Camera Localization in Outdoor Garden Environments Using Artificial Landmarks

Strisciuglio, Nicola; Vallina, Maria Leyva; Petkov, Nicolai; Salinas, Rafael Munoz

Published in:
2018 IEEE International Work Conference on Bioinspired Intelligence, IWOBI 2018 - Proceedings

DOI:
10.1109/IWOBI.2018.8464139

Publication date:
2018

Camera localization in outdoor garden environments using artificial landmarks

Nicola Strisciuglio, María Leyva Vallina, Nicolai Petkov
Johann Bernoulli Institute for Mathematics and Computer Science
University of Groningen, The Netherlands

Rafael Muñóz Salinas
Computing and Numerical Analysis Department
University of Cordoba, Spain

***Abstract*—In this paper, we present an outdoor monocular camera localization system based on artificial markers and test its performance in one of the test gardens of the TrimBot2020 project, in Wageningen. We use ArUco markers to construct a map of the environment and to subsequently localize the camera position within it. We combine the localization algorithm based on ArUco with a Kalman filter to smooth the trajectory and improve the localization stability with respect to fast movements of the camera, and blurred or noisy images. We recorded two sequences, with resolution 480p and 1080p respectively, in the TrimBot2020 garden. We compare the localization performance of ArUco with a keypoint-based approach, namely ORB-SLAM2. We analyze and discuss the strengths and problems of both marker- and keypoint-based approaches on the considered sequences. The performed comparison suggests that the two approaches might be fused to jointly improve re-localization and reduce the drift in pose estimation.**

I. Introduction

Camera pose estimation is a common problem in many applications requiring precise localization in the environment, such as augmented and virtual reality [1], [2], mobile robotics [3]–[7] and autonomous driving [8], [9]. Obtaining the camera pose from images requires finding the correspondences between known points in the environment and their camera projections. While some approaches detect and extract natural features such as key points or textures [10], [11], fiducial markers are an attractive approach because they are easy to detect and allow to achieve high speed and precision [12]–[16]. They, however, require manual installation of artificial landmarks in the environment. Lately, end-to-end pose learning algorithms based on Convolutional Neural Networks (CNN) have also been proposed [17], [18].

Using keypoint-based features requires no intervention in the environment, but it has several limitations. First, a computationally intensive processing is required for methods based on natural features in order to estimate the location. Second, they suffer in texture-less areas (such as corridors) but also in environments showing repetitive patterns (such as office-like buildings) and in environments reach of texture, such as gardens. In addition, it is well known that they tend to fail in case of blurring due to fast camera movements. Therefore, when precision, robustness and speed are a strict requirement, the use of artificial landmarks placed in the environment is a good choice.

Squared planar markers are a popular tool for fast, accurate and robust camera localization. These types of marker are composed of an external black border and an internal code (most often binary) to uniquely identify them. Their main advantage is that the corners of a single marker can be employed for camera pose estimation [12]. Despite its advantages, large-scale mapping and localization from planar markers is a problem scarcely studied in the literature. The main challenge when dealing with squared planar markers is the ambiguity problem [19], which states that, although in theory it should be possible to accurately obtain the pose from the four corners of one single planar square marker, in practice, due to noise in the localization of the corners, two solutions appear, and in some cases, it is impossible to discern the correct one. In [20], the authors propose an off-line method to create a map of squared planar markers able to deal with the ambiguity problem. To build the map, it is only necessary to print markers with a regular printer, place them in the desired environment so as to cover the working area, and to take pictures. The only requisite is that each picture has to contain at least two markers. This is needed in order to estimate the pairwise spatial relationships between them. Then, the method computes a map with the pose of the markers by analyzing the images. The method can employ an unlimited number of high-resolution images in order to achieve very high accuracy since no time restrictions are imposed. Afterward, camera localization can be done, in the correct scale.

