Acta Polytechnica **00**(0):1–6, 0000

PREPRINT 2015-09-26

IMPACT ASSESSMENT OF IMAGE FEATURE EXTRACTORS ON THE PERFORMANCE OF SLAM SYSTEMS

Taihú Pire[a,*], Thomas Fischer[a], Jan Faigl[b]

[a] *University of Buenos Aires, Intendente Güiraldes 2160, Ciudad Autónoma de Buenos Aires, Argentina*
[b] *Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27, Prague, Czech Republic*
[*] corresponding author: tpire@dc.uba.ar

Abstract. This work evaluates an impact of image feature extractors on the performance of a visual SLAM method in terms of pose accuracy and computational requirements. In particular, the S-PTAM (Stereo Parallel Tracking and Mapping) method is considered as the visual SLAM framework for which both the feature detector and feature descriptor are parametrized. The evaluation was performed with a standard dataset with ground-truth information and six feature detectors and four descriptors. The presented results indicate that the combination of the GFTT detector and the BRIEF descriptor provides the best trade-off between the localization precision and computational requirements among the evaluated combinations of the detectors and descriptors.

Keywords: Image Features, Visual SLAM, Stereo Vision.

1. Introduction

During the last decade, the Simultaneous Localization and Mapping (SLAM) problem has been one of the main research interests in mobile robotics. Particularly, the use of cameras as the main sensors has been given a special attention [1] [2] [3] [4] because of their benefits such as low-cost and passive sensing. In vision-based SLAM approaches, local image features are used to build a map and simultaneously estimate the robot pose using the environment landmarks represented as the image features. In this way, the map is represented as a sparse point cloud, where each point results from triangulating salient points (image features) matched from a pair of stereo images.

Currently, there exist several local image feature extractors in the literature. A feature extractor is a combination of a salient point (called *keypoint*) detection procedure and a computation of a unique signature (called *descriptor*) for each such detected point. The most commonly used detectors are SIFT [5], SURF [6], STAR [7], GFTT [8], FAST [9], and relatively recently proposed ORB [10], while among the most used descriptors we can mention SIFT, SURF, ORB, BRIEF [11], and BRISK [12].

In Visual SLAM systems, the feature extraction process has a huge impact on the accuracy of the whole system. On one hand, the precision of the robot localization is heavily correlated to the sparsity of features in images and the ability to track them for a long period during the robot navigation, even from different points of view. On the other hand, if the number of points in the map grows too quickly, it may slow down the whole system. To be able to keep the response of the system under real-time constraints, images have to be dropped or other parts of the system, like optimization routines, need lower computational requirements.

In this work, we evaluate the impact of different state-of-the-art feature extractors on the performance of the Visual SLAM localization method. In particular, the evaluation is based on the stereo Visual SLAM approach *S-PTAM* introduced in [4]. The presented results indicated that the combination of the GFTT detector and BRIEF descriptor is the most reliable choice for our SLAM system among the other evaluated combinations.

The rest of the paper is organized as follows. Section 2 presents overview of the related work while Section 3 summarizes the most used feature detectors and descriptors in the Visual SLAM literature. Section 4 briefly comments the considered stereo Visual SLAM system using for the evaluation. In Section 5, we present the evaluation of the features extractors and the achieved results. Section 6 is dedicated to the conclusions and future work.

2. Related Work

Several evaluations of features extractors can be found in literature. Each of them is driven by the particular application or issue at the hand they aimed to address. For example, in [13], authors evaluate several features extractors in the context of autonomous navigation in outdoor environments under seasonal changes. They came to a conclusion that the best performing method is the STAR–BRIEF combination of the detector–descriptor, which outperforms SIFT by more than thirty percentage points. In addition, they argued that the STAR–BRIEF extractor is also less computationally demanding than other extractors and thus it seems to be the most suitable feature detector–descriptor for navigational purposes.

On the other hand, authors of [14] provide a performance comparison of feature extractors against illumination changes in outdoor scenes in the context of the visual navigation. They concluded that the configuration of the FAST–SURF is the optimal in their setup. Besides, they report that this combination provides an effective computational time per image, which is favorable for the real-time vision-based navigation application.

