

Optimal Positioning of Mobile Platforms for Accurate Manipulation Operations

Iker Lluvia¹, Ander Ansuategi¹, Carlos Tubío¹, Loreto Susperregi¹, Iñaki Maurtua¹,
Elena Lazkano²

¹IK4-TEKNIKER, Eibar, Spain

²Faculty of Informatics, University of the Basque Country, Donostia, Spain

Email: ander.ansuategi@tekniker.es, carlos.tubio@tekniker.es, loreto.susperregi@tekniker.es, inaki.maurtua@tekniker.es,
e.lazkano@ehu.eus

How to cite this paper: Lluvia, I., Ansuategi, A., Tubío, C., Susperregi, L., Maurtua, I. and Lazkano, E. (2019) Optimal Positioning of Mobile Platforms for Accurate Manipulation Operations. *Journal of Computer and Communications*, 7, 1-16.

<https://doi.org/10.4236/jcc.2019.75001>

Received: April 2, 2019

Accepted: May 12, 2019

Published: May 15, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Many mobile robotics applications, especially in industrial environments, require the robot to perform safe navigation and then reach the goal with a high precision. In this research work, the objective is to analyze the appropriateness of autonomous natural navigation strategies for mobile manipulation tasks. The system must position itself in a realistic map, follow a path closely and then achieve an accurate positioning in the destination point in order to be able to perform the manipulation, inspection or pick task efficiently. Autonomous navigation is not able to fulfill the accuracy required by some of the jobs so that a second positioning system using vision is proposed in this paper. The experiments show that localization systems have, on average, an error greater than a decimetre and how an additional positioning system can reduce it to a few millimetres.

Keywords

Mobile Robot, Mapping, Localization, Vision, Accurate Positioning

1. Introduction

Natural navigation aims to design robot systems able to navigate without modifying the environment, *i.e.*, without adding extra infrastructure to it. Nowadays, those robot navigation techniques' performance changes considerably depending upon the environment, as their effectiveness directly depends on the features placed for the robot to localize. The sensors readings are matched against the map, and the pose is estimated based on the resemblance between them. As a consequence, the more featured the surroundings, the easier the system localizes

itself, increasing its accuracy, robustness and overall behaviour. Shop floors and industrial environments are full of repetitive areas or long corridors where the system localization critically depends on unreliable odometry data, due to high sensor aliasing. Consequently, the accuracy of localization estimation methods deteriorates. Moreover, those environments suffer frequent modifications that would require a remapping of the environment not always affordable.

Even though many industrial applications do not require the robot to maintain an accurate localization along the whole trajectory, they do need it at some specific points [1]. More specifically, industrial mobile manipulators are more tolerant to deviations during the navigation between destinations, being enough to reach the objective safely. In contrast, they require an excellent positioning in the target pose where they have to perform a task of manipulation, inspection or picking efficiently. The research described in this paper focuses on giving a solution for an efficient and flexible assistive robotic inspection task. The robotic platform needs to navigate through an industrial plant to reach the inspection area where there is an aileron placed in a fixture (see **Figure 1**). Afterwards, the platform must move around stopping at some preset points accurately until the robotic arm mounted upon the platform carries out a fast and robust ultrasonic scanning of the aileron to precisely follow complex-curvatures of the composite parts. As the workspace of the arm is limited, it cannot scan the whole surface at once and thus more nearby points are required.

The goal of the presented work is twofold:

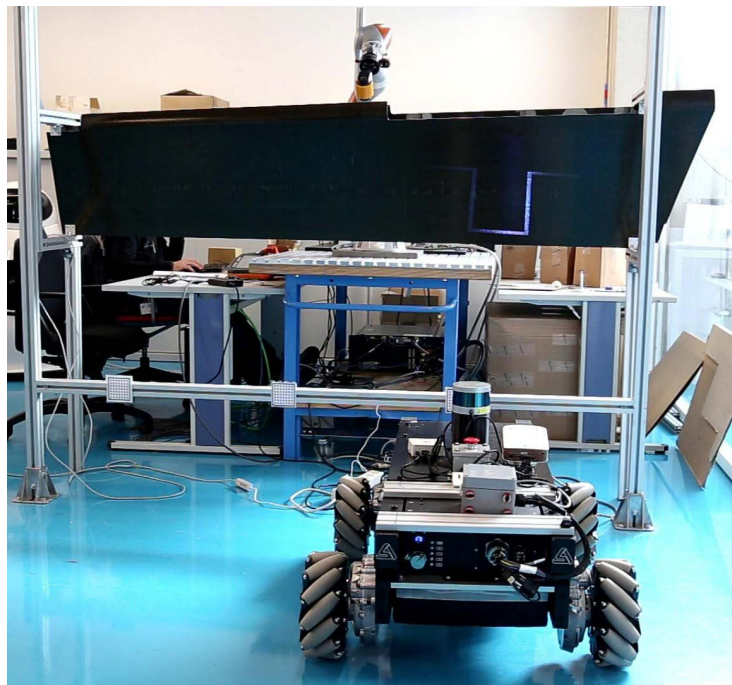


Figure 1. The accurate positioning scenario. In the middle, the aileron to be scanned. In front of it and in the lower right of the image, the mobile platform aiming one of the three vision labels. Behind the aileron there is the robotic arm that will be mounted on top of the platform.

