Comparative Study of Visual Odometry and SLAM Techniques

Ana Rita Gaspar(✉), Alexandra Nunes, Andry Pinto, and Anibal Matos

INESC TEC, Porto, Portugal
{argaspar,apn,andry.m.pinto}@inesctec.pt, anibal@fe.up.pt

Abstract. The use of the odometry and SLAM visual methods in autonomous vehicles has been growing. Optical sensors provide valuable information from the scenario that enhance the navigation of autonomous vehicles. Although several visual techniques are already available in the literature, their performance could be significantly affected by the scene captured by the optical sensor. In this context, this paper presents a comparative analysis of three monocular visual odometry methods and three stereo SLAM techniques. The advantages, particularities and performance of each technique are discussed, to provide information that is relevant for the development of new research and novel robotic applications.

Keywords: Evaluation · Odometry · SLAM · Vision

1 Introduction

The increased use of the autonomous vehicles is related to the fact that their properties allow its application in diverse tasks that can be dangerous and repetitive for the human. To ensure that vehicles are completely involved in various applications it is relevant to augment their capacity to reliably navigate autonomously even in unknown environments. In this context, a large effort is being made by the researchers to explore the concepts of the odometry and SLAM (Simultaneous Localization and Mapping) in order to support the activity of mobile robots in different scenarios. Odometry allows estimation of the robot's position from a single reference and the SLAM technique localizes the robot and constructs a map of the environment. Therefore, SLAM techniques can use odometry-based methods to provide an estimation of motion. The biggest advantage of SLAM is related with the revisiting capability, which means that, the technique reduces the positioning error along the navigation path once a revisited area is detected. Optical systems have the ability to provide information with high quality at a reasonable cost. Therefore, the development of visual odometry and SLAM approaches have been an active line of research that was followed by a large number of institutions worldwide. The appearance of several

A. Ollero et al. (eds.), *ROBOT 2017: Third Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 694, https://doi.org/10.1007/978-3-319-70836-2_38

visual-based techniques have triggered a fundamental question: *what technique is suitable for a specific application?* Therefore, the major contribution of this paper include: to provide a comparative study of some of the visual odometry and SLAM techniques that are currently available in the literature. Moreover, this paper discusses the performance of these methods for typical applications related to mapping and data fusing with other sensors. Thus, it is important to highlight this kind of studies because it allows to understand the main properties, advantages and disadvantages of each implementation, as well as its results in different environments and testing conditions. Therefore, it is possible to select the best and more convenient method for particular applications. This paper is organized as follows: Sect. 2 presents conventional methods for visual odometry and SLAM. The comparative study is presented in Sect. 3, where the results obtained by several techniques are evaluated in common testing scenarios. Finally, Sect. 4 shows the major conclusions of this paper.

2 Methods

As a prerequisite of the many tasks that involve the robot motion, the localization is the most crucial feature for an autonomous robot. In this sense, the visual odometry estimates motion with only input images from of one or multiple cameras. The use of visual odometry presents advantages compared to traditional method (encoders in wheels) since it is more reliable in slipping events, namely in rugged lands where drift errors can occur frequently. However, the analysis of egomotion from sequence of images are very complicated due to the presence of the external objects moving in the scenario (which violates the motion coherent assumption [1]). It is also necessary to ensure that the rate of acquisition is fast enough to avoid any aliasing phenomenon (and to increase the overlapping area between images). On the other hand, the SLAM is a more sofiscated approach that constructs a local map according to the navigation of the robot in the environment. Within this map, it provides the estimation of the robot's position. The implementation of visual approaches for both the odometry and SLAM (often called as vOdometry and vSLAM, respectively) usually resorts to feature-based analysis to increase the performance and, as a consequence, the frame rate of the output data from the visual system. With the aim to simulate the human vision it is possible to use stereo cameras to acquire the 3D information from the environment. It should be highlighted that non-structured and dynamic environments usually impose severe challenges for visual-based techniques and, therefore, the detection of revisited areas is a key point for the navigation stack of mobile robots since it decreases the positioning error. Considering the high number of the implementation of techniques available in the literature, the current research had selected the most promising one by considering factors such as, performance expectation reported in scientific articles, public availability of the methods and other particularities (robustness of features, internal assumptions and others). A comparative analysis is conducted by taking into consideration a set of three monocular odometry methods and three stereo-based SLAM methods.