The work [15] compared contemporary point features detector and descriptor pairs in order to determine the best combination for the task of robot visual navigation. They concluded that the FAST-BRIEF combination is a good choice when processing speed is an important parameter of the system setup. They also argued that under camera movement conditions, additional computational cost—needed for the descriptors and detectors that are robust to in-plane rotation and large scaling— seems to be unjustified. However, they do not tested the method in a real SLAM application.

Regarding the aforementioned evaluation of the detectors and descriptors, the work presented in this paper is within the context of the full 6DOF SLAM.

3. Local Image Features

An image feature extractor consists of detection and description phases. The feature detector serves to locate salient areas of the image while the feature descriptor captures information about the local neighborhood of the detected area. Here, we provide a brief overview of the considered feature extractor and descriptor algorithms in this evaluation study.

SIFT – Scale Invariant Feature Transform [5]. An established feature detector with a high precision and good robustness, which is known to be computationally demanding.

SURF – Speeded Up Robust Features [6] is a similar to SIFT, but it is computationally less demanding due to approximations.

STAR – A modified version of the CenSurE (Center Surrounded Extrema) [7] detector, which is computationally less demanding at the expense of a lower precision.

BRIEF – Binary Robust Independent Elementary Features [11] is a descriptor that describes image areas using a number of intensity comparisons of random pairs. It is saved as a binary string, which reduces the computational complexity of the subsequent matching.

FAST – Features from Accelerated Segment Test [9] is a feature detector focused on lowering the computational cost.

BRISK – Binary Robust Invariant Scalable Keypoints [12] is a scale and rotation invariant version of BRIEF, but unlike BRIEF, it uses a deterministic comparison pattern.

ORB – Oriented FAST and Rotated BRIEF [10] is another attempt to achieve a scale and rotation invariant BRIEF, as an efficient alternative to SIFT and SURF. It uses the FAST detector to achieve low computational requirements.

GFTT – A detector focused on selecting features relevant to motion tracking by analyzing the amount of information they provide for that particular task [8].

The SURF and SIFT descriptors rely on the their own detectors, which are also considered in the presented evaluation. However, for the BRIEF and BRISK binary descriptors the considered detectors are the GFTT, FAST and STAR which results in the additional six combinations of the detector–descriptor in the evaluation.

4. Overview of S-PTAM

S-PTAM [4] is a stereo Visual SLAM method for a large scale map navigation based on the known monocular Parallel Tracking and Mapping (PTAM) method introduced in [1]. The method consists of two processes working in parallel: 1) the tracking of the detected features and; 2) creating a map of the features (mapping). During a robot navigation, the method works as follows.

S-PTAM extracts features from the incoming stereo images to match and construct a virtual map of the environment. The newly extracted feature descriptors are matched against descriptors of the points stored in the map according to the estimated field of view. The matches may then be used to refine the estimated camera pose using an iterative least squares minimization method, e.g., using the Levenberg-Marquardt algorithm. The particular stereo matches between the features that cannot be matched to the map are triangulated and inserted as new map points, for the tracking of future frames. In parallel, a map refinement algorithm is running. It is also based on the Levenberg-Marquardt optimization that continuously performs the Bundle Adjustment on the current local portion of the map.

In [4], S-PTAM uses the GFTT feature detector and the BRIEF descriptor extractor. In this work, we consider other combinations of the detector and descriptor to evaluate an impact of the combination to the performance of the localization and mapping processes.

5. Evaluation

The KITTI Vision Benchmark Suite [16] is used to evaluate S-PTAM for each type of considered detector–descriptor configuration. In particular, we present the results obtained for the sequence 00, shown in Figure 1. The sequence records the stereo camera frames captured by a moving car in an urban scenario for almost 4 km long path. The particular parameters of the evaluated feature extractors have been selected in such a way that allows S-PTAM to run without ever loosing localization. They have been tuned from a strong restrictive value and then relaxed until the method completes the whole sequence. The parameters are listed in Table 1.

