

# Navigation System for Mobile Robots Using PCA-based Localization from Ceiling Depth Images: Experimental Validation

Fernando Carreira and João Calado  
IDMEC/LAETA - IST, Universidade de Lisboa  
and ADEM-ISEL, Instituto Politécnico de Lisboa  
1959-007 Lisboa, Portugal  
Email: {fcarreira,jcalado}@dem.isel.pt

Carlos Carreira and Paulo Oliveira  
IDMEC/LAETA  
IST, Universidade de Lisboa  
1049-001 Lisboa, Portugal  
Email: {carreira.p.oliveira}@dem.ist.utl.pt

**Abstract**—This paper aims the experimental validation of a mobile robot navigation system, using self-localization based on principal component analysis (PCA) of ceiling depth images. In this approach, a roadmap based on generalized Voronoi diagram (GVD) is built from an occupancy grid, that is defined in the ceiling mapping to the PCA database. The system resorts to the Dijkstra algorithm to planning paths, using the GVD-based roadmap, from which a set of waypoints are extracted. During the mission, the robot is commanded by a controller based on dipolar navigation functions (DNF) along the waypoints, being self-located using only the information provided from ceiling depth images and other on-board sensors. The navigation system ensures that the robot reaches its destination, travelling along safety trajectories, while computing its pose with global stable estimates, from Kalman filters (KF). The navigation is achieved without the need to structure the environment, searching by specific features, and to linearize the model. The results are experimentally validated in an indoor environment, using a differential-drive mobile robot.

**Index Terms**—Indoor navigation, Principal component analysis, Kalman filters, Generalized Voronoi diagram, Navigation functions

## I. INTRODUCTION

Navigation is a crucial and an inherent task to mobile robots, so they can perform missions. When a map-based navigation is considered, robots must have self-locating, planning and motion control skills. Usually robots achieve their location extracting specific features from the environment or detecting artificial landmarks. Although the most common approach is to extract features looking around [1], some robots use a camera looking upward [2]–[4]. The use of ceiling vision has the advantage that images can be considered without scaling; are less affected by occlusions due to obstacles moving around; and are usually static, increasing the mapping database reliability. Due to the non-linear kinematics, usually, such features are merged with the robot odometry by an extended Kalman filter (EKF) [2], [5]. However, even though it can give a reasonable performance, the EKF may diverge in consequence of wrong linearisation or sensor noise.

In contrast with feature based techniques that are computationally heavy, the principal component analysis (PCA) method allows simpler implementation approaches due to the high compression ratio of the environment data. The PCA algorithm was proposed by [6] as the positioning system in a terrain reference navigation of underwater vehicles. [7] and [8] implemented a PCA-based self-localization system for

mobile robots, using ceiling images, that provides globally stable estimates, from Kalman filters (KF), but the proposed approach was not integrated into a closed loop control system to perform navigation tasks. The use of depth images in [8] improves the system robustness to cope with varying illumination conditions.

The path planning is often performed by roadmaps, being the generalized Voronoi diagram (GVD) a classical approach to create routes into the free space [9], [10]. When the workspace is configured using an occupancy grid, it is common to build a GVD based on a grid [9], [11], [12]. This approach reduces the planning complexity to a few possible routes and, simultaneously, allows the definition of safety paths, ensuring that the path to the destination is found. In addition to planning, the robot motion is commanded by pose controllers that drives it along such path, until the goal is reached. Navigation functions (NF) are a type of artificial potential fields (APF) that has the advantage to plan and control the mobile robot motion, ensuring an unique minimum located in the goal [13]–[15].

This paper proposes and experimentally validates a navigation system architecture for indoor differential-drive robots moving under ceilings with rich information. The system aims to drive a robot toward a destination, along a safety path, while is self-located only with the information obtained from the PCA of ceiling depth images and other on-board sensors, merged through KF. With this system, no specific features extraction and model linearizations are required. To achieve this, the presented work combines the PCA-based self-localization system from ceiling depth images [8] with a motion controller, based on the dipolar navigation function (DNF) [13], [15], to drive the robot along a set of waypoints extracted from a GVD. Such roadmap is built according to the free space environment, using the ceiling mapping information and implementing the approach proposed by [11].

