 DOI: 10.3217/978-3-85125-663-5-03

Using High-Level Features and Neglecting Low-Level Features: Application Self-Localization

Felix König[1] and Markus Bader[2]

Abstract— **Common self-localization algorithms as well as trajectory algorithms for autonomous vehicles rely on low-level features such as laser readings. Identifying higher-level features or objects increases the system quality, but often contradicts the sensor noise model used, especially in the case of dynamic features such as doors, humans or other vehicles. A laser scan of an open door, for example, can look like the end of a corridor, leading to false data association between a map and features. The novelty in this work is that features which belong to the category of dynamic objects are only used as high-level features and are removed from the low-level feature pool. In the work presented here, rgb cameras as well as a laser scanner readings are used to detect doors and to estimate their opening angle. These dynamic features are used as landmarks for self-localization, and the corresponding laser scan readings are ignored by particle weighting. The resulting method is currently a work in progress, and preliminary results are shown. The software developed for this paper is publicly available and integrated into the open-source mobile robot programming toolkit (MRPT).**

I. Introduction

For tackling self-localization within autonomous robotics, a vast set of probabilistic [5], [13] and optimization-based approaches [16] have emerged from current research. Those algorithms rely on external and internal sensors such as cameras, laser scanners and odometry measurement devices for perceiving the environment. During localization, certain assumptions about the world have to be made. The most common assumption is the static world assumption, in which a robot assumes the perceived environment to be static [5], [13], i.e. consisting solely of non-movable objects. In order to be able to deal with dynamic or unexpected objects, the sensor noise model must incorporate unexpected obstacles like P_{unexp} in the Beam-based Proximity Model [13]. The idea behind this work is to remove data corresponding to dynamic objects in the localization algorithm and to alter the noise models accordingly. In order to present a proof of concept, we detect doors with their opening angles. Identifying high-level objects requires a suitable object detection framework, ideally applied during SLAM, in order to construct a map in which the position of all non-movable high-level objects is known. A further benefit to using high-level objects within the localization procedure is better initialization of self-localization. The initial pose of the robot has to be set manually or can be computed automatically by using a large amount of particles, thereby increasing computational complexity [13], [14]. Since this method uses high-level objects, initialization can be improved, as described in Section III.

This paper is organized as follows: Section II discusses related work on the topic of high-level object recognition and integration into self-localization. Section III outlines the approach in depth. Section IV discusses the experiments and a conclusion is presented in section VI.

*The research leading to these results has received funding from the Austrian Research Promotion Agency (FFG) according to grant agreement 855409 (AutonomousFleet) and 854865 (TransportBuddy).

[1]Felix Koenig and Markus Bader are with the Institute of Computer Engineering, Vienna University of Technology, Treitlstr. 1-3/4. Floor/E191-3, 1040 Vienna, Austria [firstname.lastname]@tuwien.ac.at

II. Related Work

Self-localization algorithms can be coarsely classified into laser-scan-based and vision-based works [4], [16]. The latter operate on images, identifying landmarks across images and using bundle adjustment [15] for position estimation. The map-building process is usually performed online (SLAM) and used only to detect loop closures [10]. During pure odometry estimation [6], [8], no map is computed and no loop closures are possible, leading to poorer location accuracy with a detailed evaluation in [16].

Scan-based techniques operate on a two dimensional laser scan, matching the measurements with the known map [13] in order to determine the correct location. As this paper focuses on a more precise treatment of dynamic non-movable objects within a known environment, and laser scans offer limited capability of object detection, it is best suited to improving this class of algorithms, provided the autonomous vehicle is equipped with a camera used for object detection. Vision-based algorithms could, however, also benefit from object detection, especially during the optimization procedure described in [6] and [16], in which a weighted loss function is used to filter outliers after features have been matched. The weighted loss could be replaced by the detection of dynamic movable objects.

