similar capabilities at lower computational cost. In this paper, we will compare the most popular feature descriptors in a typical graph-based VSLAM algorithm using two publicly available datasets to determine the impact of the choice for feature descriptor in terms of accuracy and speed in a realistic scenario.**

I. INTRODUCTION

Research in the field of Visual Simultaneous Localization and Mapping (VSLAM) has experienced an increased interest with the availability of affordable 3D camera, e.g. the Microsoft Kinect, in which depth information is added to the color image of the camera (thus dubbed RGBD cameras). The strong interest in advanced RGBD camera systems has several reasons. First, in comparison to the typical laser range finder (LRF) based solutions, they cost significantly less while providing a similar accuracy (albeit at a restricted field of view). But more importantly, in context of this paper, the camera image enables new possibilities by using advanced image processing techniques to deduce information about the environment. This is usually accomplished with a feature detector and descriptor algorithms.

A feature detection algorithm computes points of interest (keypoints) in an image. These will typically be corner-like features, as here features have a high repeatability. The feature detection algorithm thus efficiently reduces the amount of data to be analyzed, reducing the processing time of the subsequent steps of the VSLAM algorithm.

A feature description algorithm computes a distinct signature for a keypoint with the goal to correctly identify the same keypoint across several images. The feature signature is thus aimed to be invariant towards illumination, scale, or translation. In VSLAM, it can then, for example be used to compute the motion of the camera or find previously visited parts of the environment.

In this paper, we compare the performance of state-of-the-art feature descriptors in two VSLAM scenarios with respect to position accuracy. We will specifically evaluate the effect of the camera motion speed on realtime performance. We aim to draw generally applicable conclusions to enable others to a more profound choice for specific VSLAM applications, especially with respect to the more recently introduced descriptors, which have not yet been evaluated in this respect.

The authors are with the Institute for Computer Engineering, University of Lübeck, Germany. Contact: `hartmann@iti.uni-luebeck.de`

The remainder of this paper is structured as follows. First, related work will be introduced below. The feature descriptors to be evaluated are described in more detail in Sec. II. Our VSLAM algorithm will be introduced in Sec. III. The experimental results will then be shown in Sec. IV and conclusions will be finally drawn in Sec. V.

A. Related Work

There have been publications of numerous VSLAM approaches in the last years. Recent years have most notably shown the emergence of graph-based VSLAM approaches. Here, a map of the environment is built by capturing keyframes every couple of frames. The keyframes are linked by estimating the transformation between them based on feature descriptor correspondences and the depth data of the camera image. The keyframes may then form the nodes and the transformation links the edges of a graph, which can be optimized for a globally optimal solution in a least squares sense [1]. Accurate and efficient solutions have for example been introduced in [2], [3]

Gradient histogram-based feature descriptors, namely SIFT [4] and SURF [5], have been frequently used in VSLAM. They are said to deliver the most descriptive features, but are computationally expensive. Tree-based algorithms can be used to find corresponding features in different images [6].

Binary feature descriptors aim to compute similarly descriptive features at a fraction of the computation cost of SIFT and SURF, the most prominent of which being BRIEF [7], ORB [8], BRISK [9], or FREAK [10]. Here, the feature descriptor is computed by pairwise intensity comparison tests around the detected keypoint. The FAST algorithm [11] is typically used for feature detection. The descriptors differ in terms of computation speed, degree of invariance, and choice of pairs for the comparison test. Binary descriptors can be very efficiently matched using the Locality Sensitive Hashing (LSH, [12]) algorithm.

Of course, feature descriptors have been extensively compared in the past. A comparison of feature detectors - the Harris detector [13], the Lucas-Kanade-Tomasi tracker [14], and the detector part of SIFT - for VSLAM has been performed by Klippenstein and Zhang [15]. They evaluate the feature detectors in a Extended Kalman Filter-based mono camera SLAM system and conclude that the choice of detector is irrelevant. They however do not use a feature descriptor for estimating correspondences and rely on wheel odometry.

The work of Juan and Gwun compares SIFT, PCA-SIFT and SURF regarding detection rate and illumination as well as rotation changes [16]. They focus on the comparison in general and not for an explicit application as VSLAM. The recent published RGBD SLAM system from Endres et al. [2] is benchmarked using publicly available benchmark datasets [17]. They present results using SIFT, SURF and ORB, in which SIFT and SURF significantly outperform ORB. In their test, however, the effect of real-time execution is not evaluated. Due to our differently structured algorithm, we have further had better experiences with binary descriptors as will be apparent in the following sections.