This paper is organized as follows: Section II presents the proposed navigation system architecture. Section III address the GVD building and the path planning to select the set of waypoints toward a goal. Section IV presents the DNF-based controller to drive the robot toward each waypoint. Section V presents the system navigation experimental validation in an indoor environment. Section VI presents some conclusions.

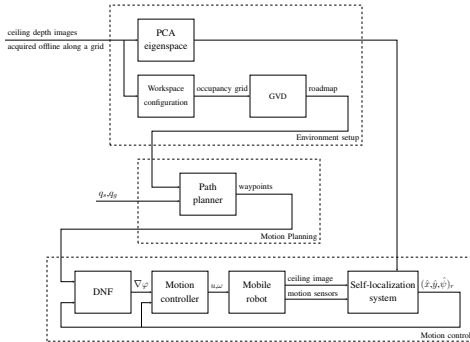


Fig. 1. Navigation system architecture.

## II. NAVIGATION SYSTEM ARCHITECTURE

This section presents the proposed navigation system for indoor mobile robot, that integrates the PCA-based self-localization system from ceiling depth images (see Fig. 1). This self-localization system implements the PCA algorithm in depth images acquired from the ceiling to estimate the robot position (see [8] for more details). Such system requires the environment previous mapping, placing the robot in specific positions along a grid, covering all free space, to capture a set of ceiling depth images. The acquired images are converted to an eigenspace, through the PCA algorithm, creating a database with high compression ratio, and stored on-board the robot.

During the motion, the depth images captured from the ceiling are compared with the PCA eigenspace to achieve the robot positioning. The robot global pose (position and attitude) is estimated by two KF, that merge the position computed from the PCA algorithm and other motion sensors installed on-board. Thus, the system achieves a robot navigation under ceilings with some information (e.g. industrial-like), using only the information provided from the on-board sensors and the knowledge about the environment, stored in a database with an high compression ratio. This approach allows to drive a differential-drive robot without the need to extract specific features from the environment and model linearizations.

The ceiling mapping and the PCA eigenspace computation is performed only once, by the capture of ceiling depth images along a grid, covering all free space (see [8] for details). Then, an occupancy grid is defined to configure the robot workspace, where cells are marked as free or occupied, according with the ceiling depth image has been captured or not in the corresponding grid position. This grid is useful to compute a roadmap into the robot free space, represented by a GVD.

Before the robot starts its motion towards a goal, a path planner (Dijkstra algorithm) computes the shortest route on the roadmap between these two positions. The reference path will be defined by a set of waypoints, corresponding to the GVD vertices found along this route.

To start the mission, the first waypoint is considered as a temporary goal which the robot will have to reach. To move

the robot toward this goal, a DNF is implemented to generate a smooth path through the potential field gradient of a local world. In the implemented approach, the goal is the attractive field while the boundary of the local world, centered on it, is the repulsive field that repels the robot to the center. The DNF gradient is computed in a closed loop, being the mobile robot pose given by the estimates results provided by the PCA-based self-localization system. The robot motion is commanded by a controller, aiming to follow the DNF gradient. When the robot gets closer to the temporary goal, it is changed to the next one. The process is repeated until the mobile robot reaches the final goal.

## III. PLANNING USING A GVD-BASED ROADMAP

### A. GVD-based Roadmap Construction

The GVD [16] is a method to design a roadmap, composed by a set of points, denoted as Voronoi edges, equidistant to the two most nearby obstacles. When more than two edges intersect the same point, this corresponds to a meet point, equidistant to three or more obstacles. In the implemented approach, a set of GVD vertices positions are selected, in the planning step, defining the waypoints that must be followed.

The GVD building for discrete grid configuration spaces is computed by a distance grid, using the brushfire algorithm [16]. The brushfire is an approach used to calculate a grid distance to the closer obstacle, representing a virtual obstacles dilation until all the free space is filled. Implementing [11], the grid distance is computed by the application of a wave front propagation from the obstacles and the external workspace border along the free space. In parallel, the GVD is defined by the set of points where the “waves” propagated by the obstacles and environment boundary meet. The points where the wave front collides are equidistant to the both obstacles and, then, these are the points on the GVD [16]. When the wave front collisions results in multiple neighbors, the points selected to the GVD are defined through a thinning algorithm [17].