Regarding object detection, a vast amount of literature as well as open source software is available. This work focuses on object detection on laser scans as well as within camera images. The current state-of-the-art in image-based object detection uses techniques emerging from deep learning [7], [12]. The authors of [11] present an improvement to [12] that is real time capable, operating at 60-70 frames per second and therefore perfectly suited to a mobile robot scenario. The algorithm identifies a vast set of object classes, including objects in a typical lab environment such as chairs, desks, people and doors, delivering bounding box hierarchies corresponding to detections.

Regarding object detection in the context of mobile robot

self-localization, a significantly smaller set of publications is available. In [1], the authors focus on identifying and modeling doors within a lab environment. The color and shape properties of doors and walls are learned with an expectation maximization (EM) algorithm, which is also used for localization, resulting in a tightly coupled localization and detection. The shape properties of doors are continuously updated during the travel time of the robot. They can subsequently be reused in the same or a different environment.
Current state-of-the-art SLAM algorithms, including object discovery, focus on non-movable objects and introduce them as landmarks, reusing them for loop closure as outlined in [3], [9]. Thrun et al. [17] propose an extension to the Monte Carlo localization framework, tightly integrating movable objects in order to improve localization accuracy. Neither does their approach make use of machine learning for object discovery, nor does it filter out semi-static objects such as doors. Furthermore, the moving object tracking is integrated into the particle filter (PF) used for localization.

III. Approach

The algorithm is presented in two subsections. First, the door detection algorithm is described, and second, the localization algorithm is explained.

A. Door detection and parameter initialization

As mentioned, the object detection of [11] is used in combination with a detector working solely on the laser scan. The idea behind the laser-scan-based segmentation is to identify door hinges and, subsequently, properties of the doors such as their opening angles. The initial procedure, segments the laser scan into contours. The Segmentation is based on the change in depth data between neighboring laser ranges. During experimentation on different datasets with varying thresholds a value of 0.25m has been determined to yield a stable true positive rate across the whole map. Subsequently, a line-fitting algorithm is then performed on each contour segment. If a line segment corresponds to a door as detected by [11], the closest neighboring line segment is determined to be the wall. Finally, the opening angle of the door can be determined as well as the location of the hinge and the length of the door. The landmark-based localization will subsequently be used to correct the parameters.

B. Localization

In this section, the particle filter is outlined based on the Rao-Blackwellized Particle Filter implementation (RBPF) within MRPT [2], [5]. For a more in-depth introduction on PFs, which is out of the scope of this paper, Thrun et al. [13] offer a detailed review of the topic. If k denotes a specific point in time, the robot's states are then written as x_k, with

$$\mathbf{x}_0^k := \{x_0, x_1, \ldots, x_k\}.$$

The states themselves are composed of the position and angular orientation with respect to the world coordinate system.

$$x_k := \begin{pmatrix} x_k \\ y_k \\ \theta_k \end{pmatrix}$$

Additionally, static objects are defined as landmarks l_k, motion sensor measurements as u_k and laser scan measurements as z_k, with the sets defined similarly to in $\mathbf{x}_0^k$.

$$\mathbf{l}_0^k := \{l_0, l_1, \ldots, l_k\}$$
$$\mathbf{u}_0^k := \{u_0, u_1, \ldots, u_k\}$$
$$\mathbf{z}_0^k := \{z_0, z_1, \ldots, z_k\}$$

The PF equation derived models the distribution, capturing the most likely robot states according to their corresponding measurements. It can be decomposed into a *perceptive* and a *predictive* part.

$$p(\mathbf{x}_0^k|\mathbf{z}_0^k, \mathbf{u}_0^k) \propto \overbrace{p(z_k|x_k)}^{\text{perception}} \cdot \underbrace{\int_0^k p(x_k|x_{k-1}, u_k) p(\mathbf{x}_0^{k-1}|\mathbf{z}_0^{k-1}, \mathbf{u}_0^{k-1}) dx_{k-1}}_{\text{prediction}}$$