In this work, we apply a localization method based on ArUco planar markers [12] in a new, outdoor environment, within the scope of the TrimBot2020 project [21]. The goal of the project is to prototype a gardening robot, that, among other things, has to be able to navigate autonomously. Thus, an accurate system for camera localization is required. Current approaches are challenged and in many cases fail when dealing with repetitive or texture-less patterns [22]. Although gardens are highly textured environments, those textures are highly repetitive (i.e. grass, pebble stones, fences, and similar), and thus, difficult to extract meaningful interest features from. Also, as outdoor environments, they are largely affected by illumination variations due to changes of light and weather conditions. In order to overcome this challenges, we placed artificial markers in the garden, and performed localization based on them.

We studied the performance of marker- and keypoint-based approaches for camera localization on two sequences recorded in the TrimBot2020 test garden. We analyzed their respective

strengths and weaknesses, in order to evaluate and discuss those cases in which the joint use of keypoints and artificial markers might contribute to increase of localization accuracy and to improve re-localization and correct drift.

This paper is structured as follows. In Section II, we explain in depth the applied method, as well as the building of the marker map and the localization process. We provide details on the experiments and evaluation procedure in Section III, and present and discuss the obtained results in Section IV. Finally, we draw conclusions in Section V.

II. Methods

We employ a method for camera localization based on a map of ArUco fiducial markers [12], [20], which we explain in the following of the section.

A. ArUco

ArUco is an OpenSource library written in C++ for detecting and identifying squared fiducial markers in images that works at 500fps in images with a resolution of 640×480 pixels using a single thread. The main steps for marker detection and identification are depicted in Figure 1. Given an input image I (Fig. 1a), the following steps are performed:

- Image segmentation (Fig. 1b). Since the designed markers have an external black border surrounded by a white space, the borders can be found by segmentation. In the original detection approach, a local adaptive method is employed: the mean intensity value m of each pixel is computed using a window size w_t. The pixel is set to zero if its intensity is greater than $m - c$, where c is a constant value. This method is robust and obtains good results for a wide range of values of its parameters w_t and c.
- Contour extraction and filtering (Figures 1(c,d)). The contour following algorithm of Suzuki and Abe [23] is employed to obtain the set of contours from the thresholded image. Since most of the contours extracted correspond to irrelevant background elements, a filtering step is required. First, contours too small are discarded. Second, the remaining contours are approximated to its most similar polygon using the Douglas and Peucker algorithm [24]. Those that do not approximate well to a four-corner convex polygon are discarded from further processing.
- Marker code extraction (Figs. 1(e,f)). The next step consists in analyzing the inner region of the remaining contours to determine which of them are valid markers. Perspective projection is first removed by computing the homography matrix, and the resulting canonical image (Fig. 1e) is thresholded using the Otsu's method [25]. The binarized image (Fig. 1f) is divided into a regular grid and each element is assigned a binary value according to the majority of the pixels in the cell. For each marker candidate, it is necessary to determine whether it belongs to the set of valid markers or it is a background element. Four possible identifiers are obtained for each candidate, corresponding to the four possible rotations of the canonical image. If any of the identifiers belongs to the set of valid markers, then it is accepted.
- Subpixel corner refinement. The last step consists in estimating the location of the corners with subpixel accuracy. To do so, the method employs a linear regression of the marker contour pixels. In other words, it estimates the lines of the marker sides employing all the contour pixels and computes the intersections. This method, however, is not reliable for uncalibrated cameras with small focal lenses (such as fisheye cameras) since they usually exhibit high distortion.