### B. Path Planning Using a GVD-based Roadmap

Path planning is essential for a robot to perform missions between two locations, using a map. Lets consider a robot located in a position  $\mathbf{q}_s \in \mathcal{Q}_{free}$ , intending to go towards a destination position  $\mathbf{q}_g \in \mathcal{Q}_{free}$ . The path planning aims a set of  $\mathbf{q} \in \mathcal{Q}_{free}$  points definition, connecting  $\mathbf{q}_s$  to  $\mathbf{q}_g$  [16].

In the proposed navigation system (Fig. 1), the path planning toward the destination is a task executed before the robot starts the corresponding motion. The path is planned by the search for one that, using the roadmap the bulk of planning, minimizes the distance from  $\mathbf{q}_s$  to  $\mathbf{q}_g$ . Such shorter path is computed resorting to the Dijkstra algorithm [18]. The points connection distributed along a grid induces a graph, where the nodes correspond to cells and the edges are the connections between them [16]. Thus, the graph used by the Dijkstra algorithm is defined by all points connection that compose the GVD, being the weights established by the distances between them. The graph is completed by adding two nodes, corresponding to  $\mathbf{q}_s$  and  $\mathbf{q}_g$ . Each of these nodes are connected to another in the GVD, corresponding to  $\mathbf{q}'_s$  and  $\mathbf{q}'_g$ , respectively. Thus, the node corresponding to  $\mathbf{q}'_s$  is found by the euclidean distance minimization to  $\mathbf{q}_s$ . The corresponding

connection between  $\mathbf{q}_g$  and  $\mathbf{q}'_g$  is performed in a similar way. These weights are set to the corresponding euclidean distance.

The shorter path is composed by a set of edges connecting the start and goal nodes, where the sum of their lengths is the minimum. Once the Dijkstra algorithm computes the shorter path, using the GVD, a set of waypoints is defined to be used as a goal by the DNF-based controller. Thus, the GVD vertices positions along the planned path are extracted, defining the set of waypoints that the robot must to follow, and the goal position is added.

#### IV. NON-HOLONOMIC MOTION CONTROL USING A DIPOLAR NAVIGATION FUNCTION

In addition to path planning, a navigation system must to safely move the robot until its destination. This section presents the applied DNF-based controller to drive the robot motion toward each waypoint.

Consider a mobile robot with radius  $r$ , located in a position  $\mathbf{q} = [x \ y]^T$  with an attitude  $\psi$  and moving with a linear and angular velocity,  $u$  and  $\omega$ , respectively, towards a desired position  $\mathbf{q}_d = [x_d \ y_d]^T$  with an attitude  $\psi_d$ , inside a bounded circular workspace. NF are a type of APF, proposed by [19], ensuring only one minimum located into the goal. When a mobile robot with differential drive kinematics is controlled by a NF, the corresponding non-holonomic constraints must be taken into account. Lets consider a mobile robot represented by the configuration state  $\mathbf{p} = [\mathbf{q} \ \psi]^T$ , described by the following kinematic model:

$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & 0 \\ \sin(\psi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \quad (1)$$

Aiming this technique implementation in mobile robots with differential drive kinematics, [13] and [15] propose the use of NF with dipolar potential fields to provide suitable trajectories for those constraints. Based on these approaches, a DNF centered in  $\mathbf{q}_0$ , for the navigation of one differential-drive mobile robot, can be defined as follows:

$$\varphi = \frac{\gamma_d}{(\gamma_d^k + H_{nh}\beta)^{\frac{1}{k}}} \quad (2)$$

where  $\gamma_d = \|\mathbf{p} - \mathbf{p}_d\|^2$  is a function that represent the attractive potential field;  $\beta = -\|\mathbf{q} - \mathbf{q}_0\|^2 + \rho_0^2$ , with  $\rho_0 = r_0 - r$ , is a function that represent the repulsive potential field, defined by the environment boundary with radius  $r_0$ ;  $k$  is an integer positive tuning coefficient; and  $H_{nh}$  assume a pseudo-obstacle form that drives all trajectories to the desired pose,  $H_{nh} = \varepsilon_{nh} + \eta_{nh}$ , with  $\varepsilon_{nh}$  a positive coefficient,  $\eta_{nh} = \|(\mathbf{q} - \mathbf{q}_d)\mathbf{n}_d\|^2$  and  $\mathbf{n}_d = [\cos(\psi_d) \ \sin(\psi_d)]^T$ .