For odometry method three algorithms were selected, namely mono-vo, viso2 and mORB-SLAM. **Mono-vo** [2] implementation was developed in 2015 and it is based on OpenCV. Uses the FAST for features detection and the Kanade-Lucas-Tomasi to search the correspondence in the next image. Incorporates a mechanism that searches for new features and uses an outliers removal mechanism. This implementation aims the use the scale information from an external source of data and, therefore, it is possible to correct the previous estimations. Moreover, the mono-vo follow a heuristic to estimate the forward motion as the dominant motion. The **viso2** [3] implementation was developed in 2011 and calculates the camera position estimation using a set of rectified images. The method has available a large number of configurable parameters which increases the flexibility, but turns the method very difficult to setup. Frequently, a motion estimation system cannot estimate with a metric scale from monocular sequences. Thus, viso2 assumes that the camera motion follows a fixed and known height from the ground (used to predict the scale factor). This method uses a bucketing technique to the correct distribution of features in images however, it has a relevant limitation related to pure rotations which degradates the estimation. The **mORB-SLAM** is the monocular implementation of the ORB-SLAM presented in [4]. Therefore, this method uses keyframes and ORB as features extractor, detecting corners from the FAST and BRIEF descriptor, to ensure the (soft) real-time capacity. A bundle adjustment is conducted with a new keyframe in order to remove some erroneous estimations (and features) and provides a better positioning.

For SLAM technique three algorithms were selected, namely RTAB-Map, S-PTAM and ORB-SLAM2. The **RTAB-Map** [5] system was developed in 2014 to capture a Graph-Based SLAM implementation and to present an incremental approach for loop closure detection. It is important to note that the calculation of the egomotion, with the own method of odometry called "s_odom", presents limitations in situations comprising "Empty Space Environments" (when the features are a distance to the camera larger than 4 m). This means that, the performance is affected by image sequences presenting large egomotion (reduces the feature matching in consecutive frames) however, it is possible to use other visual odometry methods to solve this limitation such as, viso2. The RTAB-Map was particularly developed for scenarios involving cars and based on two cameras with large focal distances. Although being used in different robotic applications, the method only estimates a new position when 6DOF motion is detected between consecutive frames. The loop closure detection is constructed online through a bag-of-words approach (with SURF descriptors). A Graph Optimization approach is used to correct the map during the robot navigation. Considering that mapping large-scale environments during long-terms navigation paths is constrained by the computational power available onboard. The RTAB-Map implements a memory management approach that considers only part of the map to fulfill online processing requirements of todays applications. The **S-PTAM** [6] implementation was developed with the goal to obtain a stereoscopic system able to help the robot navigation, by providing more reliable estimations. Divides the

SLAM-based approach into two tasks: Tracking and Map Optimization. Uses the BRIEF descriptor with binary features to reduce the storage requirements and speedup the feature correspondence. The Shi-Tomas algorithm imposes a good spatial distribution of the features. Like the ORB-SLAM, the S-PTAM uses a keyframe-based approach to estimate the motion. During the creation of map, the method adjusts the nearby points by excluding those points that are considered erroneous. This task is presented as a maintenance process independently of the Tracking, which is an advantage in terms of the processing time. The **ORB-SLAM2** [4] implementation was developed in 2015 and it is able to use monocular, stereo and RGB-D cameras. It is a feature-based approach that uses the ORB extractor because of time constraints (important in the real-time applications). Therefore, the egomotion determination is characterized by a reliable motion estimation, since it is invariant to view point and illumination changes (FAST corners with BRIEF descriptors). In terms of motion estimation, it is keyframe-based approach and avoids an excessive computational demand. Allows a camera relocalization in real-time when the Tracking process was lost, by following a bag-of-words approach. Uses the Covisibility Graph concept to bring the possibility of adding new keyframes and, consequently, to obtain an environment ideal reconstruction. The Covisibility Graph helps the closure loop detection, because this detection can be achieved by a similarity measure between bag-of-words vector and all neighbors of the Covisibility Graph. Finally, an Essential Graph provides a real-time effective loop closure since it maintains the words that represent a strong match (assuming a vocabulary constructed offline using the DBoW2 library [7]).