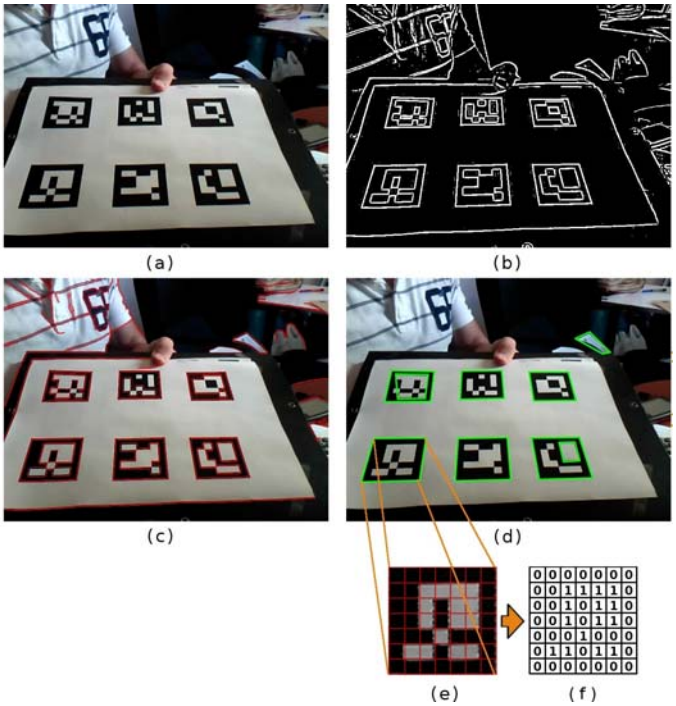


Fig. 1: **Detection and identification pipeline of ArUco library.** (a) Original image. (b) Contour image thresholded using an adaptive method. (c) Contours extracted. (d) Filtered contours that approximate to four-corner polygons. (e) Canonical image computed for one of the squared contours detected. (f) Binarization after applying Otsu's method.

B. MarkerMap

As previously indicated, a recent work on mapping with fiducial markers [20] allows extending the use of squared markers to larger environments. The employd method is named MarkerMap. While the general approach consists in using a single marker, or at most, a small set of them for which their relative pose is known beforehand (e.g., a set of markers in a printed piece of paper), the method proposed in [20] allows estimating the 3D relative pose of a set of markers freely placed in the environment. The user is required to only take a set of pictures of the markers installed in the environment to map from several viewpoints.

MarkerMap works as an off-line process. First, all the images are analyzed in order to create an initial connection quiver representing the pairwise poses between the observed markers. Only images in which at least two markers are detected are used for the construction of the quiver. Between each pair of markers, we may have more than one possible estimations of their relative pose (as many as images showing both markers). Then, the first problem becomes to decide which one of them is the valid relationship. One has to consider that estimation of the marker poses is subject to the ambiguity problem [19]. Thus, a method able to disambiguate invalid detections is employed by a consensus-based method that minimizes the global reprojection error in the images involved. As a result of the first phase, a refined graph is obtained representing the relative pose between the markers. However, this graph is yet subject to imprecision. So, a pose graph optimization method is run in order to ensure that all cycles in the graph are coherent. The resulting graph can be employed to establish an initial estimation of the relative pose of the markers that is further refined using a sparse bundle adjustment approach by minimizing the total reprojection errors.

C. Localization

Localization is treated as a non-linear optimization process minimizing the reprojection error of the observed markers in the given frame f^t, using the previous pose γ^{t-1} as starting point.

Let $\mathcal{M}$ be the set of available markers, and let us assume that all markers have the same size s. The four marker corners can be expressed w.r.t. one of them as:

$$\begin{array}{llll} c_1\text{=(} & 0, & 0, & 0\ \text{)}, \\ c_2\text{=(} & 0, & s, & 0\ \text{)}, \\ c_3\text{=(} & s, & s, & 0\ \text{)}, \\ c_4\text{=(} & s, & 0, & 0\ \text{)}. \end{array} \tag{1}$$

We shall denote by

$$\omega_m^t = \{u_{m,l}^t \mid u \in \mathbb{R}^2, l = 1 \dots 4\} \tag{2}$$

the pixel locations in which the four corners of marker m (Eq. 1) are observed in frame f^t.