Detector / Descriptor	Parameter	Value
SIFT	`nOctaveLayers`	1
	`L2NormThreshold`	100
SURF	`hessianThreshold`	1000
	`nOctaves`	1
	`L2NormThreshold`	0.2
STAR	`responseThreshold`	20
BRIEF	`bytes`	32
	`hammingThreshold`	25
FAST	`threshold`	60
BRISK	`hammingThreshold`	100
ORB	`nfeatures`	2000
	`nLevels`	1
	`hammingThreshold`	50
GFTT	`nfeatures`	2000
	`minDistance`	15.0

TABLE 1. Parameters used for feature detector and descriptor extractors. The parameters which do not appear in the list use the default value in the OpenCV implementation. In the case of the binary descriptors, the Hamming distance is used to compute the valid matches while the L2 norm is used for the SURF and SIFT descriptors.

The evaluation has been performed using an Intel Core i7 processor with 4 cores running at 2.2 GHz. Although S-PTAM strongly exploits parallelism, the experiments were run in a sequential fashion that allow us to simulate ideal conditions and abstract from the limitations of the available computational power. This ensures that no frames are dropped and that the iterative optimization routines always converge or reach a maximum threshold of iterations.

Nevertheless, the tracking process performs pose optimization using an iterative algorithm; so, the less time is used in the features extraction, the more iterations the method can compute. Figure 2 shows a characterization of the total tracking time for each pair of frames, as achieved by using the evaluated extractors.

Moreover, the iterative least-squares optimization, which is utilized in the mapping and tracking processes, depends linearly on the number of tracked points (the density of the map). Thus, regarding the computational burden, the map should be as small as possible while the map points should contain strong

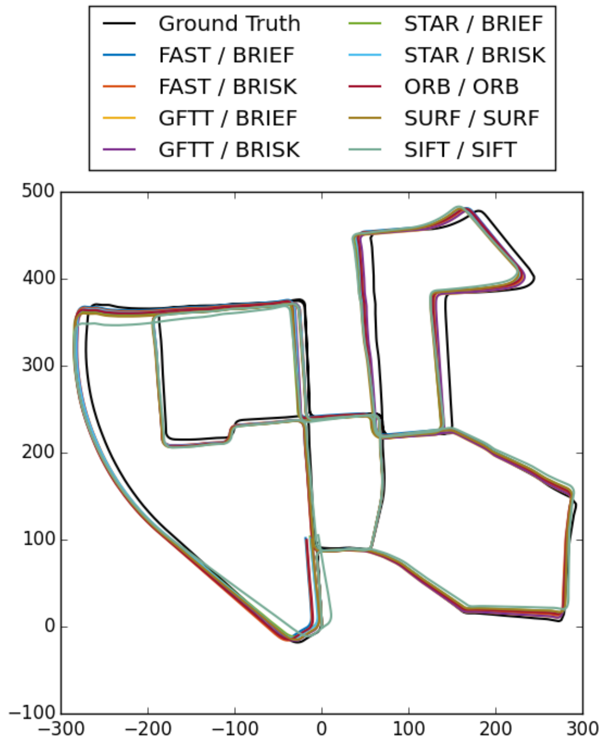

FIGURE 1. Path tracked by every method run under different extractors, against the ground truth. The path is nearly 4 km long. The shown distances at the axes are in meters.

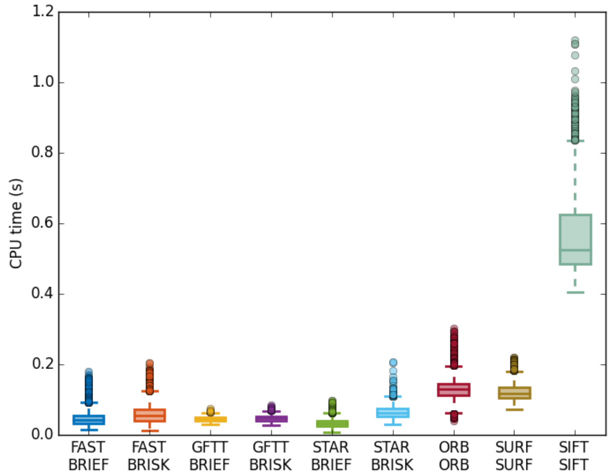

FIGURE 2. Total tracking time achieved by each configuration

enough features to support a robust tracking of the frames. Table 2 shows the final number of points contained in the map after finishing each trial for a particular combination of feature detector and descriptor. In Figure 3, we can see how the map size impacts directly on the temporal performance of the tracking process. Combinations of the detector–descriptor that build the most dense maps also take the longest time to compute.