1) Compare the adequateness of different mapping algorithms for the task in hand. The error produced by each approach in the robot localization is measured using a laser tracker device.

2) Measure the final positioning accuracy and correct the error by using vision. Based on the prior expertise in this field and a bibliographical revision [2] [3] [4] [5] [6], an additional positioning system is proposed to deal with this challenge and correct the error with which the robot has arrived at the desired point.

The paper is structured as follows: Section 2 reviews the literature. Next, the taken approach is described in Section 3. Section 4 details the experiments made in order to evaluate the different mapping and positioning systems. Section 5 analyses the obtained results. The paper concludes with Section 6, giving conclusions and pinpointing future work.

2. Related Work

Over the last years, autonomous mobile robot navigation probabilistic techniques have successfully been applied to a wide range of tasks. Those techniques make explicit the uncertainty in environment perception and robot motion not only while mapping the environment, but also while localizing the robot in the map and planning the path to the goal, the three subproblems robot navigation is divided into [7]. The Simultaneous Localization and Mapping (SLAM) problem is tackled by applying the uncertainty in the robot localization during the mapping process [8].

Probabilistic approaches have become popular as they give a robust solution to the localization problem, based on algorithms such as Kalman Filters [9], Histogram and Particle Filters [10] or Monte Carlo Localization [7]. Probabilistic localization techniques are mainly based on two information sources: motion information usually obtained from wheel encoders (odometry) and perception information needed to correct the estimated position by matching each obtained sensor pattern in the map.

Among the different sensors used for correcting the estimated odometry positions, LiDAR sensors are preferred nowadays as they are quite reliable in most scenarios [11] [12] [13] [14]. LiDARs send out laser pulses and measure the returned wavelength to obtain the distance to the first object in its path. They are robust with respect to light effects as shadows or reflections, indeed, they do not require daylight to capture the data and, for many applications, point clouds captured at night are preferred over the ones captured during the day. Nevertheless, digital cameras have recently been proposed in many solutions as they are cheap and light, and coloured representations of the environment are very informative for visual detection [15] [16] [17]. That said, many mobile LiDAR sensors are commercially sold as a unit in conjunction with a camera. Both sensors collect data simultaneously and in a post-processing the colour attribute is added to every point of the point cloud, obtaining a coloured point cloud, which

can better represent semantic information extracted from pictures.

Visual information is also used for localization purposes [18]. Visual landmarks are placed and their positions with respect to the map reference frame are located so that the robot can continuously find them [19]. Although this method improves the quality of the pose estimation in a relevant way, it has some significant disadvantages. On the one hand, the environment must be modified, making its installation expensive, longer and even impossible in some cases. On the other hand, the positioning of the external references should be measured accurately. This task's error will increase the error already produced by odometry. Besides, those measurements will only be valid for a particular map. If the map changes, the reference frame will also vary, and the coordinates of the landmarks would need to be recalculated in order to fit the new map.

Several strategies designed to work outdoors use global positioning systems (GPSs) to solve the localization problem [20] [21]. However, those systems do not succeed in common factories due to the bad quality of the signals produced by [22]: 1) metal structures in thick walls industrial constructions are built of. Those structures prevent signals from being spread properly. Moreover, in some cases completely unreachable areas occur. 2) Factories usually have a considerable amount of machines, each provided with its own communication devices. Those devices may generate interferences or even loss of information which can affect the navigation process itself [23]. These issues can be addressed in different ways. Some approaches propose to use signal repeaters or amplifiers to improve the quality of the signals [24] or replace the global positioning system with a local positioning system [25]. Both alternatives are based on the same idea of receiving the localization information from an external source.

As mentioned in the introduction, the goal of the work presented here is to develop a natural robot navigation system for industrial manipulation purposes and thus, no external environmental requirement should be added to successfully perform the goal.

However, it must be taken into account that a proper navigation does not necessarily lead to an accurate positioning. As it has been mentioned before, in the specific robotic task in hands the overall localization has a higher margin of error than the positioning at the destination point. There are some systems that do really get accurate positioning results. For instance, a three-dimensional indoor positioning system using LED lights and detecting the time difference of arrival proposed in [26] achieves a mean error as low as 1 mm but in simulation and in a relatively small room. Some RGB-D cameras based techniques state an accuracy of few centimetres with synthetic datasets [15] [27].

Nonetheless, visual odometry systems on their own do not need a map whereby navigate, *i.e.*, they just need a representation of the environment for localization. Most navigation systems divide the ground of the map in cells getting the corresponding grid-map, which allows a more efficient path planning and obstacle avoidance. Accordingly, they operate in the discrete space with the consequent loss of information whilst visual odometry and scan-matching algorithms

remain in the continuous. In that vein, Röwekämper *et al.* [2] report a maximum error of 17 mm and 0.53 deg. Other attempts that use scan-matching techniques get position errors of decimetres [3] [4]. This work tries to be ambitious in an attempt to achieve a better accuracy in the final positioning making use of a camera. In addition, Tang *et al.* [28] have probed that some IMUs can add noise in static positioning or when the robot operate in minimal velocities worsening the odometry data instead of improving it.