To drive a differential-drive robot using a DNF, this system implements the control laws proposed by [15]:

$$u = -K_u \text{sgn} \left( \frac{\partial \varphi}{\partial x} \cos(\psi) + \frac{\partial \varphi}{\partial y} \sin(\psi) \right) \left( \left( \frac{\partial \varphi^2}{\partial x} + \frac{\partial \varphi^2}{\partial y} \right) + \|\mathbf{q} - \mathbf{q}_d\|^2 \right) \quad (3)$$

$$\omega = \begin{cases} K_\omega (\psi_{nh} - \psi) & , \Delta < 0 \\ -K_\omega \frac{\partial \varphi}{\partial \psi} & , \Delta \geq 0 \end{cases} \quad (4)$$

where  $K_u$  and  $K_\omega$  are positive constants and:

$$\begin{aligned} \Delta &= K_\omega \frac{\partial \varphi}{\partial \psi} (\psi_{nh} - \psi) - \\ &- K_u \left( \left( \frac{\partial \varphi^2}{\partial x} + \frac{\partial \varphi^2}{\partial y} \right) + \|\mathbf{q} - \mathbf{q}_d\|^2 \right) \\ &\left| \frac{\partial \varphi}{\partial x} \cos(\psi) + \frac{\partial \varphi}{\partial y} \sin(\psi) \right| \\ \psi_{nh} &= \text{atan2} \left( \frac{\partial \varphi}{\partial y} \text{sgn}((\mathbf{q} - \mathbf{q}_d)\mathbf{n}_d), \frac{\partial \varphi}{\partial x} \text{sgn}((\mathbf{q} - \mathbf{q}_d)\mathbf{n}_d) \right) \end{aligned} \quad (5)$$

$$(6)$$

#### V. EXPERIMENTAL RESULTS

##### A. Experimental setup

Aiming to experimentally validate the navigation system using the PCA-based localization system proposed by [8], a differential-drive mobile robot platform is equipped with a Kinect device, looking upward to the ceiling, a rate gyro, a digital compass and encoders coupled to the motors (Fig. 2). The experiments are performed in an industrial-like indoor environment, containing several systems, such as, electrical and HVAC systems, lights, technical gutters, among others. (Fig. 3). Thus, according with [8], the environment is mapped in manual mode. A set of 1571 depth images from the ceiling was captured along a grid with 0.3 m, placing the robot in different positions (keeping the same attitude) and covering all free space. The captured depth images are cropped with a circular mask (to allow the comparison with rotated images) and sampled with a compression ratio of 100 : 1 (see Fig. 4), and converted into a vector that will be added to the PCA eigenspace. Therefore, the corresponding PCA eigenvalues are analysed and selected the best 30 eigenvectors. Then, an occupancy grid is built, based on the positions where the ceiling images were captured. The GVD roadmap was computed using the wave front algorithm, and the GVD vertices were defined as the set of possible waypoints.

##### B. Navigation with wrong initial position estimation

Aiming to experimentally validate the proposed navigation system, it was performed an experiment. The robot departs from the position  $\mathbf{q}_s = [6.0 \ 3.3]^T$  m and attitude  $\psi_s = 0^\circ$  toward a goal  $\mathbf{q}_g = [12.0 \ 9.6]^T$  m with an attitude  $\psi_g = 180^\circ$ , but considering that the position estimator is initialize with wrong values. Thus, despite the real mobile robot departs from the position  $\mathbf{q}_s = [6.0 \ 3.3]^T$  m, the self-localization is

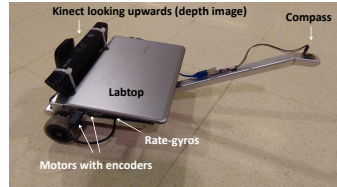


Fig. 2. Mobile robot platform equipped with a depth camera looking upward and other on-board sensors.



Fig. 3. Panoramic view of the ceiling.