The predictive element models the likelihood of the actual state with regard to its previous states and measurements, from which the most probable state is recursively inferred.
The perceptive part tries to model the correspondence of the recent state to the current measurement and is referred to as the *beam model* in the literature covering laser scan approaches [13]. This term is of interest within our framework as it also includes the noise in the data, and this is precisely how objects and doors factor into the equation. The beam model is formulated as a mixture of four densities, depicted in Fig. 1, capturing the sensor properties of the laser scanner. All of them incorporate noise in the measurements to a degree. The probability that a laser beam hits known obstacles, for instance walls, is described by p_{hit}. It is defined by a Gaussian distribution corrupted by noise around the true but unknown reading z_k^*. The distributions p_{max} and p_{rand} explain the maximum sensor reading as well as general noise in the data, respectively.

C. Occlusions, Noise and Dynamic Objects

What happens during the algorithm when an unknown obstacle is registered by a beam? In this case, p_{short} aims at capturing this occurrence. The idea is, if multiple dynamic obstacles are present, it is more likely that they will be registered at a short distance from the front of the robot, since they cause occlusions of other obstacles. Therefore, p_{short} is modeled as a Gaussian falloff curve. This term essentially captures the features from dynamic objects as noise in the data. Therefore, the approaches that tightly integrate the dynamic object tracking and estimation into the localization [17], [1], actually treat them at two different levels during localization, which is redundant and erroneous. First, they appear in the sensor noise model, second, despite

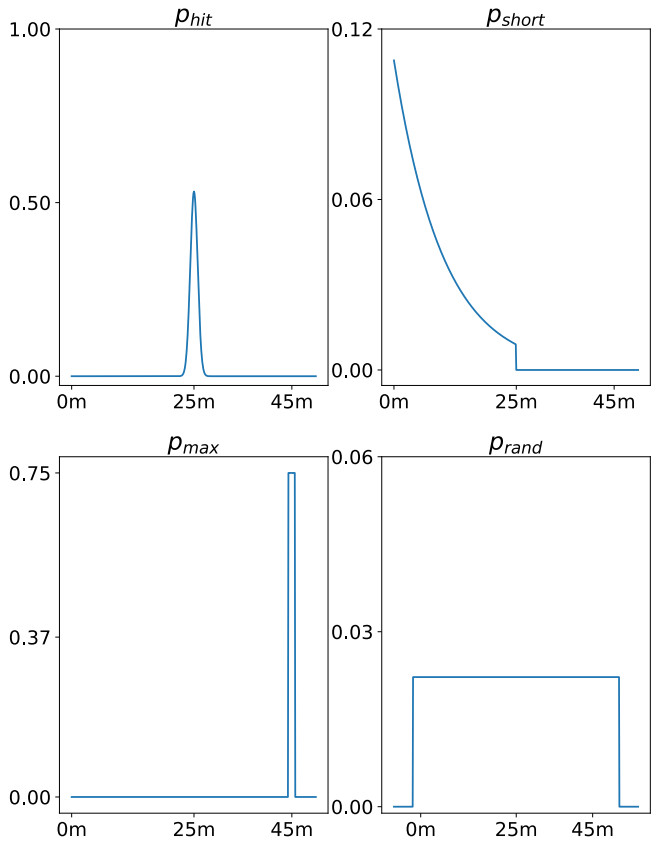

Fig. 1: Sensor model of the laser scan's measurement uncertainty model with $z_k^* = 25$ and $z_{max} = 45$

their parameters are known and estimated, the sensor noise model is not adapted and does not benefit from the gained knowledge. This has not been addressed in the literature, at least to the best of our knowledge. The successive section in this work studies the effects that a complete removal of dynamic objects has on the sensor readings and discusses possible implications.