In the approach proposed here the robot will navigate making use of a LiDAR scan together with the wheel encoders odometry rectified with the IMU, and only the last position achieved prior to the manipulation task will be corrected using vision. So it does not focus just in navigation or in the accurate positioning, as many other works do. It goes a step further and combines both processes in a unique solution, minimising the error in each of them, and especially in the final positioning.

3. Optimal Positioning Measurement Approach

As mentioned before, we aim to find the best mapping approach on the one hand, and to correct the localization error at the goal position on the other hand. Thus, the approach is divided in two main sections, focusing the first one in autonomous navigation techniques, and in accurate positioning at the destination point in the second one.

3.1. Robotic Platform

The robotic platform is a Segway RMP 440 Flex Omni with four mecanum wheels, capable of moving omnidirectionally at 2.2 m/s (5 mph) with a maximum payload of 450 kg (1000 lbs). Although its velocities are limited to 0.5 m/s (1.1 mph) and 1 rad/s during the experiments for control and safety.

It includes the following main elements (see **Figure 2**):



Figure 2. The Segway RMP 440 Flex Omni navigating through the east half of the workshop.

- **PC with Intel i7 Processor:** Core i7-3517UE (1.7 - 2.8 GHz + HD 4000), 16 GB (2 × 8 GB) DDR3-1333/1600 SDRAM, 128GB high performance SSD.
- **Velodyne VLP-16 LiDAR PUCK:** Real-time, 360°, 3D distance and calibrated reflectivity measurements with a 150 m range, 360° horizontal field of view and a 30° vertical field of view.
- **Machine vision camera:** Optical format of 1/1.8" and a focal length of 12 mm.
- **CH-Robotics UM7 IMU:** Attitude and Heading Reference System (AHRS) for precise pose estimation.

3.2. Mapping and Localization

The algorithm used for building the map may make a big difference so it is important to choose the appropriate one. Regardless of the map building method used, the procedure to be followed is nearly the same. The robot should move around the environment allowing the sensors to gather information of the area. The quality of the map will depend directly on the softness of the platform motion and thus, movements should be as smooth as possible. Hence, it is highly recommended to limit velocities specially rotational ones. Abrupt turns can result in a much harder scan matching problem. Besides, it is also a good practice to repeat the first meters of the trajectory at the end of the process because many methods have to perform a loop closure at the end of the mapping. Loop closing consists of asserting that the vehicle has returned to a previously visited location and correcting the correspondence of the scan matches in the map being built in order to overlap the initial and final positions.

Obviously, the mapping process requires a post-modelling process in which undesirable elements must be removed (for instance, dynamic obstacles detected during the map construction) and where loop closure is corrected. What obstacles should be mapped remains an open discussion. Theoretically, only static and immobile volumes should be part of it but in most cases that is unfeasible because, strictly speaking, just walls and pillars fulfil that condition. However, much relevant information would be lost and the map would have few similarities with the information the robot collects from the environment while navigating.

Here, three map building strategies are studied: GMapping, Hector Mapping and Cartographer. All of them are available as open source implementations in ROS¹. GMapping [29] is a Rao-Blackwellized particle filter to learn grid maps from laser range data. This approach uses a particle filter in which each particle carries an individual map of the environment. It computes an accurate proposal distribution taking into account not only the movement of the robot but also the most recent observation. On the contrary, the Hector Mapping approach [30] is based on an optimization of the alignment of beam endpoints with the map learnt so far. The basic idea of using a Gauss-Newton approach is inspired by the

¹<http://www.ros.org/>.

field of computer vision and it does not perform loop closure. Finally, Cartographer [31] does not employ a particle filter neither. To cope with the accumulative error, it regularly runs a pose optimisation, creating constant submaps that take part in scan matching for loop closure. Therefore loops are closed immediately using branch-and-bound when a location is revisited. As neither GMapping nor Hector Mapping have a localization-only mode, the probabilistic Adaptive Monte Carlo Localization (AMCL) algorithm has been used for that purpose driving the localization tests.

As mentioned before, industrial environments are full of symmetries and the shape of the laser scan is, to a large extent, equal in spite of height variations while keeping the same latitude and longitude. For this reason during the mapping process (SLAM) only 2D LiDAR information is used instead of the 3D pointcloud. Although the three-dimensional data is flattened in one point the conversion could also have been done by extracting the data in a range of heights and discarding the rest. This downsizing is done in order to be able to extrapolate the method to more robots because not many of them can incorporate a 3D LiDAR. Moreover, handling a complete point cloud involves a greater degree of computational complexity and the extra information does not increase accuracy proportionally.

