Robotica: (2020) volume 38, pp. 1000–1018. © Cambridge University Press 2019
doi:10.1017/S0263574719001206

Hybrid Methodology for Path Planning and Computational Vision Applied to Autonomous Mission: A New Approach

Fabrício O. Coelho†*, Milena F. Pinto†, João Pedro C. Souza‡ and André L. M. Marcato†

†*Electrical Engineering Department, Federal University of Juiz de Fora, Juiz de Fora, Brazil*
E-mails: milena.faria@engenharia.ufjf.br, andre.marcato@ufjf.edu.br
‡*Faculty of Engineering, Faculdade de Engenharia da Universidade do PortoPorto, Portugal.*
E-mail: joao.pedro@engenharia.ufjf.br

(Accepted June 24, 2019. First published online: July 25, 2019)

SUMMARY
In recent years, mobile robots have become increasingly frequent in daily life applications, such as cleaning, surveillance, support for the elderly and people with disabilities, as well as hazardous activities. However, a big challenge arises when the robotic system must perform a fully autonomous mission. The main problems of autonomous missions include path planning, localisation, and mapping. Thus, this research proposes a hybrid methodology for mobile robots on an autonomous mission involving an offline approach that uses the Direct-DRRT* algorithm and the artificial potential fields algorithm as the online planner. The experimental design covers three scenarios with an increasing degree of accuracy in respect of the real world. Additionally, an extensive evaluation of the proposed methodology is reported.

KEYWORDS: Autonomous missions; Artificial intelligence; Localisation; Mobile robots; Path planning.

1. Introduction

Mobile robots are becoming increasingly frequent in daily life activities. Usually, their tasks cover a wide range of applications. Examples include cleaning, surveillance, support for people with disabilities, as well as hazardous activities [1]. The greatest challenge in mobile robotics is to perform a complete autonomous mission with safety and without human interference. Therefore, several sensors are required to extract important information from the environment for decision-making during the missions. According to refs. [2, 3], some problems must be solved to accomplish the robot's missions autonomously: path planning, location, and motion control. Path planning ensures a feasible trajectory that connects the starting point to a goal. Additionally, path optimisation has a great importance in autonomous missions [4]. The location is necessary to indicate if the robot is fulfilling the assigned mission. Finally, motion control is needed to move the robot. It is important that these modules interact with each other, as proposed in ref. [5], to allow completely autonomous mission execution.

In general, robotic systems represent the world's knowledge for reaching the final goal. However, unexpected changes in the environment may occur during the mission. Therefore, hybrid approaches

* Corresponding author. E-mail: fabricio.coelho2010@engenharia.ufjf.br

for dynamic path planning (i.e., offline and online planners) are used to deal with real-time constraints. Besides, this kind of approach makes the response rapid and effective. In addition, the system will have a robust behaviour according to changes in the environment.

This paper proposes a new methodology for an autonomous mission in indoor environments. The offline planner is the Direct-DRRT* proposed by the authors in ref. [6] and based on the rapidly exploring random tree (RRT) [7]. The advantage of this method is its low computational cost and fast trajectory construction. In addition, the path generated by an algorithm based on the RRT respects the nonholonomic constraints of the robot. The online planner is the artificial potential fields (APFs). This method was chosen due to its efficiency, ease of implementation, and ability to control the robot during navigation. The obstacle detection is performed by the combination of an Red Green and Blue (RGB) depth camera and sonar sensors. The computer vision and encoders estimate the robot's localisation. The extended Kalman filter (EKF) method estimates the position of the robot using the information coming from camera and odometry sensors. The computer vision is responsible for extracting information from Ar Codes that are scattered throughout the environment. We chose to use a depth camera to follow recent trends related to robotics [8]. Light Detection and Ranging (LIDAR) sensors are used in several works related to robotics; however, their economic value is high, which forces researchers to replace them with depth cameras [8]. This methodology allows the recognition of substantial changes in the environment and makes the robot incorporate the new features on its map. The depth camera used in localisation also executed mapping during the mission.

1.1. Contributions

Since the methods used in the methodology presented in this paper are found in the literature, an important contribution of this research work is the definition of the optimised components of a hybrid methodology (Direct-DRRT* and the Potential Fields), and the localisation method to execute autonomous missions of mobile robots. Thus, this work can be used as a guide for these kinds of missions. The platform lays the groundwork for the development of a software aligned with the current requirements of state-of-the-art technology. This work's contribution can be summarised as follows:

- Development of a hybrid methodology to obtain an autonomous mission that supports the latest advances in state-of-the-art technology for mobile robots, that is, a complete, fast, and reliable offline path planner with robust response to changes in the environment and obstacles that can be placed or removed from the robot's view.
- A practical application of the path planner Direct-DRRT* proposed by the authors in ref. [6], another practical use of the localisation technique proposed by the authors in ref. [9] where the proposed methodology is validated.
- A solution proposal for the ambiguous problem of the localisation detected in ref. [9]. This paper presents a method based on homogeneous transformations using the angular position of the visual markers.