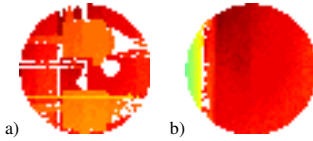


Fig. 4. Ceiling depth images after circular cropping and sampling process.

initialized with the position states  $\hat{x}(0) = 8.0$  m and  $\hat{y}(0) = 1.5$  m. To perform this mission, the points corresponding to these positions were connected to the corresponding closest points in the roadmap; the shortest path was computed through the Diskstra's algorithm; and the waypoints were selected according to this planning.

During the motion, the captured ceiling depth images are rotated, cropped with a circular mask and sampled, to be compared with the images captured in the mapping. The depth images signals are decomposed into the PCA eigenspace to obtain the robot position (no specific features extraction from the environment is required). Since the robot is moving in an environment with repetitive elements, the searching by similar images in the PCA eigenspace is performed considering a neighborhood radius,  $\delta = 3$  m, around the last robot estimated position. The results obtained from the PCA algorithm are merged with a Linear Parameter Varying (LPV), parameterized with the instantaneous angular velocity in the KF, to estimate the robot position (see [8] for more details). The robot attitude is estimated by a KF, based on [20], merging the signals obtained from a rate gyro, a compass and the encoders' odometry.

The robot motion results from a DNF defined on a local workspace, centered on the waypoint, and using it as a temporary goal position. The DNF is designed to drive the mobile robot toward its center, with a desired attitude in the direction to the next waypoint. To move the mobile robot toward each waypoint, the DNF gradients are computed for the estimated pose, and the controller defines the new linear and angular velocities, considering the gains tuned to  $K_u = 0.3$  and  $K_\omega = 0.5$ . Attending that the LPV model is designed for trajectories with constant velocity, a states replacement are made, aiming the position estimation with different velocities, i.e., the corresponding states are replaced by the new value,  $\hat{x} = u \cos(\hat{\psi})$  and  $\hat{y} = u \sin(\hat{\psi})$ . The commanded velocities are saturated with  $u_{max} = \pm 0.15$  m/s and  $\omega_{max} = \pm 1$  rad/s. Throughout the mission, when the distance error is less than

1 m, the temporary goal changes to the next waypoint and a new DNF is defined to drive the robot in direction to the next one. This process is repeated until the mobile robot reaches its destination, with a distance error less than 0.1 m.

To observe the real path described by the robot, while it was moving, some points were marked on the floor. After the experiment, these points were measured in the cartesian axes to obtain the ground truth. Figure 5 shows the robot trajectory and Fig. 6 depicts the results provided by the PCA-based positioning system and the position estimator along time. Analysing the results it is possible to observe that, despite the wrong estimated initial position, the self-localization system converge to the real trajectory and, once the position estimation is stabilized, the robot is correctly driven to its destination. Moreover, it can be observe that the travelled path succeeds in driving the robot closely to the path of largest clearance.

The self-localization system exhibits a rapid convergence to the real position because, after the robot starts the motion, the PCA algorithm finds images in the eigenspace similar to those currently captured, whose positions are close, i.e., the PCA algorithm finds the robot position. It is also possible to observe in Fig. 6 that the position estimator (KF) stabilizes in approximately 5 s and that the PCA algorithm estimates

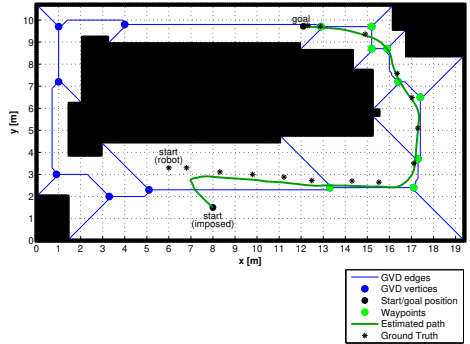


Fig. 5. Map with experimental results, starting from a wrong estimated initial position.

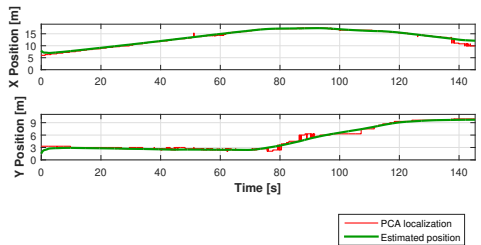


Fig. 6. Estimated position along time with wrong initial condition.

positions close to the provided by the self-localization system. Despite this, it is possible to observe some position errors provided by the PCA algorithm along the robot motion. This happens due to the distance associated with the database grid and because the PCA algorithm sometimes finds more similar images in distant locations. However, despite the variations in the positioning obtained from the PCA algorithm, the KF is able to smooth the estimated positions, ensuring a successful navigation towards the goal.