3 Comparative Analysis

Two set of experiments were conducted in this section. The first aims to provide evidences about the accuracy of egomotion estimation considering monocular images sequences - odometry trials. The second aims to provide a comparative study of different vSLAM-based approaches without any constraint about the environment or navigation path. A simple but effective comparative analysis is performed by considering all techniques introduced in the previous section. During the analysis some public datasets were considered to allow replicability of results by other scientific works (when proposing other visual-based methods). The performance of the methods are discussed by taking into consideration some metrics, namely the Central Processing Unit (CPU) utilization in percent, processing time and normalized Euclidian error between the ground-truth and estimated trajectories in the same conditions, for example data acquisition rate and image size. Regarding the results obtained in the monocular odometry, a normalization of trajectories are conducted due to the unknown scale factor between different methods. Moreover, the images sets that were choosen represent paths that do not evidence revisited areas since the analysis is focused on the accuracy of the egomotion estimation. Following a similar methodology, vSLAM techniques discussed in the previous section are evaluated by taking

into account the same metrics used in the odometry evaluation. In this case, the detection of the closure loop is also contemplated since it is a key feature for all SLAM techniques. Quantitative analysis can be retrieved by measuring the accuracy of each SLAM technique in two checkpoints (since not all techniques provide estimatives in all frames), represented by "error_point1;error_point2" in the Tables 4, 5 and 6. Qualitative discussion is made using graph representations of the trajectories. A very relevant phenomenon that it is usually ignored by the literature is the aliasing. This research discussed this phenomenon during the trials which means, the influence of the processing time in the overall accuracy of each method is investigated. To evaluate the methods were selected three test scenarios, that represent indoor and outdoor environments. These environments show a clear image of the behavior that will be expected for each technique. The three public datasets comprise KITTI, MIT Stata Center and New College.

The **KITTI**[1] dataset is composed of 22 stereo images sequences with different trajectories obtained, in urban and freeway environments, see Fig. 1(a). The height of the camera in relation to the ground and the no-oscillation have been taken into account. The camera calibration parameters are available as well as the ground-truth of the trajectory made during the acquisition of high resolution image sequences. The **MIT Stata Center**[2] is an indoor dataset, obtained from a robot, see Fig. 1(b). This dataset was made to support the development of visual SLAM algorithms and, therefore, the trajectories are longer and present various direction changes. The **New College**[3] dataset provides data that was acquired in gardens, see Fig. 1(c). All data is synchronized: images, laser, GPS and IMU information and odometry data (ground-truth) are available.

Fig. 1. Illustrative example of the used datasets: (a) KITTI (b) MIT Stata Center (c) New College

Table 1 presents a summary of the testing conditions and scenarios evaluated to each technique.

[1] Dataset available on http://www.cvlibs.net/datasets/kitti/eval_odometry.php.
[2] Dataset available on http://projects.csail.mit.edu/stata/downloads.php.
[3] Dataset available on http://www.robots.ox.ac.uk/NewCollegeData/.

Table 1. Test Conditions to methods evaluation

	Method	ROS embedded	Dataset	PC (Ubuntu 16.04)
Odometry	Mono-vo Viso2 mORB-SLAM	No	KITTI + New College	12 GiB RAM i7-4720HQ @ 2,60GHz x 8 SSD 128 GB
SLAM	ORB-SLAM	No	KITTI + MIT Stata Center	16 GiB RAM DDR4 i7-6700HQ @ 2,60 GHz x 8 SSD 240 and 512 GB
	S-PTAM	Yes		Virtual machine: (Ubuntu 14.04) SSD 512 GB 7 GiB RAM 4 CPU
	RTAB-Map			

3.1 Results

Odometry

KITTI – Sequence 07. As visible in the Fig. 2, the mono-vo implementation follows the movement of the camera for most of the time. However, the incorrect detection of a direction change caused a little error between the real and estimated trajectories.