1.2. Organisation

The remainder of the paper is organised as follows. Section 2 presents a brief review of the related works highlighting the state-of-the-art in hybrid path planners and the mobile robot's localisation. The problem formulation and the basic concepts of the proposed methodology are detailed in Section 3. The theoretical foundations of the path planners, localisation, and mapping methods are addressed in Section 4. Section 5 shows the simulations and experiments with a proper discussion of the results. The concluding remarks are in Section 6.

2. Background and Related Works

Several studies have obtained relevant solutions regarding problems related to the autonomous mission. The number of problems to solve are extensive, and among those, localisation and path planning are still an active research topic due to their importance to the field. A quick review of these separate problems provides a picture of the current state of scientific research on the subject.

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

2.1. Path planners

The offline planner is characterised by a prior knowledge of the environment; that is, the robot knows all the static obstacles on the map. Generally, offline methods minimise the cost function related to the displacement between two states [10]. Contrarily, the online planner, also known as a reactive planners, consists of planning the path according to the results observed by the sensors. However, once the reactive planners is sensitive to local minima problems and more difficult to build for optimal solutions, the online method is used along with offline ones to solve this issue.

Given the advantages of offline and online methods, several studies have obtained satisfactory results by merging these two approaches and creating the hybrid path planners. The RRT* was used in ref. [11] to perform planning and re-planning. This sampling algorithm has the intention of obtaining an optimal offline path. However, during the mission, the path given by the trajectory may be unavailable due to the obstacles. Another application of RRT* is found in ref. [12]. In this work, the authors proposed an efficient method based on extended rapidly exploring random tree to allow the multi-robot team to quickly develop path plans due to the dynamics of the environment. Thus, this work executes one more time the RRT* algorithm in the modified map. Note that the RRT* has a high computational cost which may not be suitable to real-time online on-board application or environments with fast-moving objects. Other works, such as ref. [13], also tried similar methodologies.

Besides the RTT, other methods have been applied to this two-stage concept. As an example, in ref. [10], the offline planner was performed by the *k*-best algorithm that produced a trajectory composed of several paths while the online approach was based on the Hamilton–Jacobi–Bellman equations. Another hybrid approach was developed in ref. [14]. The mobile robot Jhonny performed autonomous missions in indoor environments. The robotic system was developed on a VolksBot modular platform [15]. This work used laser scans for localisation and cameras for object recognition.

Other works have tried to improve the RTT algorithm through some kind of sampling methodology. In ref. [16], the well-known reactive method APF [17] was applied. That work modified the sampling phase of the RTT to guide the random samples in the direction of the decreasing potential of the APF. However, that research may have problems of local minima. In ref. [18], another RTT/APF combination was presented. The authors developed solutions to narrow passages and to escape from local minima. However, the work stated the necessity of using sampling algorithms for dynamic environments. Thus, they presented the algorithm I-RRT*, which is an improvement of the RRT*. As a drawback, the article did not propose an interaction between the offline and online algorithms, and there was a lack of practical results.

The APF method is designed to attract the robot to the desired goal configuration and repel it from obstacles, which is due to the readings from the sensors, as shown in refs. [19–21]. Contrary to the RRT, the APF is fast enough to be applied on board of mobile robots in real time. As mentioned above, the operation of this path planner is responsive to dynamic obstacles in the environment. Also, the goal provides an attractive influence on the tree expansion, and the interaction of the sensor with the obstacles provides a reactive behaviour to the expansion of the tree. This feature makes the APF suitable to be applied in conjunction with the proposed Direct-DRRT* and, thus, is used in this research.

2.2. Mobile robot's localisation

The previous subsection focused on the algorithms related to path planning. This subsection highlights the main issues of localisation execution. The robot's position must be known throughout the path to reach an efficient autonomous mission. However, according to ref. [22], the localisation is corrupted due to errors generated by sensory noises, robot kidnapping, and environmental dynamism. Thus, several works have focused on localisation. The mobile robot needs to extract information from the environment through its sensor for self-localisation. The information may come from different sensors, such as laser scans, sonars, cameras, GPS, and encoders, among others. Another example is the odometry localisation used by the authors in ref. [23]. However, the errors inherent in these sensors accumulated over time. An alternative approach is to estimate the position based on the Monte Carlo problem as proposed in ref. [24]. Nonetheless, these methods require a high computational cost.