Differences in the map size for the evaluated descriptor extractors with the same detector can have

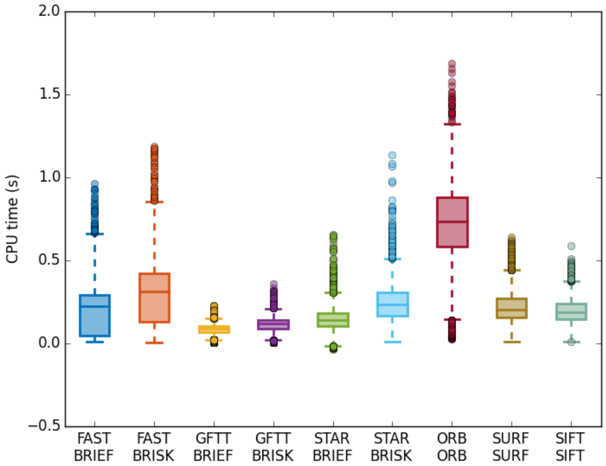

Figure 3. Tracking time without taking into account feature extraction

two reasons. The first reason is that new points are created from the stereo features only if these features are not matched to the map. The second reason is that the points marked as outliers during the refinement processes are discarded. In the first case, this can be caused by descriptors that are not robust enough to be matched to the map for a long time. In the second case, the descriptor matching may be too permissive and it allows bad matches that are later discarded as outliers.

Extractor	Final map size
GFTT / BRIEF	990 455
GFTT / BRISK	1 314 356
SIFT / SIFT	1 581 876
STAR / BRIEF	1 893 372
SURF / SURF	2 059 879
FAST / BRIEF	2 420 652
STAR / BRISK	2 447 418
FAST / BRISK	3 207 003
ORB / ORB	5 192 885

Table 2. The number of points contained in the map after completing the sequence for each evaluated extractor, in ascending order.

Since the goal of this work is to assess the impact of the feature extractor choice also on the accuracy of the SLAM method, the achieved performance is presented as two independent relative errors for each estimated pose: ϵ_t for the translation error and; ϵ_θ for the orientation. Let $\mathbf{x}_k$ be the estimated pose at the frame k, which can be decomposed as the translation $\mathbf{t}_k$ and the rotation $\mathbf{R}_k$. Let $\mathbf{x}_k^*$ be the reference pose, which can be decomposed in the same fashion. The aforementioned errors are computed as

$$\begin{aligned}\epsilon_{t,k+1} &= \|(\mathbf{t}_k \ominus \mathbf{t}_{k+1}) \ominus (\mathbf{t}_k^* \ominus \mathbf{t}_{k+1}^*)\|,\\ \epsilon_{\theta,k+1} &= angle\left((\mathbf{R}_k \ominus \mathbf{R}_{k+1}) \ominus (\mathbf{R}_k^* \ominus \mathbf{R}_{k+1}^*)\right),\end{aligned}$$

where $\ominus$ is the inverse of the standard motion composition operator. For pure translations, we can rewrite $t_1 \ominus t_2 = t_2 - t_1$, and for the pure rotations as $R_1 \ominus R_2 = R_1^t R_2$. $\|\mathbf{x}\|$ stands for the Euclidean norm and $angle\,(\mathbf{R})$ extracts the magnitude of the rotation.

The computed errors are shown in Figure 4 and Figure 5, respectively. Although the angular deviation to the ground truth, shown in Figure 5, seems to be similar in all methods, the same is not true for the translation error, as it can be seen in Figure 4. The BRISK descriptor seems to be a more reliable with the FAST detector, while the same holds for the BRIEF descriptor with the GFTT detector.