Then, the localization problem can be enunciated as obtaining the pose γ^t that makes the reprojection error of the marker corners minimal. We shall define

$$e_m^t(\gamma) = \sum_{l=1}^{4} \left[\Psi(\gamma, c_l) - u_{m,l}^t\right]^2, \tag{3}$$

as the reprojection error of marker m given the pose γ, where Ψ is the function that projects a point c_l on the camera image.

Then the optimal pose γ^t is obtained as:

$$\gamma^t = \underset{\gamma}{\operatorname{argmin}} \sum_{m \in \mathcal{M}(t)} e_m^t(\gamma \cdot \gamma_m), \tag{4}$$

where $\mathcal{M}(t)$ represents the set of markers visible in f^t, and γ_m the pose of marker m. Finally, the Levenberg-Marquardt's (LM) algorithm [26] is used for the minimization of Eq. 4 using γ^{t-1} as initial solution.

Kalman filter The localization algorithm based on fiducial markers is subject to errors caused by the marker pose ambiguity problem or the difficult accurate detection of markers in blurred images (i.e. due to fast camera movements). Hence, the traced camera trajectory can be noisy and the localization not precise.

In order to smooth the computed camera trajectory and improve the robustness of camera localization to noisy detection over time, we employ a Kalman filter that takes as input the camera poses estimated by ArUco. The Kalman filters uses a series of noisy measurements over time and produces estimates of unknown variables (i.e. the state of the system) that are more precise than the estimates based on a single measurement. We use a Kalman filter to reduce the eventual pose estimation error on the basis of the pose estimated in the previous frames.

III. Experiments

A. Test garden

We carried out experiments in the test garden of the TrimBot2020 project [21] in the campus of Wageningen University and Research. In Fig. 2, we show some views of the test environemnt. The garden is of size 18×20 meters, approximately, with a double fence to ensure safety when a bush cutting robotic arm is operational. The garden contains different objects, such as boxwoods, rose bushes, trees, and different terrains, e.g. grass, pebble stone and woodchips. Some of the topiary bushes are placed on top of a slope of 10°. The garden is constantly subjected to modifications due to plant growing, seasonal changes and varying weather conditions, making it difficult to find stable landmarks for localization.

B. Environment mapping

We installed a set of ArUco markers in the test environment, as shown in Fig. 3, and mapped their position in the garden by using MarkerMap. The resulting map of markers is metrically consistent with the environment and allows for camera pose

(a) (b) (c) (d)

Fig. 2: Test garden of the TrimBot2020 project at the Wageningen University and Research campus (The Netherlands).

(a) (b) (c) (d)

Fig. 3: Pictures of the garden with installed ArUco markers.

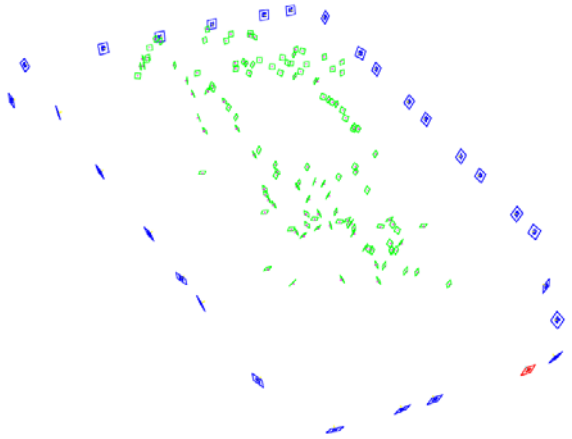


Fig. 4: Marker map reconstructed in the Wageningen garden, which we use as reference for localization of the camera pose. We used 29 markers to map a garden environment of approximately 18×20 meters.

estimation and localization at real-scale. In Fig. 4, we show the marker map reconstructed in the Wageningen garden, which includes 29 markers and that we use for the camera localization experiments. In order to construct the map of markers, we took 126 pictures in the garden with a smartphone. Each image contains at least two markers in order to take into account their relative positions in the MarkerMap reconstruction algorithm. The locations from which the pictures were taken are indicated with the green elements in Fig. 4. It is possible to perform the mapping using only a smartphone, whose camera is calibrated and that efficiently runs the reconstruction algorithm on board.