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

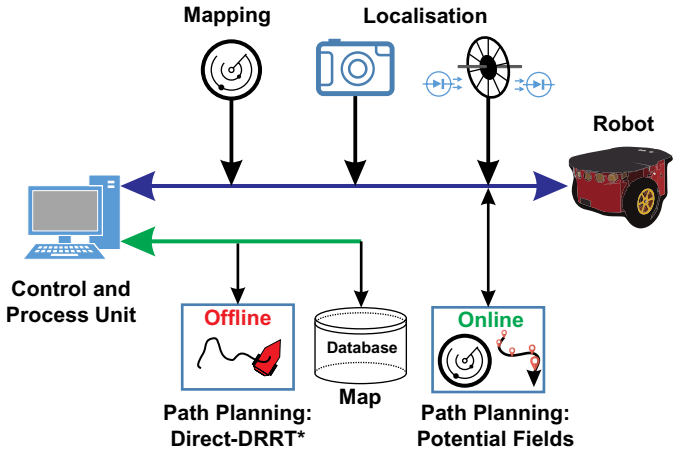

Fig. 1. Autonomous mission proposal.

An alternative to this issue that is extensively applied in the literature is the EKF used to perform filtering and sensory fusion. For instance, ref. [25] addressed several fusions, highlighting sonar fusion. The authors in refs. [26, 27] have an advanced implementation of EKF to create a robust, reliable robotic localisation and navigation system for autonomous robot in bridge deck inspection. The work of ref. [22] fused a sonar with velocity sensors in simulated environments. These works extracted characteristics using a sonar and they looked for matches on a map marker, which was saved in the robot database. Note that this match may cause problems, such as the wrong markers associated on the stored map. In ref. [28], the authors use the EKF to fuse information from a laser scan with an odometry sensor. The information extracted from the environment is the object's localisation with a high index of reflection. The association problem of these objects is emphasised since they are indistinguishable from each other. The association problem of these objects was underlined due to the fact that they are indistinguishable from each other. This research applies a similar methodology using the Kalman filter to handle sensor estimation and to combine with the Robot Operating System (ROS) package OpenNI 2 detection when it is available.

In ref. [28], a robot localisation method based on the EKF was presented. The current research is similar to the one proposed by ref. [28]. However, they used a Laser Scan sensor to detect reflective beacons. As mentioned above, due to the high cost of this kind of sensor, the computer vision is more economically viable. Also, the beacons are indistinguishable to each other, so the proposed localisation cannot be considered global. In our work, the localisation used does not need an initial position and it is robust enough to face cases of robot kidnapping as shown in the results section.

3. Problem Formulation and Robot Components Models

This section discusses the problem formulation and details the basic concepts of the proposed autonomous mission highlighting the robot, odometry, and camera models. The autonomous mission framework is based on the schematic of Fig. 1. The path planner Direct-DRRT* and the map directly interact with the Control and Processing Unit. This means that this process does not require any environmental information for functioning. The mobile system has a camera to assist during the localisation, mapping, and detection of obstacles. Note that the APF method is applied to avoid non-stored obstacles on the map.

3.1. Problem formulation

Autonomous navigation considers a set of factors, such as path planning, localisation, and mapping [2]. Thus, these main problems need to be addressed to perform the robot mission. The system dynamics is governed by Eq. (1).

$$\mathbf{X_t} = f(\mathbf{X_{t-1}}, u) \tag{1}$$

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

where $\mathbf{X_{t-1}} \in \mathbb{R}^n$ corresponds to the robot's past position, $\mathbf{X_t} \in \mathbb{R}^n$ is the actual position, and $u \in \mathbb{R}^m$ is the input vector (e.g., angular and linear speeds) in the robot's dynamics that is subject to the constraints present in Eq. (2).

$$u_{imin} \leq u_i \leq u_{imax},\ i \in \{1, \ldots, m\} \tag{2}$$

Obstacle and contour constraints can be found in Eqs. (3) and (4), respectively.

$$s(\mathbf{X}) \geq 0;\ \ s \in \mathbb{R}^k \tag{3}$$

$$\mathbf{X}(0) = \mathbf{X_0};\ \ \mathbf{X}(t_e) = \mathbf{X}_e \tag{4}$$

where k is the quantity of obstacles in the environment, and t_e is the mission's final time. This research assumes that the obstacles do not overlap and are not in the final goal ($\mathbf{X}_e$).