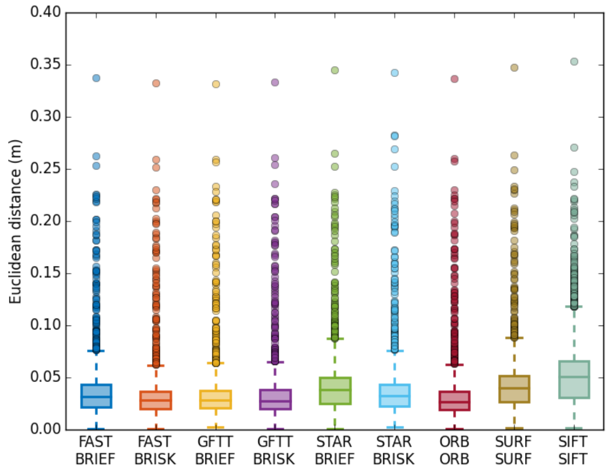

Figure 4. Relative translation error

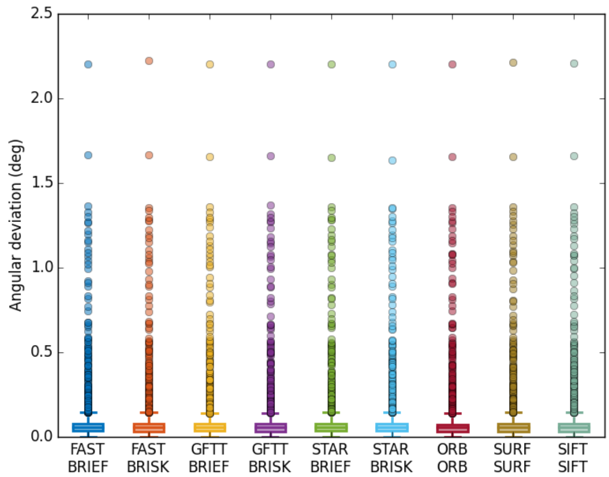

Figure 5. Relative orientation error

For completion, the absolute errors

$$\begin{aligned}{\epsilon'}_{t,k} &= \|\mathbf{t}_k \ominus \mathbf{t}_k^*\|\\ {\epsilon'}_{\theta,k} &= angle\,(\mathbf{R}_k \ominus \mathbf{R}_k^*)\end{aligned}$$

are shown in Figure 6 and Figure 7.

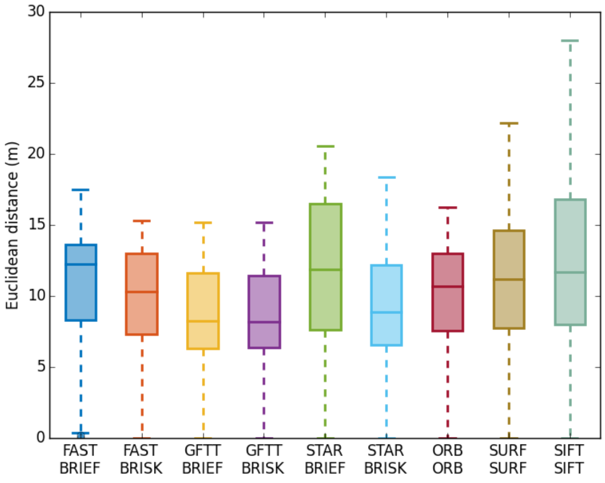

FIGURE 6. Absolute translation error

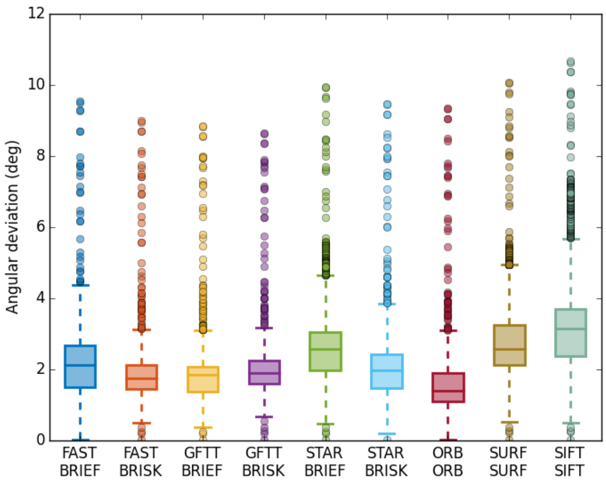

FIGURE 7. Absolute orientation error

6. Conclusions

In this paper, we present an evaluation of the impact of different state-of-the-art image feature extractors on the performance of the SLAM method proposed in [4]. The KITTI Benchmark Suite dataset with a ground truth is used to evaluate the achievable precision of the method for different feature extractors. Based on the presented results, the main conclusion is that the GFTT detector is the most suitable choice for the best performance in the evaluated dataset. The GFTT (accompanied with the BRIEF or BRISK descriptors) outperforms the other methods in terms of the required computational time and the map quality. Although the map density is far smaller, the computed translation error is similar, even slightly better, than the one achieved by other extractors. This insight can be interpreted as the most useful features (regarding the navigation) are extracted while the descriptor also support efficient matching resulting in a more precise localization.