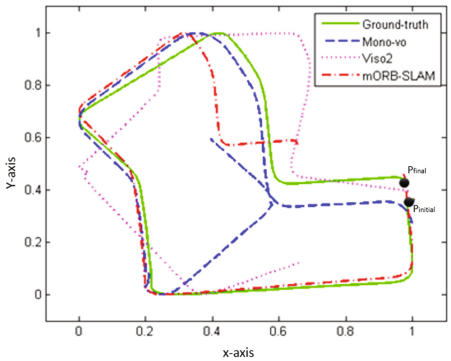

Fig. 2. Normalized trajectories obtained by KITTI dataset (07)

On the other hand, mORB-SLAM implementation estimates the camera position correctly, but with lower error, since it captures all direction changes with a satisfactory accuracy. It can be noticed in Table 2 that the mORB-SLAM presents the better egomotion estimation but it takes longer processing time (in average)[4]. In the majority of cases, the mono-vo implementation captures

[4] KFr represents the number of keyframes used to egomotion estimation.

the egomotion during for a large part of the trajectory made by the observer, however the largest maximum error was caused by a wrong detection of one direction change. The viso2 implementation presents a higher error, because of the deviations of the first positions.

Table 2. Comparison of the normalized trajectories obtained by KITTI dataset (07)

	Processed frames	Normalized error			Time	CPU
		Maximum	Average	Std		
Mono-vo	1000	0,64	0,13	0,17	85 s	>35% max. = 40%
Viso2	999	0,62	0,32	0,18	**74 s**	**>12% max. = 15%**
mORB-SLAM	996 $KFr = 374$	**0,36**	**0,10**	**0,10**	109 s	>18% max. = 20%

New College. The mORB-SLAM and mono-vo implementations try to follow the circular motion of the observer, see Fig. 3.

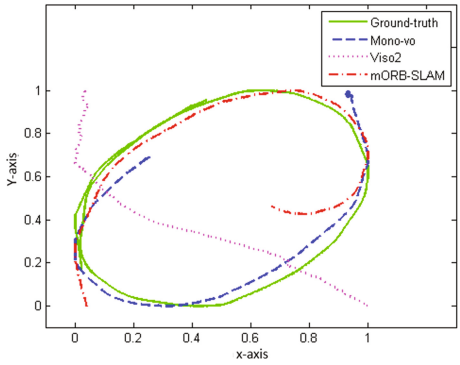

Fig. 3. Normalized trajectories obtained by New College dataset

The viso2 provides an incorrect estimation, that can be justified by some oscillations in the camera during the motion of the observer as well as the relative changes of depth, according to the literature. Therefore, the performance of viso2 is severely affected by these conditions (which reduces its robustness and reliability). From the Table 3, it is visible that, although all implementations present errors, the mORB-SLAM is the best. However, neither this implementation nor mono-vo can obtain the final position intended.

In terms of the processing time, the mono-vo implementation presents better results, even using more frames but with a slightly higher CPU usage.

Table 3. Comparison of the normalized trajectories obtained by New College dataset

	Processed frames	Normalized error			Time	CPU
		Maximum	Average	Std		
Mono-vo	2500	1,22	0,61	0,26	127 s	>28% max. = 34%
Viso2	1765	1,11	0,85	0,19	**124 s**	**>13% max. = 18%**
mORB-SLAM	1657 $KFr = 293$	**0,69**	**0,37**	**0,16**	273 s	>19% max. = 24%

SLAM

KITTI – Sequence 05. According to the Fig. 4, it is possible to observe that the ORB-SLAM2 implementation is the only one that estimates all camera positions, detects the loops and adjusts its trajectory. It should be noticed that, as expected, the RTAB-Map system (with the viso2 providing the egomotion estimation) try to replicate the motion made by observer however, there are deviations. These deviation could be justified by the susceptibility of the method in situations with inclination changes or even with camera rotations.

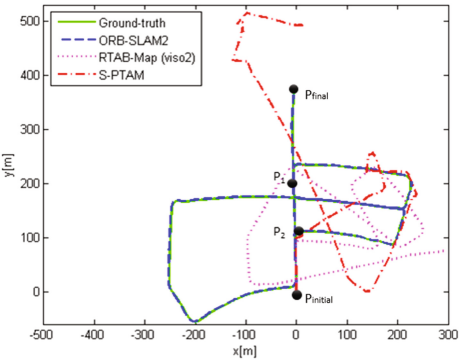

Fig. 4. Trajectories obtained by KITTI dataset (05)

Taken into account the results presented in Table 4, it is safe to say that ORB-SLAM2 lead to lower errors between the real trajectory and ground-truth. In terms of the CPU utilization and processing time, the ORB-SLAM2 implementation presents higher values comparatively with the S-PTAM implementation.