A problem solved by the proposed methodology is the path planning optimisation, which is presented in Eq. (5). The methodology uses the algorithm Direct-DRRT* where, given a determined configuration space C, navigable regions C_{free}, and obstacles C_{obs}, the path $\mathbf{L}$ will link the starting position $\mathbf{X_0} = q_{ini}$ with the final point $\mathbf{X_e} = q_{end}$. The Direct-DRRT* performs the optimisation problem analysing the relationship among the displacement costs (ρ) established in the environment. This is given by a Euclidean distance shown in Eq. (6).

$$min_u \int_0^{t_e} L(\mathbf{X}, u) \tag{5}$$

$$\rho : C \times C \rightarrow [0, \infty) \tag{6}$$

Path planning is also optimised in this work by applying the fusion of online and offline techniques for path planning. This aims at creating a hybrid system for the design of an efficient methodology. The path created by the Direct-DRRT* is defined by $\mathbf{L} \in \mathbb{R}^o$, where o is the number of coordinates capable of linking q_{ini} to q_{fim}. These coordinates are used by the APF method to make the robot reach the target position q_{end}. The APF method is defined by $\mathbf{U} \in \mathbb{R}^2$ ($\mathbf{U}$ is the potential field) and behaves according to the number of obstacles closer to the robot. The potential force F is found from this field. This one is responsible for guiding the robot, as shown in Eq. (7).

$$F(\mathbf{X}) = -\nabla U(\mathbf{X}) \tag{7}$$

Equation (7) corresponds to the negative gradient of the potential function. As it is a vector, the orientation obtained through the potential function needs to be analysed. Section 4 discusses this feature in detail.

3.2. Differential robot model

The model used to represent the robot's displacement provides the pose $\mathbf{X_t} = (x_t\ y_t\ \theta_t)^T$ from $\mathbf{X_{t-1}} = (x_{t-1}\ y_{t-1}\ \theta_{t-1})^T$, where v and ω are the linear and angular velocities, respectively. These control units are in respect of the robot frame (R).

The robot model and its behaviour are based on ref. [29]. The new pose is computed in constant time spaces (Δt) and depends on the wheel speeds and increments, as shown in Eq. (8).

$$\mathbf{X_t} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} \tag{8}$$

The increments Δx, Δy, and $\Delta\theta$ correspond to the displacements made by the robot. These displacements can be found by the algebraic relations between the robot frame (R) and the global frame (O). They are presented in Fig. 2.

Two restrictions on movement may be considered. The first one refers to the slip of the wheels, which corresponds to the angular displacements that will not generate the robot's displacements in the environment. The second restriction is the impossibility of the wheels' lateral movement; that is, there is only movement in the direction X_R or $-X_R$ in the robot reference. Equation (9) addresses these restrictions. The mathematical foundations are detailed in ref. [30].

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

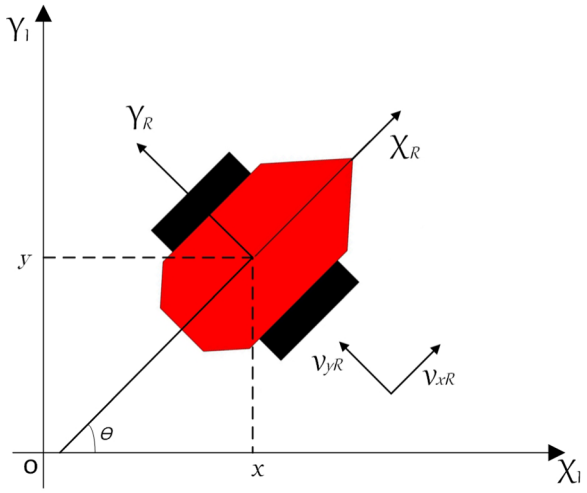

Fig. 2. Relationship between frames.

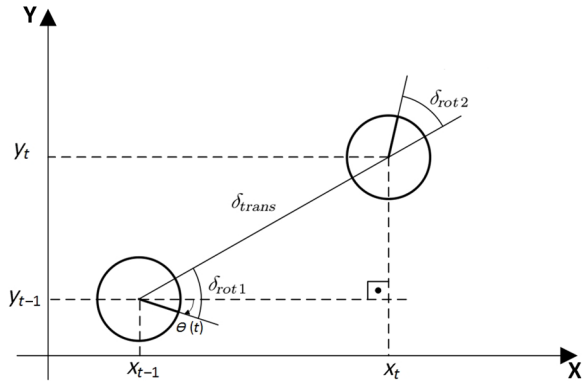

Fig. 3. Motion model (adapted from [5]).

$$\begin{bmatrix} \dot{x}_{t-1} \\ \dot{y}_{t-1} \\ \dot{\theta}_{t-1} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{t-1}) & 0 \\ \sin(\theta_{t-1}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{9}$$

where $[\dot{x}_{t-1}\ \dot{y}_{t-1}\ \dot{\theta}_{t-1}]$ represents the velocities in the global frame (O). Thus, the increments are extracted using the fraction of time in which the robot is submitted to the control units.

3.3. Odometry model

The P3DX robot has an odometry sensor coupled to the wheels to estimate the pose during the robot's movement. However, the data present noise, and the errors can spread along each iteration. Thus, another sensor is necessary to improve the measurement's quality. Note that motion information is defined only by the difference between the previous and current poses [31].