II. FEATURE EXTRACTION

This sections gives a more detailed overview on the feature description as well as feature detection algorithms evaluated in this paper. We will try to differentiate the descriptors in terms of method, degrees of invariance, and efficiency.

A. SIFT

The Scale-Invariant Feature Transform (SIFT, [4]) encompasses both a feature detection and feature description algorithm. Feature extraction is performed in four steps. First, keypoints are detected by searching for extremas of a difference-of-Gaussian function at different scales. The location of keypoints is then further refined. The orientation of keypoints is estimated based the local image gradient. A 128-element feature descriptor is then derived from local image gradients in a region defined by the scale and orientation of the keypoint. The descriptor is normalized and thresholds are applied to limit the effects of image saturation.

The SIFT descriptor is largely invariant to scale, orientation, and illumination changes. It is, with SURF, likely the most frequently used feature descriptor in SLAM and often seen as the gold standard in feature extraction. Yet, its computation is very complex and thus often not sufficient for realtime application.

B. SURF

The Speeded-Up Robust Features (SURF, [5]) is largely inspired by SIFT and specifically aimed to improve the computation time. Features are detected using a Fast Hessian detector at different scales, the descriptor estimated based on local image gradients. To increase the efficiency, SURF relies on integral images to estimate image convolutions needed for the Hessian and gradient measures. SURF estimates a 64- or 128-element descriptor, which is similarly invariant to scale, orientation, and illumination as SIFT.

C. BRIEF

Binary Robust Independent Elementary Features (BRIEF, [7]) are the first approach to binary image descriptors derived from intensity difference tests. Such a test yields 1, if the intensity at one specific location is larger than at another specific location, 0 otherwise. The BRIEF descriptor is formed by conducting the intensity difference test on a number (128, 256, or 512) of randomly chosen but fixed location pairs in a pre-defined patch around a keypoint.

The BRIEF descriptor is invariant to illumination changes, but not to rotation or scale. It does not come with a specific feature detection algorithm, for which, typically, the FAST corner detector [11] is used. FAST compares the intensity of a point of interest with the intensities on a circle around that point. A decision tree is learned to efficiently classify corner-like features.

D. ORB, BRISK, and FREAK

Based on the BRIEF descriptor, several feature description algorithms have been introduced to improve in terms of invariance to scale and rotation and choice of intensity test pairs.

Oriented FAST and rotated BRIEF (ORB, [8]) introduces a method of orientation estimation to the FAST detection algorithm, thus providing independence to rotation. Keypoints are further detected at different scales. A resulting loss of variance is counteracted by choosing a fixed set of intensity test pairs with minimal correlation.

The Binary Robust Invariant Scalable Keypoints (BRISK, [9]) further improve the BRIEF concept. The BRISK detector is based on the AGAST detector, which improves FAST by choosing a more robust method of estimating the decision tree. BRISK adds a sophisticated method for estimating keypoints at different scales. The 512-element BRISK descriptor significantly differs from BRIEF, as intensity tests are conducted on a regular circular pattern around the keypoint, using each location multiple times to improve the lookup performance. Additionally, intensity values are smoothed based on the distance to the keypoint location and the keypoint orientation is estimated based on long-distance pairs. BRIEF thus exhibits similar properties as the complex SIFT and SURF descriptors.

Most recently, the Fast Retina Keypoint algorithm (FREAK, [10]) was introduced. It uses the BRISK feature detector and, like BRISK, a circular sampling pattern which, is inspired by the retinal ganglion cells distribution in the human eye.

E. Matching

The efficient estimation of feature correspondences is an important part of the VSLAM algorithm. Matching is typically performed in a nearest neighbor search. Here lies one of the main drawbacks of gradient-based feature descriptors, as SIFT and SURF, who require more computationally expensive matching algorithms due to their high dimensionality. Matching speed can be improved by using tree-based approaches, as for example the Fast Library for Approximate Nearest Neighbors (FLANN, [6]). FLANN encompasses several efficient algorithms for nearest neighbor search in high-dimensional data (randomized kd-trees or hierarchical k-means trees) and allows to automatically choose the appropriate algorithms for the current dataset.