### C. Navigation along a large trajectory, describing two laps in a closed loop

Another experimental test is performed, considering the robot motion along a larger trajectory, in a closed loop. In this experiment, the robot departs from the position  $\mathbf{q}_s = [1.8 \ 10.0]^T$  m and attitude  $\psi_s = 180^\circ$  and navigates along a path with approximately 90 m, describing two laps in closed loop, toward a final goal  $\mathbf{q}_g = [1.0 \ 9.7]^T$  m with the same attitude ( $\psi_g = 180^\circ$ ). Since these positions are close, to ensure that the robot describes a larger trajectory in a closed loop, two intermediate goals are defined in the positions  $\mathbf{q}_{g1} = [6.0 \ 2.4]^T$  m and  $\mathbf{q}_{g2} = [12.9 \ 9.7]^T$  m, respectively.

To select the waypoints needed to reach each goal (intermediates or final), the points correspondent to the start and goal positions were connected to the corresponding closest points in the roadmap. The shortest path was computed through the Diskstra's algorithm and, finally, the waypoints were selected according to this planning. Since the path is composed by two intermediate goals ( $\mathbf{q}_{g1}$  and  $\mathbf{q}_{g2}$ ) and a final goal ( $\mathbf{q}_g$ ), the planning step is performed three times along the experiment. Thus, a first planning is computed to define the waypoints to drive the mobile robot to  $\mathbf{q}_{g1}$  and the DNF-based controller moves the robot toward this intermediate goal. When it is close to the first intermediate goal, another planning is computed from the robot current pose to the second intermediate goal,  $\mathbf{q}_{g2}$ , and so on until the robot reaches the final position,  $\mathbf{q}_g$ . Then, all process was repeated from the current robot estimated pose, in order to describe the second lap.

In this experiment it was considered that, along the motion, the temporary goal (waypoint) changes to the next when the corresponding distance error is less than 1 m, and the robot stops when the distance to the final goal position is less than 0.3 m. Aiming to check the real path described by the robot, along the experiment some points were marked on the floor and, after, measured in the cartesian axes.

Figure 7 shows the robot trajectory described along the experiment. It is possible to observe that the robot navigates along two laps, passing by the two intermediate goals and describing similar trajectories in both. Along the experiment, the robot position estimated by the self-localization system is close to the ground truth points the bulk of the trajectory. However, it is possible to observe that in the top of Fig. 7 the estimates are less accurate than the presented in the rest. These estimates are obtained in an area where the robot navigates under a ceiling with very few information, capturing images similar to the depicted in Fig. 4b, and even those with scenery repeatability. When this happen, the PCA algorithm has more difficulty in finding an image that was captured

in mapping with a position close to the real one. Then, the positions results obtained from the PCA algorithm have greater error and correspond to the instants 220-300 s and after 520 s (approximately) depicted in Fig. 8. Despite this, as can be observed in Figs. 7–8, when the robot returns to an area with rich information ceiling, capturing images similar to the depicted in Fig. 4a, the results provided by the PCA-based positioning system are again better. Nevertheless, even in the case where the PCA algorithm provides wrong positions, the KF estimator is able to smooth the trajectory, using the information provided from the motion sensors, improving the estimates results. Moreover, comparing the results of both laps, the positioning system provides similar results when the robot returns to the same place and captures depth images that can be more easily distinguished (under rich information ceilings). However, this is performed without the need to recognize specific features from the environment that was previous viewed.

Analysing the results, it is possible to conclude that the integration of the PCA-based self-localization system in a navigation system allows to move the mobile robot along a large trajectory in a closed loop, providing estimates close to the ground truth. Moreover, after a navigation with approximately

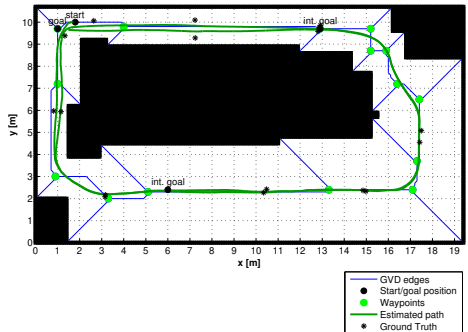


Fig. 7. Map with path following along a large trajectory with two laps in a closed loop.