Recently, a novel stereo feature extractors have been proposed, e.g., [17], which motivates us to consider them in S-PTAM. An evaluation of the novel extractors is a subject of our future work.

Acknowledgements

This work is a direct result of the bilateral cooperation program between the Czech and Argentinian Republics support by the Argentinian project ARC/14/06 and travel support of the Czech Ministry of Education under the project No. 7AMB15AR029. The work of J. Faigl is supported by the Czech Science Foundation (GAČR) under the research project No. GJ15-09600Y.

References

[1] G. Klein, D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *ISMAR*, pp. 1–10. IEEE Computer Society, Washington, DC, USA, 2007. DOI:10.1109/ISMAR.2007.4538852.

[2] C. Mei, G. Sibley, M. Cummins, et al. Rslam: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision* **94**(2):198–214, 2011. DOI:10.1007/s11263-010-0361-7.

[3] R. Mur-Artal, J. M. M. Montiel, J. D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *CoRR* **abs/1502.00956**, 2015. DOI:10.1109/TRO.2015.2463671.

[4] T. Pire, T. Fischer, J. Civera, et al. Stereo parallel tracking and mapping for robot localization. In *IROS*. 2015. (to appear).

[5] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60**(2):91–110, 2004. DOI:10.1023/B:VISI.0000029664.99615.94.

[6] H. Bay, T. Tuytelaars, L. Van Gool. Surf: Speeded up robust features. In *ECCV*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417. Springer Berlin Heidelberg, 2006. DOI:10.1007/11744023_32.

[7] M. Agrawal, K. Konolige, M. Blas. Censure: Center surround extremas for realtime feature detection and matching. In *ECCV*, vol. 5305 of *Lecture Notes in Computer Science*, pp. 102–115. Springer Berlin Heidelberg, 2008. DOI:10.1007/978-3-540-88693-8_8.

[8] J. Shi, C. Tomasi. Good features to track. In *CVPR*, pp. 593–600. 1994. DOI:10.1109/CVPR.1994.323794.

[9] E. Rosten, T. Drummond. Machine learning for high-speed corner detection. In *ECCV*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 430–443. Springer Berlin Heidelberg, 2006. DOI:10.1007/11744023_34.

[10] E. Rublee, V. Rabaud, K. Konolige, G. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, pp. 2564–2571. 2011. DOI:10.1109/ICCV.2011.6126544.

[11] M. Calonder, V. Lepetit, C. Strecha, P. Fua. Brief: Binary robust independent elementary features. In *ECCV*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 778–792. Springer Berlin Heidelberg, 2010. DOI:10.1007/978-3-642-15561-1_56.

[12] S. Leutenegger, M. Chli, R. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, pp. 2548–2555. 2011. DOI:10.1109/ICCV.2011.6126542.

[13] T. Krajník, P. de Cristóforis, M. Nitche, et al. Image features and seasons revisited. In *European Conference on Mobile Robotics (ECMR)*. 2015. (to appear).

[14] Dzulfahmi, N. Ohta. Performance evaluation of image feature detectors and descriptors for outdoor-scene visual navigation. In *ACPR*, pp. 872–876. 2013. doi:10.1109/ACPR.2013.159.

[15] A. Schmidt, M. Kraft, M. Fularz, Z. Domagala. Comparative assessment of point feature detectors in the context of robot navigation. *Journal of Automation, Mobile Robotics and Intelligent Systems* **7**(1):11–20, 2013.

[16] A. Geiger, P. Lenz, C. Stiller, R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR* **32**(11):1231–1237, 2013. doi:10.1177/0278364913491297.

[17] R. Arroyo, P. Alcantarilla, L. Bergasa, et al. Fast and effective visual place recognition using binary codes and disparity information. In *IROS*, pp. 3089–3094. 2014. doi:10.1109/IROS.2014.6942989.