Binary descriptors allow to determine nearest neighbors based on the minimal Hamming distance, which is faster to compute. The Locality Sensitive Hashing (LSH, [12]) algorithm may significantly reduce the number of features

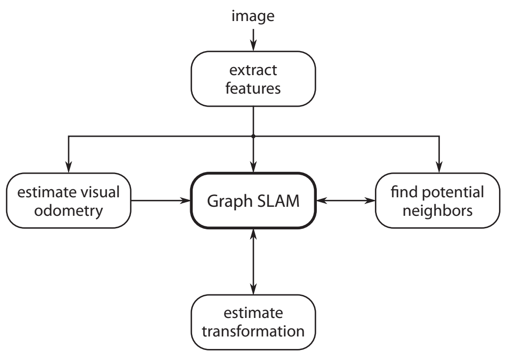

Fig. 1: General dataflow between the different modules of the VSLAM algorithm.

that need to be evaluated by only considering those descriptors that share equal subsets of descriptor bits. This can be efficiently implemented using a randomized set of hash tables.

III. THE SLAM SYSTEM

The VSLAM system used in this evaluation is based on the graph-based visual SLAM approach as described in [18]. The local camera motion is estimated in form of a visual odometry estimate. A new node in the SLAM graph is constructed after the camera has moved a pre-defined distance (5° rotation or $0.1m$ translation in our experiments). The node stores a global pose as obtained by the visual odometry and the feature descriptors and their corresponding 3D position. Similar nodes are searched using the Global Feature Repository (GFR). The links between similar nodes are then estimated and the resulting SLAM graph optimized to obtain a globally consistent map of the environment.

The most important parts of the VSLAM algorithm, as shown in Fig. 1, will be explained in more detail in this section.

a) Visual Odometry: To obtain a stable visual odometry estimate, the local camera motion is always estimated relative to several of the previous frames. Current image features are matched to the previous features. The motion is then estimated based on the correspondences and 3D data obtained from the depth image in a RANSAC framework. We therein use the EPnP algorithm [19] to estimate the camera motion. In our previous work, we have used graph optimization to further refine the visual odometry results, but have in these experiments dropped the optimization in favour of a faster computation. All frames in a window of $500ms$, but at least the last 5 frames are considered.

b) Global Feature Repository (GFR): The Global Feature Repository (GFR) is a means of efficiently retrieving similar images to the current image in the SLAM graph. In contrast to the popular vocabulary tree-based approaches, e.g. [20], it can efficiently handle the binary descriptors as well as SIFT or SURF. Whenever a new node is added to the graph, the GFR matches the new feature descriptors to all previous feature descriptors, using either FLANN or LSH depending

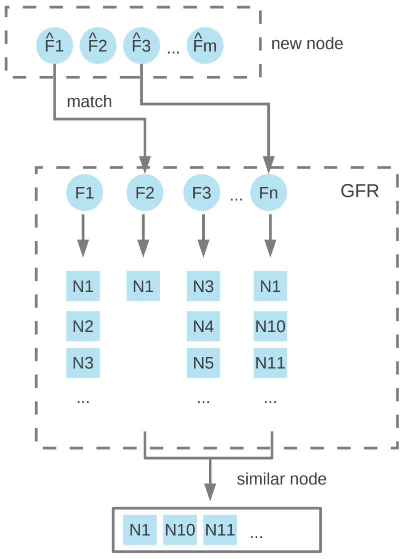

Fig. 2: Architecture of the Global Feature Repository (GFR). Every feature descriptor F_i in the GFR holds references to all nodes in which F_i has been detected. Given a new node, similar nodes can be found by matching the descriptors $\hat{F}_j$ with the descriptors in the GFR.

on the descriptor type. If there was a positive match, a reference to the new node is added to the matched descriptor, otherwise the new descriptor is added to the repository.