C. Evaluation

We evaluate the performance of the localization in terms of Absolute Trajectory Error (ATE) [27], and the translational (T) and rotational (R) errors [8]. The ATE is a measure of the trajectory estimate error with respect to a ground truth trajectory. The computation of the ATE first aligns the two trajectories and then evaluates directly the pose differences. The translational and rotational errors are a measure of the percentage of pose difference at one meter steps.

We evaluate the performance of the localization algorithm based on ArUco markers on two sequences recorded in the Wageningen garden. The sequences have resolutions 480p (garden-seq1) and 1080p (garden-seq2), respectively. For each sequence, we performed two rounds of evaluation using the camera trajectory computed by ORB-SLAM2 [28] and by ArUco extended with a Kalman filter as the baseline trajectory, respectively.

Since the ground truth is not available for the recorded sequences, in this paper we perform a qualitative evaluation of the methods by setting, in turn, one of them as baseline and evaluating the others against it. This study serves as carrying out a preliminary analysis of performance and complementarity of marker- and keypoint-based camera localization approaches in garden environments.

IV. Results and Discussion

In Table I and Table II, we report the performance measurements that we recorded on the sequences garden-seq1 and garden-seq2, respectively.

The use of a Kalman filter contributes to smoothing the trajectory estimated by the frame-wise localization algorithm based on ArUco markers. This results in a reduction of the localization errors, when the estimated trajectories are compared to the trajectory of ORB-SLAM2 taken as the baseline. The absolute trajectory error (ATE) is indeed reduced by 14.8% on the garden-seq1 sequence and by 8.34% on the garden-seq2 video, by using a kalman filter with an averaging window of 50 frame size. In the computation of the results, we take into account only those trajectory points that are present in both baseline and test sequence. We report in the last column of Table I and Table II the percentage of the baseline frames matching the test sequence and that we take into account during the computation of results.

We compute the localization accuracy also taking the trajectory computed by ArUco with Kalman filter as the baseline. This test serves as an evaluation of the complementarity of keypoint- and marker-based localization approaches. On one hand, in the garden-seq1, the higher ATE obtained by ORB-SLAM2 w.r.t. the one of ArUco is mostly due to a drift of the localization performed by ORB-SLAM2, which is noticeable in the last part of the trajectory in Fig. 5a. On the other hand, on garden-seq2 video, ORB-SLAM2 obtained

Results on garden-seq1 (640p)

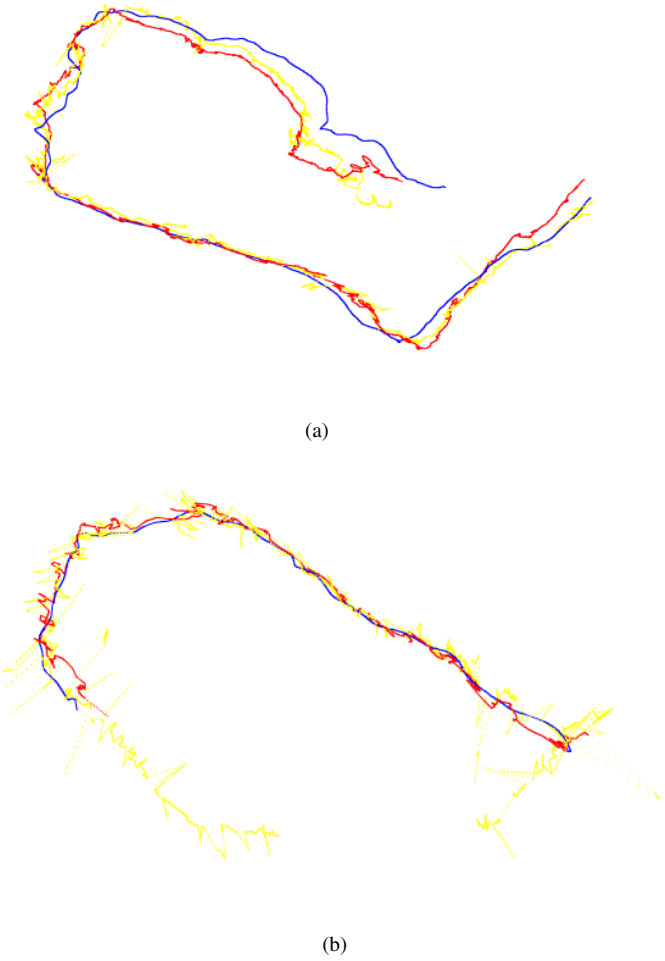
	ATE	T	R	% frames
	ORB-SLAM2 Baseline			
ArUco	0.133915	2.88699	0.247464	92.3%
ArUco+KF	0.114052	2.72413	0.464687	87.89%
	ArUco+KF Baseline			
ArUco	0.147281	0.652293	0.356804	100%
ORB-SLAM2	0.468451	0.617638	0.296056	100%