Once a realistic map is created, the location of the origin is saved with an external measurement unit, which serves as ground truth when comparing the localization precision using that map. In this case, a laser tracker is used for that purpose (Figure 3), a device that measures distances with a micrometric accuracy and that serves as ground truth. The robot travels around the environment



Figure 3. The laser tracker measures distances with a micrometric accuracy. It serves as as ground truth to measure localization errors.

making runs with different styles of navigation, varying the lineal and rotational speeds and accelerations, the sequence and the visited areas of the map, the amount of turns done or the length and duration of the runs. The more realistic the simulated scenarios the better the guess of the behaviour of the algorithm, and the more meaningful the experiment will be. The pose estimations of the algorithms are tracked while the real position of the robot is saved with the laser tracker and afterwards both data are associated based on their timestamp. Statistical values are obtained from that analysis which allows to establish a quantitative performance of each algorithm.

3.3. Positioning

The error calculated from the comparison between the estimations of the algorithms and the measurements of the laser tracker is the accuracy with which the robot arrives to the last phase. The procedure defined to correct the deviation of the navigation and achieve an accurate positioning must be capable of absorbing that value. In this experiment, that task is done with vision and it takes into account the error of the navigation at the time of choosing the lens, the label and the spatial distances between them. The relation from the label to the target pose is also calibrated with the laser tracker, every measurement or calculus must be done as precise as possible. There are many links in the transformations chain and assuming minor errors in any step can lead to a significant error in the final result. Finally, when the robot reaches the destination it corrects the pose based on where the robot identifies the label, trying to get as close as possible to the ideal pose.

The vision system requires the spatial relationship between the target and the label to be known and measured accurately, as the label is not placed with respect to a known global frame. Consequently, another requisite is that the label must be located within the field of view of the sensor that will afterwards be used for correcting the final positioning error. Ergo there is a maximum error tolerance in natural navigation determined by the features of the camera and lens chosen. Aspects like the focus, the depth of field or the angle of view should be calculated beforehand as those features mark the boundaries of the region inside which the vision system must fall after the navigation. And then the configuration of the navigation system is adjusted based on those margins. Summarising, a balance between both systems must be maintained.

3.4. Ground Truth Generation

The proposed vision system is validated using the TRITOP photogrammetric unit. TRITOP is an optical, mobile, measurement system, which accurately defines the 3D coordinates of object points at quasi-static conditions. Based on this information TRITOP is capable of calculating 3D displacements and deformations of objects and components. It consists of one digital single lens camera (DSLR) camera, contrast coded and uncoded bars, scale bars and the correspon-

dent software (in this case TRITOP v6.2 has been used). As in [32], it is used for quality control and validation of the vision system incorporated in the robotic platform. This measuring procedure achieves an accuracy of 0.015 mm in an area of $1 \times 0.5 \times 0.5 \text{ m}^3$, which fits in the settings defined for the validation. The process is as follows. Firstly, the coded plate used in the robot vision system is fixed simulating the same conditions there will be in the real use. Afterwards, several foils with coded points of the TRITOP system are placed around it, so as everything can be seen together. Then, some images are taken with the camera, varying the point of view but trying to catch as much markers as possible. The relation between the plate and the coded foils must be the same in all the images so they must be placed at unmovable locations. In this experiment 14 images have been taken. Then, both the TRITOP system and the robot vision system find the labels in the images and calculate the pose of the camera with respect to a common reference frame. Besides, the TRITOP system computes the intrinsic parameters of the camera too. The results estimate an average image point deviation of 0.02 pixels and an average object point deviation of 0.01 mm. Accordingly, the vision positioning system assumes this error at all times, which is negligible.

4. Experimental Setup

The experiment is proposed to be performed in an industrial rectangular workshop of 90×30 metres, with a wall in the longitudinal axis that divides it in two halves that are communicated by two big sliding doors (they were completely opened during the experiment). This structure forms a sort of circuit with a width of 4 - 8 metres as there are machines, benches and industrial robots next to the walls alongside the route (**Figure 4**).

4.1. Tests

The experimentation is divided in three main sections: mapping, localization



Figure 4. West half of the workshop. At the bottom, the mobile platform navigating.

and positioning. They are performed in that order as each of them has direct impact on the next one. On the one hand, localization estimations may vary depending on the map being used, as it represents how the robot understands the environment. On the other hand, how much the final positioning needs to be corrected is determined by the error on the localization process at the target pose. In these terms, if the map built is mediocre, the whole system will perform poorly. Thus, in order to avoid that strict dependency and to enhance the robustness of the system, the correction of the final pose should not make use of the map.

4.2. Mapping

The robot needs a map for its navigation and it must be as realistic as possible from the point of view of the robot. This means that the map must represent the environment according to how the robot sensors capture it, which does not necessarily correspond with a human concept of realistic. During this experimentation, firstly, a location is set to be the starting point of the mapping process and the origin of the map. With the aid of a joystick, the robot is driven around the environment smoothly showing it every area of the factory floor. The three maps are built simultaneously, during the same teleoperated trajectory.

4.3. Localization

In order to measure the localization error magnitude that the use of the different maps produces, teleoperated trajectories are recorded and afterwards data is reproduced in each map using AMCL offline for localization purposes. First, we set a point with the laser tracker from where all the paths start. This is the origin of the map as well to ensure that the poses given by the three estimations and the laser tracker are referenced with respect to the same coordinate frame.