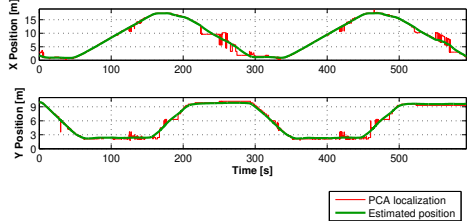


Fig. 8. Estimated position along time in a large trajectory with two laps in closed loop.

90 m, the robot stops close to the final goal with an error in the position estimated of  $e_x = 0.16$  m and  $e_y = 0.27$  m, relating to the real measured. This means that the robot reaches the destination with an error smaller than the ceiling mapping grid size, i.e., smaller than the position resolution provided by the PCA algorithm (0.3 m). This is achieved even finishing each lap with bad conditions, such as, capture ceiling depth images with few information (similar than Fig. 4b) and repeatability.

## VI. CONCLUSIONS

This paper presented the experimental validation of a navigation system for mobile robots, self-locating only from the PCA of ceiling depth images and other on-board sensors, merged by linear KF. In the implemented approach, the information about the captured depth images for the PCA database was used to define an occupancy grid. Such grid was useful to build a roadmap based on a GVD, from which a set of waypoints to guide the robot can be obtained. To perform a mission, the set of waypoints was selected, corresponding to the GVD vertices along the shorter path, computed by the Dijkstra algorithm. The DNF-based controller implementation allows to drive a mobile robot with differential-drive kinematics close to the roadmap.

In order to analyse the proposed system, some experiments were performed in a real environment, using a mobile robot platform equipped with on-board sensors. The experimental validation was performed considering navigation: i) via waypoints between two locations with wrong initial estimates; ii) along a large trajectory, describing two laps in a closed loop.

The experimental results showed that the proposed approach is able to stabilize the estimations and to move the robot toward its destination, along a clearance path, providing estimates close to the real trajectory. With the proposed system, the mobile robot can navigate, localizing itself only with the information provided from motion sensors installed on-board and the comparison of ceiling depth images with a database previous collected, through the PCA algorithm. In addition, when closed loop trajectories are considered, the robot can know that it returned to the same place, without the need to recognize specific features from the environment that was previous viewed.

As future work the self-localization system integration with a ground truth system, will allow to analyse the estimated error in real-time and with more precision. Another issue is the system expansion to multi-robots navigation.

## ACKNOWLEDGMENT

This work was supported by FCT, through IDMEC, under LAETA, project UID/EMS/50022/2013. This work was partially supported by the Polytechnic Institute of Lisbon through the funds given to the research project IPL/2017/PROMPT2/ISEL.