IV. Experiments

A. Setup

This section describes the preliminary results of the above presented approach. At the moment, the door detector is employed and doors are removed from the scan. The sensor readings are then analyzed and a conclusion about the noise model is made. The following two experiments have been conducted:

1) Driving along an office hallway with opened doors.
2) Standing in front of an open door.

In each experiment a map of the environment is present and used for localization. Doors have either not been modeled within the map or they are modeled as being closed. Fig. 2 depicts a robot travelling in the office environment. The traversable region is black and walls are colored in gray. The map has a resolution of 3737×2338 with each pixel representing a 1cm^2 grid, limiting the accuracy of our evaluation to this range. The evaluation has been performed via ray-tracing. Each laser beam, as shown in red in Fig. 2, is traversed until it hits the first pixel on the wall. The pixel coordinates are subsequently converted from the world into the laser reference frame and its euclidean norm represents the expected distance measurement $z^*_{i,k}$ for the i^{th} beam at time k. The measured laser range value is denoted as $z_{i,k}$. Within the Figure, expected measurements are colored in green, while observed measurements are colored in blue.

In order to transform points into the reference frame of the laser and successively establish a relationship between the world and robot coordinate system the position of the robot is needed. Localization errors happen during abrupt movements of the robot and are not entirely avoidable. In this particular case, they will factor into the sensor noise model together with the beams hitting dynamic obstacles. This can be observed in Fig. 2 where the expected laser ranges do not precisely coincide with the observed ones on the northern wall. However, it is shown in the next section that this does not affect the overall quality of our comparison between the measurements with and without dynamic obstacles. In order to provide an example without localization inaccuracies, a second experiment has been conducted with the robot standing still in front of the door. During the first experiment, a total of 7 doors have been opened in a $45-120°$ angle (being closed is equal to angle 0) along the hallway and the robot is travelling from right to left (see Fig.2). In the second experiment, the robot is standing in front of a door and is not moving.

B. Evaluation

The two experiments have been evaluated as follows. First, the robot is driving along the hallway and records all the laser measurements into a ROS-bag file. During the travel time the localization runs in parallel so that the correct frame transformations can be associated to the measurements. Second, the data is then processed offline both with and without door detection.

The space of all possible laser ranges is represented as a two-dimensional histogram h where each cell has the dimensions of 25×25cm with the entry $h(0,0)$ denoting the robots origin. For each pair of expected measurements $z_{i,k}$ and observations $z^*_{i,k}$, the cell of $h(id(z_{i,k}), id(z^*_{i,k}))$ is incremented by 1, resulting in a correlation matrix. Here, the function $id()$ determines the correct grid cell for each measurement. Finally the values are normalized along each column that represents all observations for a single expectation:

$$\forall l,i : h(l,i) = \frac{h(l,i)}{\sum_j h(j,i)}$$

V. Results

Fig. 3 shows the matrix h for the scenario in experiment 1. The correct observations are present on the diagonal and, for the purpose of visualization, have been clipped at 0.25%. The left image depicts the matrix h_u obtained from equation IV-B using the raw laser scan whereas the right image shows the

Fig. 2: Map of the office environment used for conducting the experiments together with a laser scan at time k.

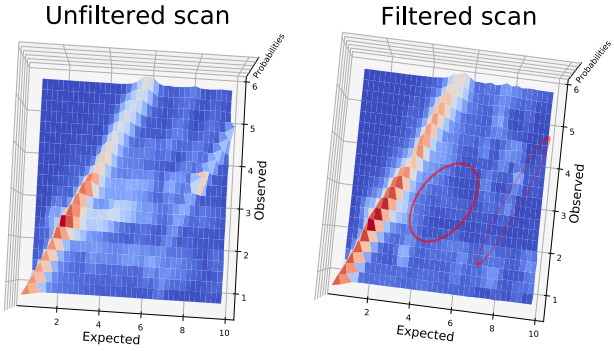

Fig. 3: Plot of h_u and h_f from the first experiment. The red ellipses show the region affected by removing outliers corresponding to doors.