4.4. Positioning

The last step consists of measuring and correcting the final positioning error during navigation. In this case, the robot uses AMCL for localization and A* for trajectory planning. The robot starts in a in a big work-floor and navigates to a laboratory next to it, passing through a narrow entrance and travelling a distance of approximately 20 m to its final destination. To proceed with the correction of the positioning using the camera, the robot must fall in an area where it is capable of detecting the label. If there is not an error in the orientation, the variability is in area of 20 cm in radius. In contrast, if there is no translational deviation, a maximum error of 16° (0.28 rad) is allowed. The error margin can increase or decrease if both translation and orientation are not perfect in the last pose of the navigation, which is what usually happens. Then, the label is detected and its pose estimation is reliable enough to get an accurate positioning as the method is validated by the TRITOP system.

5. Results

5.1. Maps

It is impossible to determine at a glance which of the maps obtained with the three algorithms is better without having a real map that serves as ground truth. In a sense, this is quantified in the next section but a first comparison is done overlapping all the maps in the same image (**Figure 5**). Apparently, *hector_slam* has a significant deviation in the orientation with respect to *gmapping* and *cartographer* in the right (north) part of the map of **Figure 5**, which is the side that corresponds to the last phase of the trajectory of the mapping process. It may be because *hector_slam* does not have loop closing ability and therefore it does not apply the corresponding transformation in the data to rectify the graph. Besides, *gmapping* and *cartographer* seem to cover a wider range, *i.e.*, there are elements that they add to the map while *hector_slam* do not, as can be clearly seen in the bottom right of **Figure 5**.

5.2. Localization

The amount of tests done is 12, in which 35 different poses are recorded (see **Figure 6**). Among these, 7 are long routes executing each of them a complete loop of the workshop and the other 5 are shorter trajectories that do not visit the whole environment, but all of them finish in the origin. Besides, in 6 out of the 12 navigations the driving is smooth, whereas in other 6 the robot rotates 360?? every 10 m. These sudden rotations are done with the aim of adding uncertainty to the system and see how each configuration responds to them. As expected, jerky navigation does add uncertainty in localization as the errors in translations are bigger in those cases (see **Table 1** and **Table 2**). Translational mean errors in smooth navigation are 0.19, 0.34 and 0.17 meters for each system, smaller compared with the 0.24, 0.50 and 0.25 meters of the abrupt navigation. Nevertheless, although it happens the same with *hector_slam* (1.61 against 2.20 degrees), the opposite occurs in rotation with the two other systems (see **Table 3** and **Table 4**). In the smooth navigation the rotational mean errors got with *gmapping* and *cartographer* are, respectively, 1.08 and 0.94 degrees whereas values of 0.24 and 0.80 degrees are obtained navigating with sudden rotations.

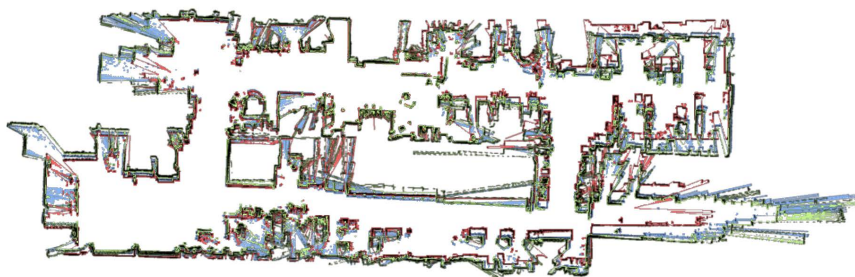


Figure 5. Overlapping of the maps obtained with each algorithm: *gmapping*, blue; *hector_slam*, red; *cartographer*, green. Although in this first comparison it cannot be determined which one is better, *hector_slam* demonstrates a poorer performance in the subsequent tests.