## REFERENCES

- [1] C. M. Costa, H. M. Sobreira, A. J. Sousa, and G. M. Veiga, "Robust 3/6 dof self-localization system with selective map update for mobile robot platforms," *Robotics and Autonomous Systems*, vol. 76, pp. 113–140, 2016, doi: 10.1016/j.robot.2015.09.030.
- [2] S.-Y. Hwang and J.-B. Song, "Monocular vision-based slam in indoor environment using corner, lamp, and door features from upward-looking camera," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4804–4812, Oct 2011, doi: 10.1109/TIE.2011.2109333.
- [3] W. Huang, C. Tsai, and H. Lin, "Mobile robot localization using ceiling landmarks and images captured from an rgb-d camera," in *Proceedings of AIM 2012, the IEEE/ASME International Conference on Advanced Intelligent Mechanisms*, Kachsiung, Taiwan: IEEE, Jul 11–14 2012, pp. 855–860, doi: 10.1109/AIM.2012.6265979.
- [4] F. Dias, H. Schaffer, L. Natal, and C. Cardeira, "Mobile robot localisation for indoor environments based on ceiling pattern recognition," in *Proceedings of ICARSC 2015, the IEEE International Conference on Autonomous Robot Systems and Competitions*, Vila Real, Portugal: IEEE, May 8–10 2015, pp. 65–70, doi: 10.1109/ICARSC.2015.32.
- [5] W. Yuan, Z. Li, and C.-Y. Su, "Rgb-d sensor-based visual slam for localization and navigation of indoor mobile robot," in *Proceedings of ICARM 2016, International Conference on Advanced Robotics and Mechatronics*, Macau, China: IEEE, Aug 18–20 2016, pp. 82–87, doi: 10.1109/ICARM.2016.7606899.
- [6] P. Oliveira, "MMAE terrain reference navigation for underwater vehicles using PCA," *International Journal of Control*, vol. 80, no. 7, pp. 1008–1017, Jul 2007, doi: 10.1080/00207170701242515.
- [7] F. Carreira, C. Christo, D. Valério, M. Ramalho, C. Cardeira, J. M. F. Calado, and P. Oliveira, "2d pca-based localization for mobile robots in unstructured environments," in *Proceedings of IROS 2012, the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal: IEEE, Oct 7–12 2012, pp. 3767–3868, doi: 10.1109/IROS.2012.6386272.
- [8] F. Carreira, J. M. F. Calado, C. Cardeira, and P. Oliveira, "Enhanced pca-based localization using depth maps with missing data," *Journal of Intelligent & Robotic Systems*, vol. 77, pp. 341–360, 2015, doi: 10.1007/s10846-013-0013-6.
- [9] S. Garrido and L. Moreno, "Mobile robot path planning using voronoi diagram and fast marching," *Robotics, Automation, and Control in Industrial and Service Settings*, pp. 92–108, Apr 2015, doi: 10.4018/978-1-4666-8693-9.
- [10] P. Bhattacharya and M. L. Gavrilova, "Roadmap-based path planning using the voronoi diagram for a clearance-based shortest path," *IEEE Robotics & Automation Magazine*, vol. 15, no. 2, pp. 58–66, Jun 2008, doi: 10.1109/MRA.2008.921540.
- [11] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–649, Dec 1991, doi: 10.1177/027836499101000604.
- [12] M. Kuderer, C. Sprunk, H. Kretschmar, and W. Burgard, "Online generation of homotopically distinct navigation paths," in *Proceedings of ICRA 2014, the IEEE International Conference on Robotics and Automation*, Hong Kong: IEEE, May 31 – Jun 7 2014, pp. 6462–6467, doi: 10.1109/ICRA.2014.6907813.
- [13] H. G. Tanner and K. J. Kyriakopoulos, "Nonholonomic motion planning for mobile manipulators," in *Proceedings of ICRA 2000, the IEEE International Conference on Robotics and Automation*, vol. 2, San Francisco, CA, USA: IEEE, Apr 24–28 2000, pp. 1233–1238, doi: 10.1109/ROBOT.2000.844767.
- [14] S. G. Loizou and K. J. Kyriakopoulos, "Closed loop navigation for multiple non-holonomic vehicles," in *Proceedings of ICRA 2003, the IEEE International Conference on Robotics and Automation*, vol. 3, Taipei, Taiwan: IEEE, Sep 14–19 2003, pp. 4240–4245, doi: 10.1109/ROBOT.2003.1242255.
- [15] S. Loizu, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Decentralized feedback stabilization of multiple nonholonomic agents," in *Proceedings of ICRA'04, the IEEE International Conference on Robotics and Automation*, vol. 3, Barcelona, Spain: IEEE, May 1 - Apr 26 2004, pp. 3012–3017, doi: 10.1109/ROBOT.2004.1307519.
- [16] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, Jun 2005.
- [17] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1991, vol. 1.
- [18] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, Dec 1959, doi: 10.1007/BF01386390.
- [19] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on robotics and automation*, vol. 8, no. 5, pp. 501–518, Oct 1992, doi: 10.1109/70.163777.
- [20] F. Carreira, J. M. F. Calado, C. Cardeira, and P. Oliveira, "Complementary filter design with three frequency bands: Robot attitude estimation," in *Proceedings of ICARSC 2015, the 15th IEEE International Conference on Autonomous Robot Systems and Competitions*, Vila Real, Portugal: IEEE, Apr 8–10 2015, pp. 168–173, doi: 10.1109/ICARSC.2015.33.