TABLE I: Localization performance obtained on the garden-seq1 video (of resolution 480p), using the trajectory of ORB-SLAM2 (upper rows) and of ArUco with Kalman Filter (lower rows) as the baseline.

Results on garden-seq2 (1080p)

	ATE	T	R	% frames
	ORB-SLAM2 Baseline			
ArUco	0.0460907	3.707031	0.277564	70.07%
ArUco+KF	0.0422443	3.29234	0.278	70.07%
	ArUco+KF Baseline			
ArUco	0.266403	1.3307	0.561751	100%
ORB-SLAM2	0.206727	0.645569	0.247646	73.36%

TABLE II: Localization performance obtained on the garden-seq2 video (of resolution 1080p), using the trajectory of ORB-SLAM2 (upper rows) and of ArUco with Kalman Filter (lower rows) as the baseline.



(a)

(b)

Fig. 5: Trajectories computed by ArUco (yellow line), ArUco with Kalman filter (red line) and ORB-SLAM2 (blue line) on (a) garden-seq1 and (b) garden-seq2 videos.

higher localization accuracy than ArUco. As it can be seen in Fig. 5b, the trajectory computed by the ArUco camera localization system is noisy. This is mainly cause by fast camera movements and large blurring effects, which make accurate detection of markers difficult. Furthermore, the presence of only one marker in some frames causes the ambiguity problem that corrupts the correct localization output. However, the smoothing performed by the Kalman filter produces a better trajectory.

The ORB-SLAM2 algorithm necessitates of an initialization step, usually performed on the initial part of the test sequence, and subsequently constructs a map of the environment. This process does not allow the computation of the trajectory for the whole sequence and reduces the number of frames on which the evaluation can be performed. ArUco does not need an initialization algorithm, apart from the map construction process which is performed off-line. However, ArUco is not able to localize the camera and estimate its pose if markers are not appropriately present in the acquired scene and detected. In order to improve the localization using ArUco markers, a multi-camera system can be employed, with a calibrated rig of cameras looking at different directions. In such a case, if no markers are visible in the scene acquired by one camera, it is likely to detect markers in other camera views, so not loosing track of the mobile agent position in the environment. The main advantages of using the proposed method over ORB-SLAM2 are two. First, our proposed approach is less computationally demanding, requiring a single thread to achieve real-time processing. Second, the position estimated by our method is in the correct scale, while the positions estimated by monocular ORB-SLAM2 are scale agnostic and not consistent along the sequence. In consequence, ORB-SLAM2 can not be used in a realistic scenario for tracking using monocular cameras.