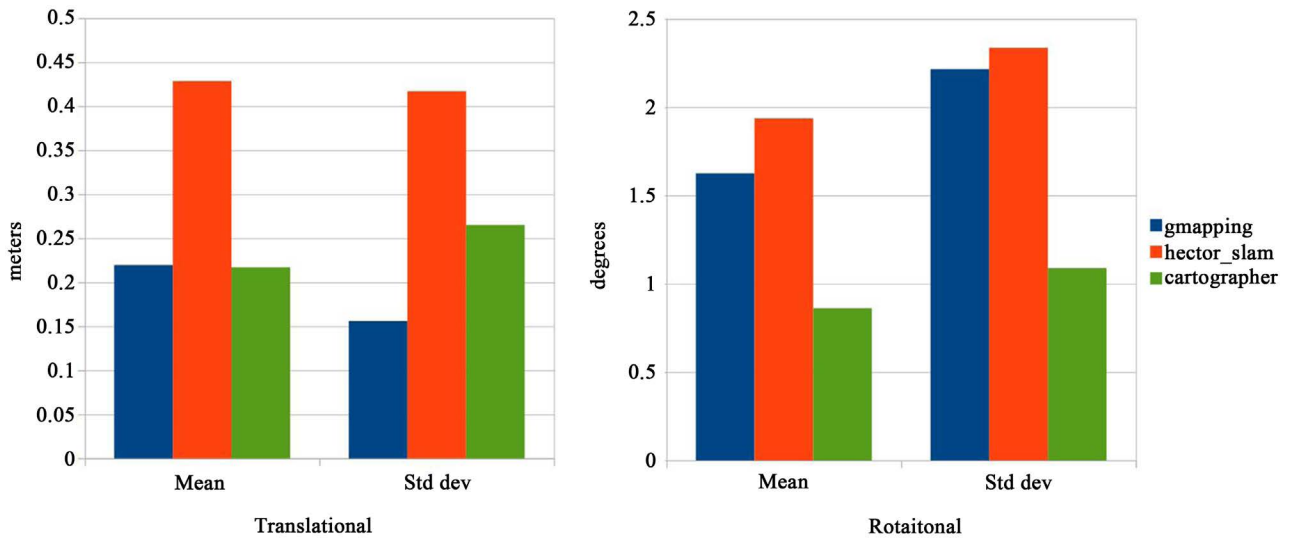


Figure 6. Overall translational and rotational error.

Table 1. Smooth navigation: Translational error in metres.

	gmappig	hector_slam	cartographer
Mean	0.191	0.340	0.170
Min	0.063	0.041	0.001
Max	0.506	1.099	0.543
Std dev	0.130	0.317	0.185

Table 2. Abrupt navigation: Translational error in metres.

	gmappig	hector_slam	cartographer
Mean	0.242	0.498	0.254
Min	0.019	0.019	0.005
Max	0.544	1.509	0.997
Std dev	0.176	0.482	0.317

Table 3. Smooth navigation: Rotational error in degrees.

	gmappig	hector_slam	cartographer
Mean	1.088	1.606	0.939
Min	0.074	0.122	0.037
Max	3.305	5.467	2.884
Std dev	1.077	1.847	0.963

Table 4. Abrupt navigation: Rotational error in degrees.

	gmappig	hector_slam	cartographer
Mean	2.048	2.196	0.800
Min	0.021	0.024	0.028
Max	10.185	10.423	4.393
Std dev	2.782	2.700	1.210

5.3. Positioning

In 25 experiments the robot navigates autonomously to the same destination point but from different starting points. There, the final pose the navigation system leads to is calculated with the vision label, as the location of this marker is known with respect to the map. The purpose of this test is to measure the correction the vision process adds to the system. A total of 25 poses have been recorded, in which an error of 459 mm is corrected in the worst case (see **Figure 7**). $P(0,0)$ is the target location of the navigation where the robot performs the task of inspection. Thanks to the vision system the robot can reach the goal accurately, with a tolerance of 20 mm in x , 3 mm in y and 0.28° (0.002 rad) in rotation (set by software due to the limitations of the control of the platform itself). In all tests, a mean error of 217 mm is amended with a standard deviation of 104 mm and a minimum improvement of 68 mm.

6. Conclusions and Future Work

The main conclusion this experiment leads to is that relative small differences in the generated maps are directly related with the localization procedure accuracy in each one. Therefore, using an appropriate environment representation for the task in hand is of high importance to obtain good results during pose estimation. To determine if a map is good enough, it is mandatory to experimentally test it through real situations. Based on the navigation done with the three maps, *gmapping* and *cartographer* achieve best results. But, in this specific use case, the system using *gmapping* is the preferred one because of its robustness and stability.

As expected, rough movements increase the error in localization, so the robot should have both angular and linear velocities and accelerations limited during the general usage. Even though navigating smoothly, the accuracy of the positioning in the destination point is not enough for some operations and a vision

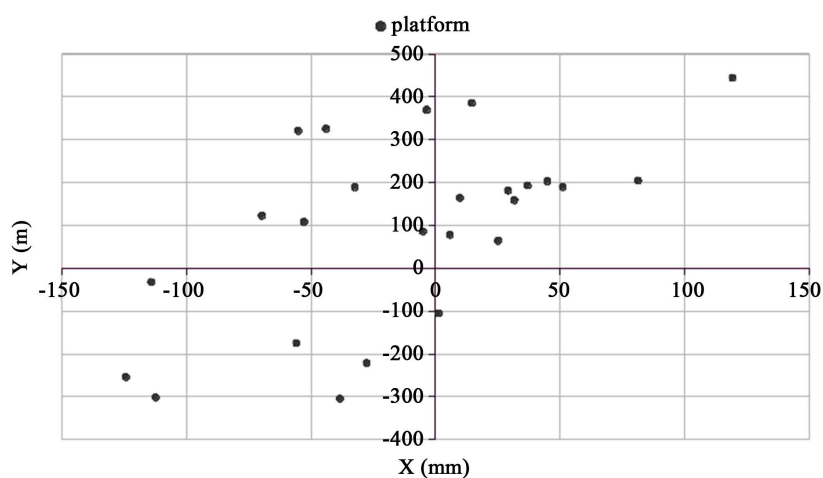


Figure 7. Position of the robot after the navigation and before the visual correction. The $(0,0)$ represents the destination point, and also where the robot is placed after the accurate positioning process.

system could solve this problem, as presented in this work. With the proper vision system, the localization of the mobile platform can be measured and corrected with millimetric accuracy.