This motion model consists of slicing the displacement performed by the robot into three basic movements (see Fig. 3), first, a rotation (δ_{rot1}). Then, move (δ_{trans}) to the points x_t and y_t, and finally, another rotation (δ_{rot2}). We define the pose at time t as $\mathbf{X_t} = [x_t\ y_t\ \theta_t]$ and at time $t-1$ as $\mathbf{X_{t-1}} = [x_{t-1}\ y_{t-1}\ \theta_{t-1}]$. The three discretised motions are described according to the following equations:

$$\hat{\delta}_{rot1} = \tan^{-1}\left(\frac{y_t - y_{t-1}}{x_t - x_{t-1}}\right) - \theta_{t-1} \tag{10}$$

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

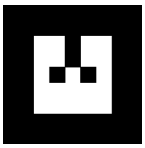

Fig. 4. AR Code example.

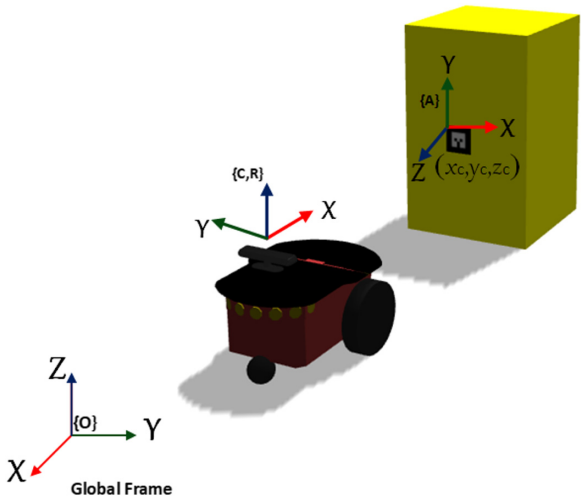

Fig. 5. Frames in the environment.

$$\hat{\delta}_{trans} = \sqrt{(x_{t-1} - x_t)^2 + (y_{t-1} - y_t)^2} \tag{11}$$

$$\hat{\delta}_{rot2} = \theta_t - \theta_{t-1} - \hat{\delta}_{rot1} \tag{12}$$

The model defined by the Eqs. (10)–(12) considers only the measurements of the odometry at time $t-1$ and t. The value observed by the odometry sensor $\mathbf{M_o}$ is modelled by (13), where $\mathbf{M_i} = (\hat{\delta}_{rot1}\ \hat{\delta}_{trans}\ \hat{\delta}_{rot2})^T$ and $\mathbf{E_o} = (\epsilon_{rot1}\ \epsilon_{trans}\ \epsilon_{rot2})^T$. The matrix $\mathbf{E_o}$ represents the Gaussian noises for each motion portion. The mean of each modelled noise is zero and the variances are equal to σ^2_{rot1}, σ^2_{trans}, and σ^2_{rot2}, respectively.

$$\mathbf{M_o} = \mathbf{M_i} + \mathbf{E_o} \tag{13}$$

Therefore, the function g is defined. This function has as input the control units (u) and the previous pose ($\mathbf{X_{t-1}}$) to obtain the future pose. Thus, it is possible to characterise the motion model from $\hat{\delta}_{rot1}$, $\hat{\delta}_{trans}$, and $\hat{\delta}_{rot2}$.

3.4. Camera model

This research uses the artificial features Augmented Reality (AR) Codes of Fig. 4 to estimate the position and the relative orientation of the robot, such as in ref. [32].

The localisation method addressed by this work is similar to the one presented in ref. [9]. However, this article also deals with ambiguity problems of an autonomous mission. This approach aims at finding the robot's actual position in the global frame. The methodology is based on the homogeneous transformations presented in Fig. 5.

Figure 5 presents the three frames: global {O}; camera {C} (i.e., same as the robot {R}); and the AR Codes {A}. When the AR Code is detected by the camera, the ROS package OpenNI 2 returns to the relative position of the camera in respect with the marker. Note that the AR Codes are fixed and their positions are known. In this way, it is possible to find the robot's position in relation to the global frame. Each transformation corresponds to the rotations and translations defined by algebraic matrices as used in ref. [33].

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

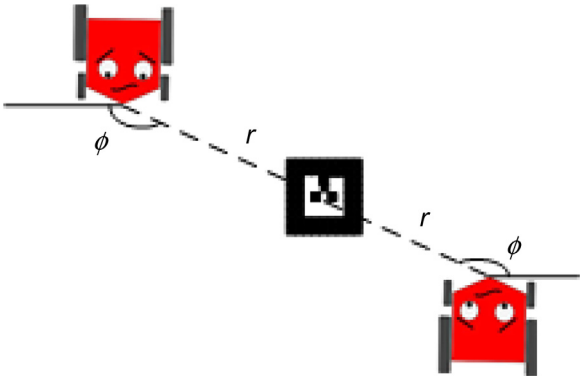

Fig. 6. Possible ambiguity.