Similar images and thus potential neighbors to a node can now be similarly found. The query node descriptors are matched to all descriptors in the GFR. The most similar nodes are then those nodes that occur most often in the references of the matched descriptors of the GFR. See Fig. 2 for a visual representation of this process.

c) Link Estimation: The transformation between two nodes is, similarly to the visual odometry, estimated by matching the descriptors of the two nodes. From the feature correspondences, the transformation is now estimated using a simple SVD-based algorithm as, for example, is also utilized in [2], as we have found the EPnP algorithm to deliver poor results for larger position and orientation changes.

d) Graph SLAM: The graph SLAM backend manages the nodes and links to obtain a globally consistent map with minimal error. The error, here, is the difference between the global position of a node to the position as obtained by the link estimation, i.e. relative to neighbor nodes. The SLAM graph is therefore optimized using the general graph optimization framework g2o [1], which efficiently minimizes the error in a least squares sense.

In addition to our previous efforts, in this work, the covariance matrix of link estimates will be dynamically adjusted to accommodate for potentially erroneous link transformations. After graph optimization, the link covariance will be increased, if the link error exceeds a pre-defined value (in our experiments $0.3m$), and decreased otherwise. If the covariance grows too large, the link is erased from the graph. This has proven to provide more reliable results whenever environments are repetitive, e.g. in hallways.

IV. EXPERIMENTAL RESULTS

The feature descriptors presented in Sec. II will be evaluated using two different datasets. First, we will use the *FR1* test sequences of the RGBD SLAM dataset [17], which have been frequently used to benchmark VSLAM algorithms, e.g. in [2] and in our previous experiments [18]. The RGBD SLAM dataset provides a varied set of test sequences in an office environment with different degrees of translational and rotational movement. The dataset was captured using a handheld Microsoft Kinect camera, ground truth from an external tracking system is provided. Error measurements were evaluated using the tools provided with the RGBD SLAM dataset[1].

We will further evaluate the feature descriptors using the first three test sequences of our own datasets[2]. The ITI dataset was recorded in the computer science building of the University of Lübeck, the three test sequences show path lengths between $230m$ and $330m$ containing large parts of the hallway, a repetitive environment containing few visual features, and require to close loops as large as $200m$. The dataset was recorded using a Microsoft Kinect camera mounted on a mobile robot. A laser range finder (LRF) localization estimate is used as ground truth. The ITI dataset encompasses an environment that we would expect to find for a "typical" wheeled mobile robot.

The VSLAM algorithm, as shown in Sec. III, was implemented in the Robot Operating System (ROS) framework. More importantly, for the implementation of the feature extraction algorithms, we have used the OpenCV 2.3[3] library as provided by ROS. For the BRISK descriptor, which is not available in OpenCV 2.3, we have used the implementation provided by the authors[4]. We have used the feature description algorithms with standard parameters, but restricted the maximum number of keypoints to 300. Experiments were run on a Intel Core i7 quad-core processor clocked at $3.4GHz$.

For the evaluation of the different feature descriptors, we will, first, discuss the general performance in the next section. The datasets for the two two datasets will then be shown in Sec. IV-B and IV-C. Here, we evaluate the accuracy in terms of position error for visual odometry and SLAM seperately. For the RGBD dataset, we will further evaluate the performance with respect to realtime capabilities, showing how the mean position error changes with different playback speeds of the image data.

A. General Feature Descriptor Performance

Tab. I shows several measures estimated over all test runs, that we find relevant in terms of general performance of the feature descriptors. Most important may be the mean extraction time per keypoint (first column). While BRIEF is (in combination with FAST) indeed the most efficient feature descriptor, BRISK and FREAK are significantly slower. SURF, as expected, is even slower. But most surprisingly, SIFT matches BRISK and FREAK in terms of extraction time. Of course, these measurements may be the result of the small image size (640x480 pixels) and very much depend on the specific implementation, yet, as the OpenCV library is frequently used, we feel that they are nevertheless relevant.

[1]http://vision.in.tum.de/data/datasets/rgbd-dataset
[2]http://www.iti.uni-luebeck.de/navigation.html
[3]http://opencv.willowgarage.com/wiki/
[4]http://www.asl.ethz.ch/people/lestefan/personal/BRISK

descriptor	**I**	**II**	**III**	**IV**
BRIEF	0.0364 ms	63.9%	24.0 ms	3.90
BRISK	0.2259 ms	49.9%	23.1 ms	3.30
FREAK	0.2943 ms	51.2%	20.2 ms	3.04
ORB	0.0634 ms	53.1%	21.0 ms	3.49
SIFT	0.2508 ms	49.7%	34.5 ms	2.72
SURF	0.4205 ms	48.5%	41.4 ms	2.38

TABLE I: General performance for the feature descriptors. **I**: mean feature extraction time per keypoint. **II**: Percentage of geometrically correct matches divided by the total number of matches in the visual odometry. **III**: mean GFR time. **IV**: links per node.