The next step is to test the system with a robotic arm on the platform and to analyze how it affects its movements. The arm can generate inertia forces which can increase uncertainty in the platform localization. These should be identified and taken into account in order for the arm to be able to perform the task effectively. Moreover, the capacity of communicating with a robotic arm mounted on the platform opens a wide range of new possibilities for localization, navigation and task performance.

Acknowledgements

This project has received funding from the Clean Sky 2 Joint Undertaking under the European Unions Horizon 2020 research and innovation programme under grant agreement No. 716935.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Martínez-Barberá, H. and Herrero-Pérez, D. (2010) Autonomous Navigation of an Automated Guided Vehicle in Industrial Environments. *Robotics and Computer-Integrated Manufacturing*, **26**, 296-311. <https://doi.org/10.1016/j.rcim.2009.10.003>
- [2] Röwekämper, J., Sprunk, C., Tipaldi, G.D., Stachniss, C., Pfaff, P. and Burgard, W., (2012) On the Position Accuracy of Mobile Robot Localization Based on Particle Filters Combined with Scan Matching. 2012 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, 7-12 October 2012, 3158-3164. <https://doi.org/10.1109/IROS.2012.6385988>
- [3] Tang, J., Chen, Y., A. Jaakkola, J., Liu, J., Hyyppä, J. and Hyyppä, H. (2014) NAVIS-An UGV Indoor Positioning System Using Laser Scan Matching for Large-Area Real-Time Applications. *Sensors*, **14**, 11805-11824. <https://doi.org/10.3390/s140711805>
- [4] Gao, Y., Liu, S., Atia, M.M. and Noureldin, A. (2015) INS/GPS/LiDAR Integrated Navigation System for Urban and Indoor Environments Using Hybrid Scan Matching Algorithm. *Sensors*, **15**, 23286-23302. <https://doi.org/10.3390/s150923286>
- [5] Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D. (2012) A Benchmark for the Evaluation of RGB-D Slam Systems. 2012 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, 7-12 October 2012, 573-580. <https://doi.org/10.1109/IROS.2012.6385773>
- [6] Winterhalter, W., Fleckenstein, F., Steder, B., Spinello, L. and Burgard, W. (2015) Accurate Indoor Localization for RGB-D Smartphones and Tablets Given 2D Floor Plans. 2015 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, 28 September-2 October 2015, 3138-3143. <https://doi.org/10.1109/IROS.2015.7353811>

- [7] Thrun, S., Burgard, W. and Fox, D. (2005) Probabilistic Robotics. The MIT Press, Cambridge, MA.
- [8] Durrant-Whyte, H. and Bailey, T. (2006) Simultaneous Localization and Mapping: Part I. *IEEE Robotics & Automation Magazine*, **13**, 99-110. <https://doi.org/10.1109/MRA.2006.1638022>
- [9] Bailey, T., Nieto, J., Guivant, J., Stevens, M. and Nebot, E. (2006) Consistency of the EKF-Slam Algorithm. 2006 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, 9-15 October 2006, 3562-3568. <https://doi.org/10.1109/IROS.2006.281644>
- [10] Montemerlo, M., Thrun, S., Koller, D. and Wegbreit, B. (2003) FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. *Proceedings of International Joint Conference on Artificial Intelligence*, Acapulco, 9-15 August 2003, 1151-1156.
- [11] Rekleitis, I., Bedwani, J.-L., Dupuis, E., Lamarche, T. and Allard, P. (2013) Autonomous Over-the-Horizon Navigation Using LIDAR Data. *Autonomous Robots*, **34**, 1-18. <https://doi.org/10.1007/s10514-012-9309-9>
- [12] Weiss, U. and Biber, P. (2011) Plant Detection and Mapping for Agricultural Robots Using a 3D LIDAR Sensor. *Robotics and Autonomous Systems*, **59**, 265-273. <https://doi.org/10.1016/j.robot.2011.02.011>
- [13] Raibert, M., Blankespoor, K., Nelson, G. and Playter, R. (2008) BigDog, the Rough-Terrain Quadruped Robot. *IFAC Proceedings Volumes*, **41**, 10822-10825. <https://doi.org/10.3182/20080706-5-KR-1001.01833>
- [14] Levinson, J., Montemerlo, M. and Thrun, S. (2007) Map-Based Precision Vehicle Localization in Urban Environments. *Robotics Science and Systems*, Atlanta, GA, 27-30 June, 2007, 4, 1. <https://doi.org/10.15607/RSS.2007.III.016>
- [15] Mur-Artal, R., Montiel, J.M.M. and Tardos, J.D. (2015) ORB-SLAM: A Versatile and Accurate Monocular Slam System. *IEEE Transactions on Robotics*, **31**, 1147-1163. <https://doi.org/10.1109/TRO.2015.2463671>
- [16] Engel, J., Schöps, T. and Cremers, D. (2014) LSD-SLAM: Large-Scale Direct Monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B. and Tuytelaars, T., Eds., *Computer Vision-ECCV 2014. ECCV 2014. Lecture Notes in Computer Science*, Vol. 8690. Springer, Cham. https://doi.org/10.1007/978-3-319-10605-2_54
- [17] Naseer, T., Burgard, W. and Stachniss, C. (2018) Robust Visual Localization across Seasons. *IEEE Transactions on Robotics*, **34**, 289-302. <https://doi.org/10.1109/TRO.2017.2788045>
- [18] Taketomi, T., Uchiyama, H. and Ikeda, S. (2017) Visual SLAM Algorithms: A Survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, **9**, 16. <https://doi.org/10.1186/s41074-017-0027-2>
- [19] Endo, Y., Balloch, J.C., Grushin, A., Lee, M.W. and Handelman, D. (2016) Landmark-Based Robust Navigation for Tactical UGV Control in GPS-Denied Communication-Degraded Environments. *Unmanned Systems Technology XVIII*, **9837**, 98370F. <https://doi.org/10.1117/12.2224231>
- [20] Agrawal, M. and Konolige, K. (2006) Real-Time Localization in Outdoor Environments Using Stereo Vision and Inexpensive GPS. *18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, 20-24 August 2006, 1063-1068. <https://doi.org/10.1109/ICPR.2006.962>
- [21] Velázquez, R., Pissaloux, E., Rodrigo, P., Carrasco, M., Giannoccaro, N.I. and Lay-Ekuakille, A. (2018) An Outdoor Navigation System for Blind Pedestrians Using GPS and Tactile-Foot Feedback. *Applied Sciences*, **8**, 578.