The experiments that we performed suggest that marker- and a keypoint-based localization can be performed jointly, in order to gain from the strengths of both approaches. The marker based approach can be used for initialization and for reducing the drift while estimating the camera pose with a keypoint-based approach. Furthermore, re-localization is a difficult process to be performed with keypoint-based algorithms, and which can be simplified by the presence of unique landmarks. A combined keypoint-marker approach would require

to install a lower number of markers in strategic positions of the environment (e.g. on parts with repetitive patterns) and can, thus, improve localization and re-localization accuracy.

V. Conclusions

In this paper we presented an experimental study on camera localization in an outdoor environment, namely the test garden of the TrimBot2020 project, for applications of gardening robotics. We carried out experiments with a localization method based on fiducial ArUco markers (with and without a Kalman filter) and with ORB-SLAM2, and analyzed the points of strength and weakness of the two approaches. The use of Kalman filter contributes to improvement of the stability of the trajectory detected by ArUco and to partially overcome the marker pose ambiguity problem. Marker- and keypoint-based localization are complementary approaches and their joint employment for localization would improve the problems of map initialization, drift of the estimated position and re-localization.

Acknowledgment

This research has been partially funded by the EU H2020 research and innovation framework (grant no. 688007, TrimBot2020), and by projects TIN2016-75279-P and IFI16/00033 (ISCIII) of Spain Ministry of Economy, Industry and Competitiveness, and FEDER.

References

[1] R. T. Azuma, "A survey of augmented reality," *Presence*, vol. 6, pp. 355–385, 1997.

[2] V. Lepetit and P. Fua, "Monocular model-based 3d tracking of rigid objects: A survey," in *Foundations and Trends in Computer Graphics and Vision*, 2005, pp. 1–89.

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

[4] 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*, pp. 1188–1197, 2009.

[5] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1643–1649.

[6] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," in *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*. Citeseer, 2013, p. 2013.

[7] M. Lopez-Antequera, R. Gomez-Ojeda, N. Petkov, and J. Gonzalez-Jimenez, "Appearance-invariant place recognition by discriminatively training a convolutional neural network," *Pattern Recognition Letters*, vol. 92, pp. 89–95, 2017.

[8] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[9] C. McManus, W. Churchill, W. Maddern, A. D. Stewart, and P. Newman, "Shady dealings: Robust, long-term visual localisation using illumination invariance," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 901–906.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct 2015.

[11] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, Nov 2007, pp. 225–234.

[12] S. Garrido-Jurado, R. Muoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014.

[13] M. Fiala, "Designing highly reliable fiducial markers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1317–1324, 2010.

[14] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*. IEEE Computer Society, 1999, pp. 85–94.

[15] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavári, L. M. Encarnação, M. Gervautz, and W. Purgathofer, "The studierstube augmented reality project," *Presence: Teleoper. Virtual Environ.*, vol. 11, no. 1, pp. 33–54, Feb. 2002.

[16] S. Garrido-Jurado, R. Muoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*, vol. 51, pp. 481 – 491, 2016.

[17] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2938–2946.

[18] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. CVPR*, vol. 3, 2017, p. 8.

[19] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 252–286, 2014.

[20] R. Muoz-Salinas, M. J. Marn-Jimenez, E. Yeguas-Bolivar, and R. Medina-Carnicer, "Mapping and localization from planar markers," *Pattern Recognition*, vol. 73, no. Supplement C, pp. 158 – 171, 2018.

[21] N. Strisciuglio, R. Tylecek, M. Blaich, N. Petkov, P. Bieber, J. Hemming, E. van Henten, T. Sattler, M. Pollefeys, T. Gevers, T. Brox, and R. B. Fisher, "Trimbot2020: an outdoor robot for automatic gardening," in *50th International Symposium on Robotics*, 2018.

[22] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic visual localization," *arXiv preprint arXiv:1712.05773*, 2017.

[23] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32 – 46, 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0734189X85900167

[24] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 2, no. 10, pp. 112 – 122, 1973.

[25] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan 1979.

[26] K. Madsen, H. B. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems (2nd ed.)," Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, p. 60, 2004.

[27] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 573–580.

[28] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.