The third column similarly shows the mean time for GFR place recognition. Here, the binary descriptors outperform the floating point-based ones significantly. Nevertheless, as the GFR search is only performed on the creation of a new node, the difference will likely not bear a great effect on the general performance.

The other two measures are related to the recognition accuracy of the descriptors. First, we have evaluated how many feature matches are detected as true positives in geometric terms by the visual odometry algorithm. Here, the BRIEF descriptor outperforms the other descriptors, surprisingly also the more complex binary descriptors. We indeed attribute this to the fact that BRIEF is not invariant to scale and orientation. When images are very similar and only gradually change, this seems to be an advantage, as a false orientation may be estimated from the other descriptors (this is backed by results in e.g. [7], where BRIEF shows good performance for low image changes).

The fourth column shows the number of edges that were constructed per node in the SLAM graph. Here, the binary descriptors generally find more links, with SURF finding significantly less edges than the other descriptors. Here, we assume the reason to be of similar nature as with the feature matches and links are more frequently found for very similar images.

B. RGBD SLAM Dataset

The experimental results for the RGBD SLAM dataset are shown in Fig. 3. The two plots on top show an analysis of the mean error over all test sequences for different speeds, which were emulated by playing back the datasets at different rates. A speed of 1.5 here means that the motion is faster by 50%, a speed of 0.1 corresponds to one 10th speed of motion.