- <https://doi.org/10.3390/app8040578>
- [22] Scaramuzza, D., Achtelik, M.C., Doitsidis, L., Friedrich, F., Kosmatopoulos, E., Martinelli, A., Achtelik, M.W., Chli, M., Chatzichristofis, S., Kneip, L., et al. (2014) Vision-Controlled Micro Ying Robots: From System Design to Autonomous Navigation and Mapping in GPS-Denied Environments. *IEEE Robotics & Automation Magazine*, **21**, 26-40. <https://doi.org/10.1109/MRA.2014.2322295>
- [23] Lucas-Esteban, M., Maestre, J., Coll-Perales, B., Gozalvez, J. and Lluvia, I. (2018) An Experimental Evaluation of Redundancy in Industrial Wireless Communications. 2018 *IEEE 23rd International Conference on Emerging Technologies and Factory Automation*, Turin, 4-7 September 2018, 1075-1078. <https://doi.org/10.1109/ETFA.2018.8502497>
- [24] Xu, R., Chen, W., Xu, Y. and Ji, S. (2015) A New Indoor Positioning System Architecture Using GPS Signals. *Sensors*, **15**, 10074-10087. <https://doi.org/10.3390/s150510074>
- [25] Skomra, S.A. and Durand, J. (2016) Method and Apparatus for a Local Positioning System. US Patent No. 9258797.
- [26] Nadeem, U., Hassan, N., Pasha, M. and Yuen, C. (2014) Highly Accurate 3D Wireless indoor Positioning System Using White Led Lights. *Electronics Letters*, **50**, 828-830. <https://doi.org/10.1049/el.2014.0353>
- [27] Handa, A., Whelan, T., McDonald, J. and Davison, A.J. (2014) A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and Slam. 2014 *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 31 May-7 June 2014, 1524-1531. <https://doi.org/10.1109/ICRA.2014.6907054>
- [28] Tang, J., Chen, Y., Niu, X., Wang, L., Chen, L., Liu, J., Shi, C. and Hyyppä, J. (2015) Lidar Scan Matching Aided Inertial Navigation System in GNSS-Denied Environments. *Sensors*, **15**, 16710-16728. <https://doi.org/10.3390/s150716710>
- [29] Grisetti, G., Stachniss, C. and Burgard, W. (2007) Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, **23**, 34-46. <https://doi.org/10.1109/TRO.2006.889486>
- [30] Kohlbrecher, S., Meyer, J., von Stryk, O. and Klingauf, U. (2011) A Flexible and Scalable Slam System with Full 3D Motion Estimation. 2011 *IEEE International Symposium on Safety, Security and Rescue Robotics*, Kyoto, 1-5 November 2011, 155-160. <https://doi.org/10.1109/SSRR.2011.6106777>
- [31] Hess, W., Kohler, D., Rapp, H. and Andor, D. (2016) Real-Time Loop Closure in 2D Lidar Slam. 2016 *IEEE International Conference on Robotics and Automation*, Stockholm, 16-21 May 2016, 1271-1278. <https://doi.org/10.1109/ICRA.2016.7487258>
- [32] Koutecký, T., Paloušek, D. and Brandejs, J. (2013) Method of Photogrammetric Measurement Automation Using Tritop System and Industrial Robot. *Optik—International Journal for Light and Electron Optics*, **124**, 3705-3709. <https://doi.org/10.1016/j.ijleo.2012.11.024>