After the transformations, the position in the global frame is obtained, and the camera provides the final position [34]. The study in ref. [9] did not consider that the marker angle was oriented by the global frame. Thus, the AR Code information obtained by the robot was the angle (ϕ) and the distance (r), as illustrated in Fig. 6. These variables are also extracted of package ROS OpenNI 2. Figure 6 shows two possible positions for the robot.

The mobile robot should have stored in its database a map of features of the marker positions to perform self-localisation. The map of features is $M_f = \{m_1, m_2, ...\}$, where each element corresponds to the AR Code pose $(m_{j,x}, m_{j,y}, m_{j,\theta})$. The parameter j is the bookmark index stored on the map M_f. The contribution of AR Codes in the localisation is the possibility of associating with artificial markers presented in a featured map once the codes distributed in the environments are unique.

Differently from ref. [9], in this paper, the camera modelling requires the angle where the marker is, that is, $(m_{j,\theta})$. This procedure inhibits ambiguities. At the end of this process, the algorithm defines the function h to find the vector with the robot localisation $\mathbf{Z}$ from the computer vision at time t, which is associated with the j contained in the map M_f. This is according to Eq. (14). Note that each coordinate has noises inherent to the sensors and they should be considered.

$$h = \mathbf{Z_t} = \begin{pmatrix} x_c \\ y_c \\ \theta_c \end{pmatrix} + \begin{pmatrix} \epsilon_{\sigma_x^2} \\ \epsilon_{\sigma_y^2} \\ \epsilon_{\sigma_\theta^2} \end{pmatrix} \tag{14}$$

Figure 7(a) exhibits the case of ambiguity. Note that there are two distinct positions that have minimal errors. Figure 7(b) shows the same result for the modelling using the homogeneous transform and considering the angular position of the marker in the global frame. Therefore, this represents the improvement of the work presented in ref. [9].

4. Path Planning, Localisation, and Mapping Methods

This section details the functioning of the path planners, localisation, and mapping methods of the autonomous mission methodology. The offline path planner used in the proposed framework is based on Direct-DRRT*. This method has the advantage of using less memory and computational time for a route design. As possible obstacles may arise during the robot's displacement, a reactive path planning method to avoid obstacles is necessary for many applications. Thus, the well-known APF is applied in this context. As new obstacles appear, the path found by the offline planner may turn unfeasible. In this way, the robot needs to take into account these new objects for future decisions by the use of mapping. Finally, the localisation uses the EKF method once the respective models are not linear.

The Direct-DRRT* is the offline method responsible for finding a set of coordinates that corresponds to the path $\mathbf{L}$ between two points. These points are the initial $\mathbf{X_0}$ and the final $\mathbf{X_e}$ positions as defined in Section 3. The method is derived by merging the techniques RRT*, DRRT, and Direct-RRT. The improvement provided by the Direct-DRRT* is presented in ref. [6]. In general, these methods are divided into two phases: expansion and definition. Basically, the RRT method consists of a tree expansion from a starting point that coincides with the current position $\mathbf{X_0}$. New elements,

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

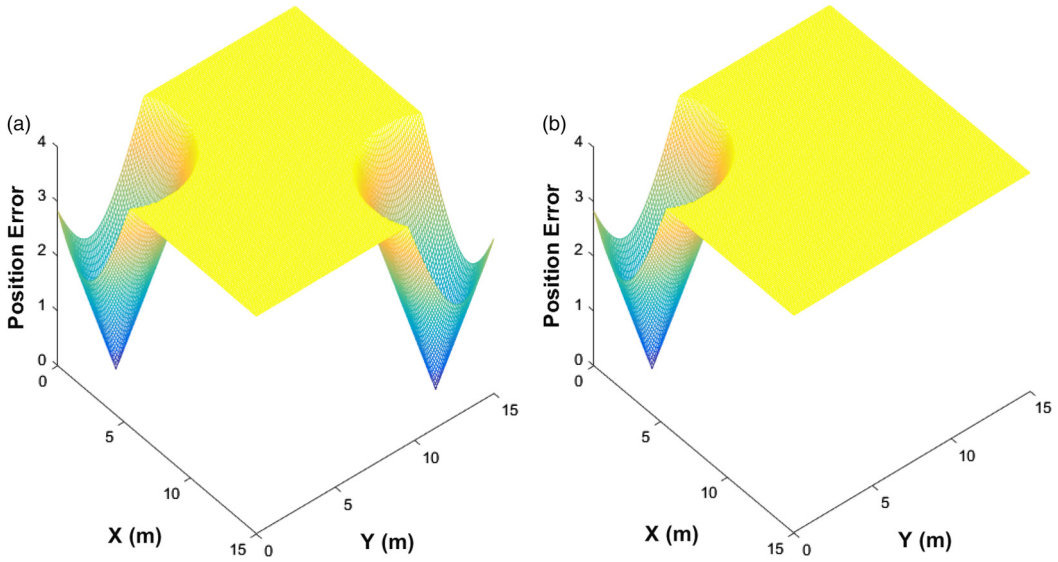

Fig. 7. Error of the positions obtained by each model. (a) Ambiguity. (b) New model.