resulting histogram h_f with filtered doors. It is apparent that the line parallel to the diagonal of h_u is not present in h_f and is caused by travelling towards a door opened directly in front of an expected wall. Therefore, the observations are shifted by the distance between the door and the wall. Furthermore, it is also visible that the entries along the diagonal on the right are higher than those on the left, especially along entries where the door has been filtered in $h_f(4\ldots6,4\ldots6)$.
The noise cluster within $h_u(2\ldots4,2\ldots8)$ has been filtered accordingly. Since we drove close to the doors and then around them, they are observable in the vicinity of the robot. Fig. 4 includes the resulting histograms of the second experiment where the robot is standing in front of the door.

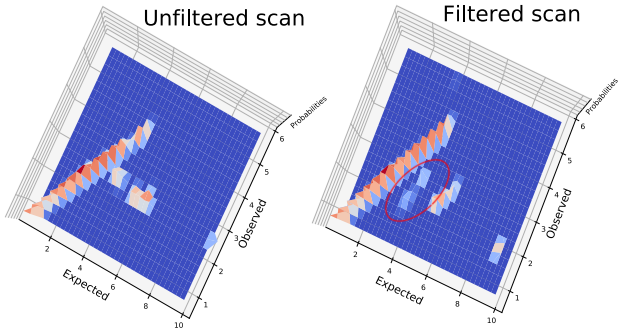

Fig. 4: Plot of h_u and h_f from the second experiment. The red ellipse shows the region affected by removing outliers corresponding to doors.

The filtered result is still noisy primarily due to outliers that do not belong to the door but have not been filtered, leading to the conclusion that the filtering is more beneficial in the moving robot scenario of the first experiment. Both experiments show that sensor measurements observed from dynamic objects manifest themselves in the sensor noise model in the range $[0,z_k^*]$. It is therefore necessary to make different assumptions of the model and potentially remove or redesign the function p_{short}. Future work will aim at addressing this issue.

VI. Conclusion

This work shows the importance of treating object tracking and detection at the level of the sensor noise model. Furthermore, an approach is outlined describing a basic door detector in conjunction with a localization procedure. Precise treatment of obstacles within the sensor noise model and successive integration into the localization procedure remains the topic of future research.

References

[1] D. Anguelov, D. Koller, E. Parker, and S. Thrun, "Detecting and modeling doors with mobile robots," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 4, April 2004, pp. 3777–3784 Vol.4.

[2] J.-L. Blanco-Claraco, "MRPT," https://www.mrpt.org/, 2019, [Online; accessed 14-March-2019].

[3] S. Choudhary, A. J. B. Trevor, H. I. Christensen, and F. Dellaert, "Slam with object discovery, modeling and mapping," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 1018–1025.

[4] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[5] A. Doucet, N. d. Freitas, K. P. Murphy, and S. J. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, ser. UAI '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 176–183. [Online]. Available: http://dl.acm.org/citation.cfm?id=647234.720075

[6] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Mar. 2018.

[7] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, cited By 7777.

[8] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015, cited By 241.

[9] J. McCormac, R. Clark, M. Bloesch, A. J. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," *CoRR*, vol. abs/1808.08378, 2018. [Online]. Available: http://arxiv.org/abs/1808.08378

[10] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018, cited By 11.

[11] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016. [Online]. Available: http://arxiv.org/abs/1612.08242

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

[13] S. Thrun, W. Burgard, D. Fox, and R. Arkin, *Probabilistic Robotics*, ser. Intelligent robotics and autonomous agents. MIT Press, 2005. [Online]. Available: https://books.google.at/books?id=2Zn6AQAAQBAJ

[14] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Monte carlo localization for mobile robots," in *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[15] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1883, pp. 298–372, 2000.

[16] V. Usenko, J. Engel, J. Stueckler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[17] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007.