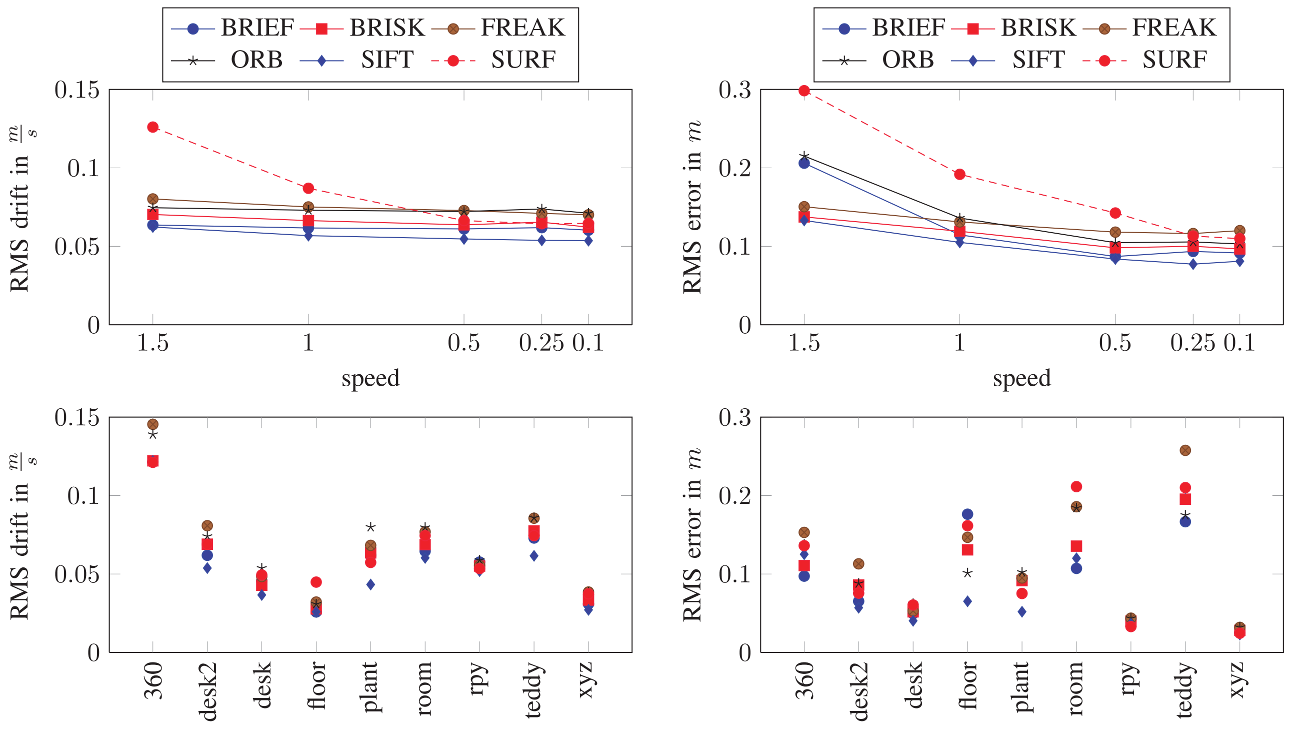

Fig. 3: Accuracy with different feature descriptors for the RGBD SLAM dataset. Visual odometry drift is shown in $\frac{m}{s}$ (left), SLAM position error in m (right). The top plots show the mean accuracy over all test sequences at different rates, where the test sequences were played back with the given speedup, simulating different movement speeds of the robot. The bottom plots show the results for a speed of 0.1, as a measure for best case results, for each test sequence. All values were estimated for 10 runs on each test sequence, descriptor, and speed.

In general, we see an increased accuracy with a lower speed, as could be expected. SIFT performs best at all speeds, BRIEF mostly performs better than the other binary descriptors. The feature extraction time (see Tab. I), therefore does not seem to significantly affect the accuracy. Here, the exception is SURF, in which case we account the poor performance, especially for the visual odometry, to the poor extraction time. For speeds of 0.5 and less, we see no significant change in both odometry and SLAM accuracy.

The two bottom plots show the error for each test sequence separately for the best case scenario, i.e. a speed of 0.1. Generally, the accuracy in odometry translates directly to the accuracy in SLAM for many test sequences (desk, desk2, plany, rpy, xyz). The other sequences show a larger difference between the descriptors. Especially the floor, room, and teddy sequences exhibit large loops, with loop closures only at the end of the sequence, where SURF and FREAK seem to fail at consistently detecting these loop closures. The 360 sequence exhibits frequent loop closures and therefore enables the SLAM algorithm to better correct the large odometry drift for all descriptors.

These results do not fully represent what was to be expected. SIFT performs very good, as was shown in numerous other publications, yet SURF performs poorly, especially within realtime requirements. This seems to be a problem of the implementation, rather than the descriptor, which was found to deliver good results previously. Yet, as the OpenCV framework is very popular, it matters nonetheless.

The second surprise was the good performance of BRIEF, which showed better results than the other binary descriptors. As was already mentioned in the last section, this may be largely attributed to the specifics of the SLAM problem. Images are very similar in case of visual odometry estimation and the estimation of node links, the abandonment of some degrees of invariance may in this case give more accurate results.

C. *ITI Dataset*

Tab. II shows the results obtained with the ITI dataset at realtime rate (results do not significantly improve with lower speed). Please note, that we have used the wheel odometry for this purpose. The results should therefore reflect a more traditional problem, as would be encountered with wheeled mobile robots. It should further reflect the degree of accuracy that can be obtained using a single RGBD camera in large scale environments.

In this scenario, results are very similar for all descriptors (with the exception of FREAK), with mean RMS errors of around $40cm$. FREAK is not able to close the loop in the first test sequence for one test run. Again, the SIFT descriptor performs best.

In the end, the results for a wheeled robot SLAM scenario reinforce the findings of [15]. When using wheel odometry,

descriptor	iti4_1	iti4_2	iti4_3	mean
BRIEF	0.2571	0.4068	0.5295	0.3978
ORB	0.3335	0.3919	0.4728	0.3994
BRISK	0.2215	0.4195	0.5030	0.3813
FREAK	1.1570	0.4086	0.6979	0.7545
SIFT	0.2387	0.3698	0.4883	0.3656
SURF	0.3467	0.3955	0.4945	0.4123

TABLE II: RMS SLAM position error in m for the ITI test sequences using wheel odometry. All values were estimated for 5 runs on each test sequence and descriptor.

the choice of feature descriptor does not significantly affect the accuracy of the SLAM algorithm. All descriptors (with the exception of FREAK) are able to close large loops and provide accurate results at larger scale.