called nodes, are incorporated into the tree by state sampling. A node can be defined as a data structure with the following information:

- **State**: the coordinate (x, y) that belongs to the settings space.
- **Relationship**: This informs which nodes already inserted in the tree connect to a determined node.
- **Cost**: this is the cost of moving from a node state to a tree root. In this problem, the costs are defined by Euclidean distances. In this way, it is possible to perform optimisation for finding the shortest path among all the tree nodes, as presented by Eq. (5).

The expansion stage ends when the node state that will be incorporated in the tree approaches the end position by a default Euclidean distance $\mathbf{X_e}$. Figure 8 exemplifies the tree expansion on a map. During this expansion, four stages of the tree were separated.

The path definition phase performs the path search through the costs and relationships that were defined in the expansion stage. Thus, path $\mathbf{L}$ is the smallest one, and it consists of the node states found in the second phase, which is also represented in four stages as shown by Fig. 9. The RRT is susceptible to the heuristics inserted for performance improvement. Thus, this method finds paths that would tend to an optimal as the number of samples increases. The algorithm performs a new analysis of the node's relationships that are inserted in the tree at each iteration.

Note that the search for optimal paths increases the computational costs. Thus, the fusion between the RRT* and DRRT methods has the advantage of decreasing this search time during the path creation. The DRRT creates circular exclusion regions around each state belonging to a node. As an example, if the sampling occurs within one of these regions, it is disregarded. Then, the DRRT reduces the number of nodes needed to find possible paths used to connect the starting position with the final one. Therefore, this fusion allows the use of the RRT* advantages with a shorter time.

However, the search time may still not be considered satisfactory. In this way, the results obtained by the fusion of the Direct-DRRT* method present a response with an efficient number of iterations, number of nodes, path lengths, and good computational time. A condition for the activation of the heuristic is verified in all iterations. In case this situation occurs, the samples with fixed states are carried out, highlighting the direct characteristics. Figure 10 details the process result.

Once the offline stage is finished, the online planner is applied. The APF method is used due to the advantages presented in the introduction. The model suggests that the robot and obstacles are compared to positive charges, repelling each other. The goal position (q_{end}) is characterised to the negative charges. Thus, the robot and goal position are attracted by each other. An advantage of

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

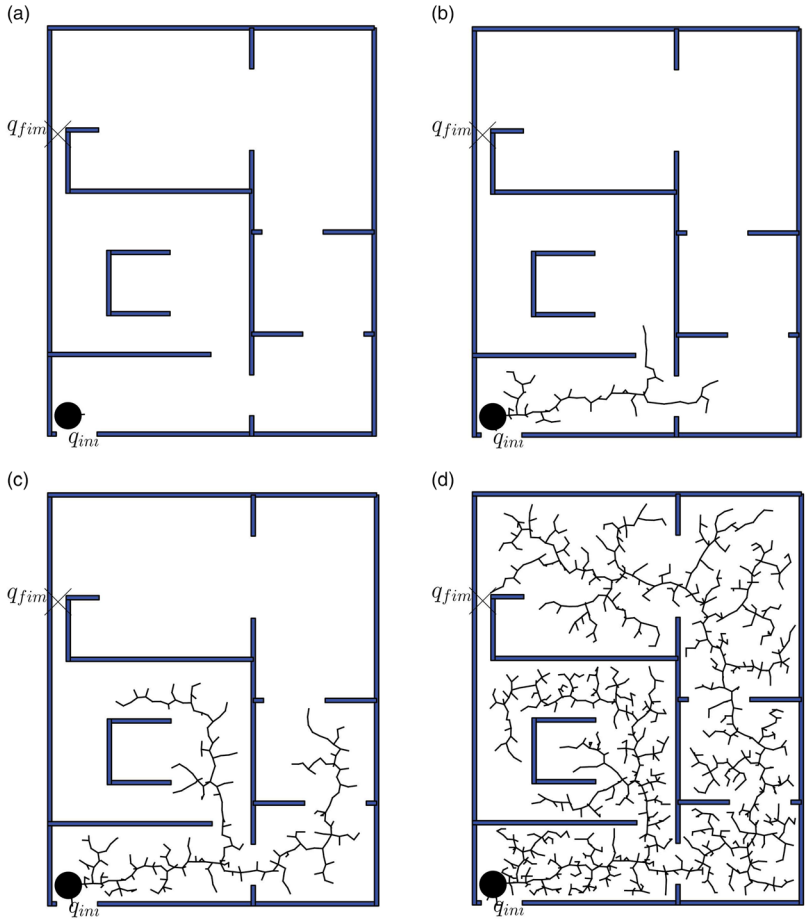

Fig. 8. RRT behaviour at the first phase: (a) Stage 1; (b) Stage 2; (c) Stage 3; (d) Stage 4.

using this method is the possibility of turning the APF the robot's controller. Although widely used, the APF is still a well-applied method in the robotics field ([35–38]). Figure 11 illustrates the APF functioning.