KITTI – Sequence 09. Figure 5 depicts that ORB-SLAM2 has some drifts during the estimation of the trajectory in this sequence. Moreover, the technique do not detect any revisited area and, as a consequence, the trajectory was not adjusted by the closure loop detection mechanism. In relation to the other implementations, is not possible to conclude about the effectiveness of closure loop, once the estimated final position was not close enough of the initial position (circular path). One relevant issue was the number of frames that were not

Table 4. Comparison of the trajectories obtained by KITTI dataset (05)

		ORB-SLAM2	RTAB-Map		S-PTAM
			viso2	s_odom	
Processed frames		2761	1224	2745	1916
Error	Maximum	**1,64 m**	420,91 m	—	123,29 m
	Average	**0,55 m;1,05 m**	30,84 m;17,73 m	—	33,05 m;8,41 m
Processing time		5 min 32 s	4 min 57 s	2 min 19 s	**4 min 43 s**
CPU		**47%**	57%	28%	54%

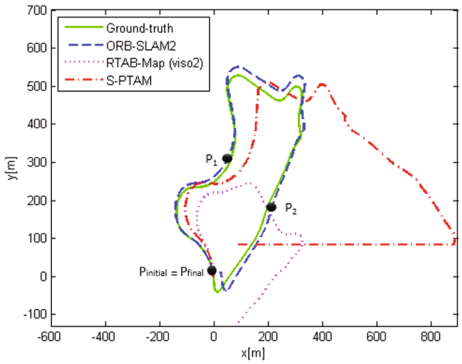

Fig. 5. Trajectories obtained by KITTI dataset (09)

considered by the SLAM techniques, which lead to aliasing situation - and some motion components were not captured by these visual techniques. Although the other implementations try to follow the camera motion, they get lost along the trajectory.

Table 5 demonstrates that the ORB-SLAM2 has the ability to characterize the observer motion with better accuracy and reliability (no aliasing phenomenon because the entire image sequence was processed). The CPU utilization and processing time have higher values, but it uses all frames to trajectory estimation. On the other hand, the RTAB-Map (viso2) does not estimate the realistic path, being difficult to provide a quantitative analysis of the performance of this method.

MIT Stata Center. Figure 6 shows that RTAB-Map, with the odometry incorporated directly by the implementation, is not able to provide data (lack of inliers). It is important to emphasize that were modified some parameters to the features extraction but without changes in the obtained results. This fact can be explained because of the limitation of this method in indoor environments, in particular the presence of large homogeneous spaces (halls), represent a challenging problem for this method.

Table 5. Comparison of the trajectories obtained by KITTI dataset (09)

		ORB-SLAM2	RTAB-Map		S-PTAM
			viso2	s_odom	
Processed frames		1591	648	1557	884
Error	Maximum	**28,87 m**	145,14 m	—	115,06 m
	Average	**11,46 m;37,62 m**	—;207,65 m	—	60,61 m;103,25 m
Processing time		3 min 4 s	2 min 49 s	1 min 20 s	**2 min 40 s**
CPU		**36%**	55%	30%	80%

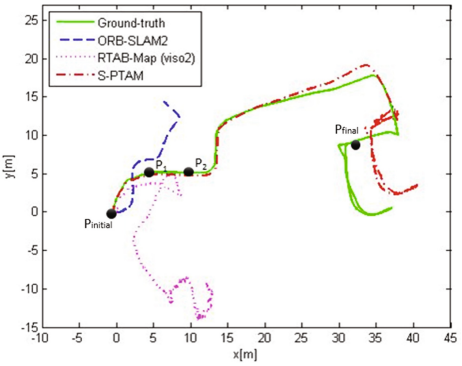

Fig. 6. Trajectories obtained by MIT Stata Center dataset

The S-PTAM implementation estimates correctly the trajectory since the loops are correctly detected and, therefore, the method redefines the trajectory taken by the observer. In fact, this technique was able to estimate the entire path with a realistic scale. The RTAB-Map implementation (viso2) tries to follow the direction changes occured during the path. The result of RTAB-Map (viso2) shows that the path has suffered from a wrong estimation at the beginning of the sequence (y-axis) which caused a total deviation. This fact can be explained by the existence of a high and fast rotation at the start of the trajectory which clearly demonstrates that this method is quite susceptible to errors in these situations. All methods have demonstrated similar CPU usages (see Table 6), however, ORB-SLAM2 had the best performance in this metric. Although the best CPU usage, the ORB-SLAM2 did not have the best accuracy since the localization was lost for a while (during a transition to a darker area) and the method was not able to detect the loop closure. Thus, it is possible to conclude that was calculated wrongly the first direction change and, consequently, it is quite difficult to characterize the error. The RTAB-Map (viso2) system was also unable to determine the loop closure, because discards many frames, possibly due to the lack of the 6DOF motion, between consecutive frames.