V. CONCLUSION

In this paper, we have shown experimental results on the comparison of feature description algorithms for the specific task of Visual Simultaneous Localization and Mapping (VSLAM). In VSLAM, feature descriptors are used to efficiently estimate the local motion of the camera, i.e. visual odometry, and find previously visited parts of the environment. Gradient histogram-based descriptors, e.g. SIFT and SURF, are frequently used in VSLAM, but are said to be computationally too expensive for realtime application. More recently established solutions in the field of binary descriptors claim to improve the feature extraction runtime significantly while producing similar results.

We have compared the SIFT, SURF, BRIEF, ORB, BRISK and FREAK descriptors on two different datasets. The RGBD SLAM dataset evaluates VSLAM with a handheld camera and complex 3D motion. Here, we have especially evaluated realtime performance of the descriptors by altering the playback speed of the dataset. Our own dataset evaluates VSLAM in a typical wheeled robot scenario.

For both datasets, SIFT performs best, even in the evaluation of the realtime capability. In case of our own datasets, all descriptors perform similarly well, but results differ for the RGBD SLAM dataset. While for the former the choice of descriptor does not play an important role in the accuracy of the VSLAM algorithm, the choice of descriptor in the latter will make a significant difference, especially in terms of realtime performance. Here, SURF performs poorly. We attribute this to the specific implementation of the feature descriptors, in which the feature extraction time of SIFT is surprisingly faster than SURF. A remarkable result could be obtained for the binary descriptors, where the most simple and efficient descriptor BRIEF performs best, with an accuracy close to SIFT. We assume this to be a result of the lack of invariance to scale and rotation. In the specific problem of VSLAM, where images change only gradually and are generally very similar, the estimation of orientation and scale may then indeed reduce the accuracy.

To conclude in a general statement, SIFT will be a good choice for feature descriptor for any VSLAM scenario, as has been similarly shown in other feature descriptor comparisons. Whenever computational load is an important factor, BRIEF, which will improve the feature extraction time by a factor of 7, can be used with similar accuracy in most cases. A combination of these two descriptors, e.g. BRIEF for visual odometry and SIFT for the linking of places in the SLAM map, may be interesting to have the best of both worlds. We will continue the comparison of feature descriptors in VSLAM in future works, analyzing more closely the performance in different methods of place recognition.

REFERENCES

[1] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g2o : A general framework for graph optimization,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 3607–3613.

[2] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An Evaluation of the RGB-D SLAM System,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[3] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, “Double window optimisation for constant time visual SLAM,” in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011, pp. 2352–2359.

[4] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[6] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *Int. Conf. on Computer Vision Theory and Application*, 2009, pp. 331–340.

[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary robust independent elementary features,” in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010, pp. 778–792.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[9] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011, pp. 2548–2555.

[10] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK : Fast Retina Keypoint,” *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 510–517, 2012.

[11] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European Conf. on Computer Vision (ECCV)*, vol. 1, May 2006, pp. 430–443.

[12] A. Gionis, P. Indyk, and R. Motwani, “Similarity search in high dimensions via hashing,” in *Proc. of the Int. Conf. on Very Large Data Bases (VLDB)*, 1999, pp. 518–529.

[13] C. Harris and M. Stephens, *A combined corner and edge detector*, 1988.

[14] J. Shi and C. Tomasi, “Good features to track,” *IEEE Conf. on Computer Vision and Pattern Recognition CVPR94*, 1994.

[15] J. Klippenstein and H. Zhang, “Performance evaluation of visual SLAM using several feature extractors,” in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[16] L. Juan and O. Gwun, “A Comparison of SIFT , PCA-SIFT and SURF,” *International Journal of Image Processing IJIP*, 2009.

[17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “Towards a benchmark for RGB-D SLAM evaluation,” *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf. (RSS)*, 2011.

[18] J. H. Klüssendorff, J. Hartmann, D. Forouher, and E. Maehle, “Visual Graph-Based SLAM and Visual Odometry using an RGB-D Camera,” in *To Appear in Proc. of the 9th International Workshop on Robot Motion and Control (RoMoCo)*, 2013.

[19] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An Accurate O(n) Solution to the PnP Problem,” *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, July 2008.

[20] D. Nister and H. Stewenius, “Scalable Recognition with a Vocabulary Tree,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR’06)*, 2006.