Table 6. Comparison of the trajectories obtained by MIT Stata Center dataset

		ORB-SLAM2	RTAB-Map	S-PTAM
			viso2	
Processed frames		2312	2730	2962
Error	Maximum	38,92 m	26,95 m	**2,57 m**
	Average	6,30 m;—	3,29 m;14,67 m	**8,87 m;0,34 m**
Processing time		4 min 47 s	4 min 23 s	**4 min 10 s**
CPU		**33%**	53%	68%

4 Conclusion

This article studies several visual-based techniques. It presents a comparative analysis of odometry and SLAM approaches in realistic indoor and outdoor scenarios. Moreover, a quantitative and qualitative discussion is presented by taking into account several metrics such as, graphic representation of the navigation path, processing time and the aliasing phenomenon (considering real-time constraints). This phenomenon causes positions error along the trajectory, once are lost some frames between consecutive samples.

The results showed that the mono-vo follows relatively well the motion of the majority of the trajectories, but any motion detected provide a new position even when the camera does not move, which causes an increase in the error. The viso2 presents a good motion estimation always that detects 6DOF, however it provides erroneous estimations when the camera presents oscillations or changes its height in relation to the ground. The mORB-SLAM also generates good results in most cases, The mORB-SLAM also generates good results in most cases, with errors lower in relation to the others (approximately 45%). Thus, the mORB-SLAM and viso2 are the most complete with the principal difference in the requirement by the mORB-SLAM estimator of a vocabulary constructed à prior. In the case of the SLAM implementations, the ORB-SLAM2 presents, in the majority of the cases, a good motion estimation and provides estimations with lower errors (decrease more than 80%). It is important to reinforce that the RTAB-Map with own odometry method is difficult to parametrize and highly dependent on the environment, such as in the case of the KITTI dataset that can be possible to considered a “Empty Space Environment”. The S-PTAM is suitable only for MIT Stata Center dataset and, in this case, it was the only one that provided correct results. This fact can be explained by the higher time between images input in relation to the KITTI dataset. Thus, it is notable that the S-PTAM and ORB-SLAM2 are the most adequate. These implementations are differentiated by the fact of the S-PTAM does not present an approach for loop closure detection, but only Bundle Adjustment, which does not provide results so good to known revisited areas.

To future work, the authors will conduct novel datasets (and incorporate these datasets in the discussion), including new environments, to support the scientific community and intend to reinforce the study namely with other methods.

Acknowledgements. This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project ≪POCI-01-0145-FEDER-006961≫, and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

1. Pinto, A.M., Moreira, A.P., Correia, M.V., Costa, P.: A flow-based motion perception technique for an autonomous robot system. J. Intell. Robot. Syst. **75**(3), 475–492 (2014). doi:10.1007/s10846-013-9999-z
2. Singh, A.: An OpenCV based implementation of Monocular Visual Odometry. Indian Institute of Technology Kanpur. Technical report, Kanpur (2015)
3. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In: IEEE Intelligent Vehicles Symposium. University of California, San Diego, CA, USA, pp. 486–492 (2010)
4. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015). doi:10.1109/TRO.2015.2463671
5. Labbé, M., Michaud, F.: T Online global loop closure detection for large-scale multi-session graph-based SLAM. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2014). doi:10.1109/IROS.2014.6942926
6. Pire, T., Fischer, T., Civera, J., Cristóforis, P., Berlles, J.J.: Stereo parallel tracking and mapping for robot localization. In: Intelligent Robots and Systems, pp. 1373–1378 (2015). doi:10.1109/IROS.2015.7353546
7. Galvez-Lopez, D., Tardós, J.D.: Bags of binary words for fast place recognition in images sequences. Intell. Robots Syst. **28**(5), 1188–1197 (2012). doi:10.1109/IROS.2012.2197158