Mathematically, the interactions among the models are described by potential functions. The analysis of their gradients obtains the maximum and minimum points. The repulsive functions (U_{rep}) are required between the robot and the obstacles where their gradients correspond to vectors pointing in a contrary direction of the collision. On the other hand, the interaction between the robot and the target position is given by the attractive potential function (U_{att}), where the gradient results in a vector that points the robot to the goal position. The sum of these two potential functions is given by Eq. (15). The potentials U_{att} and U_{rep} are the attractive f_{att} and the repulsive f_{rep} forces, respectively.

$$U(q)_{tot} = U_{att} + U_{rep} \tag{15}$$

The obstacles are detected by a depth camera. This sensor is responsible for finding the artificial markers that are necessary for the localisation. Due to the small angle of view (AoV), this work uses the information from the sonars (i.e., 180º of AoV from the robot's front) to detect the obstacles. Only obstacles within a circular region with radius ε are considered. Figure 12 demonstrates the method's behaviour.

In this methodology, the mobile robot has a database with default obstacles. An example is presented in the Fig. 13(a). The new obstacles that are not on the map are symbolised by a yellow colour in Fig. 13(b). There is a remapping of all obstacles when the robot takes off towards the final positions, as shown in Fig. 13(c).

During the process, the proposed architecture applies a filter to verify whether or not the detected obstacle points belong to the stored map. With the remaining points, there is a classification in the clusters using the Euclidean distance among the points. Thus, there is the distinction between different objects, as shown in Fig. 13(d).

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

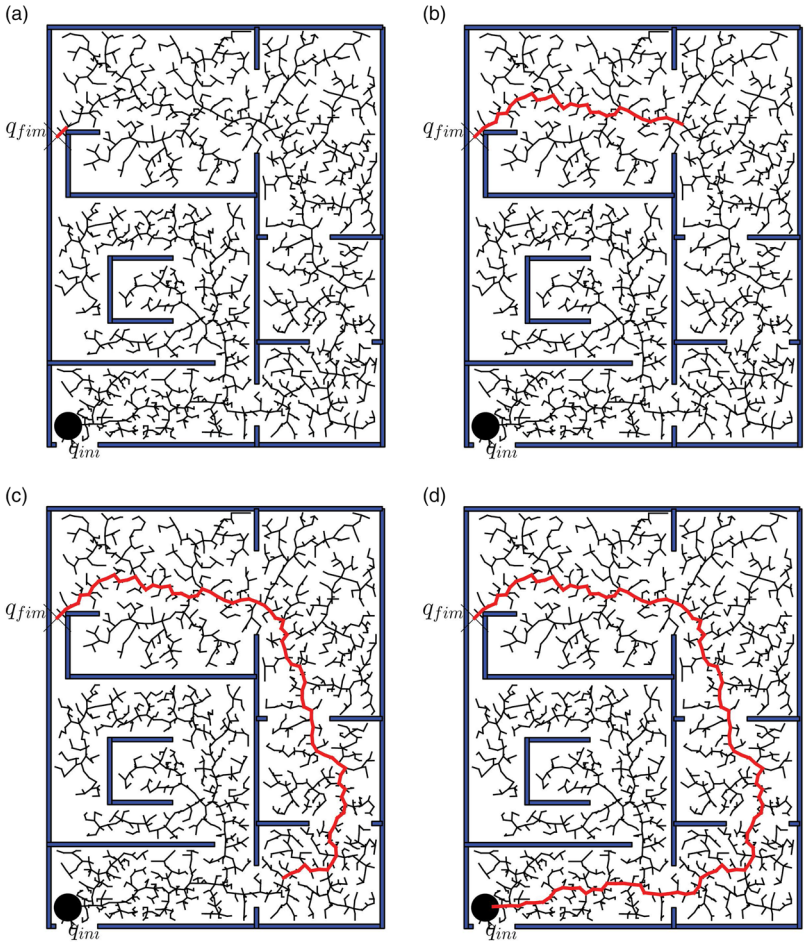

Fig. 9. RRT behaviour at the second phase: (a) Stage 1; (b) Stage 2; (c) Stage 3; (d) Stage 4.

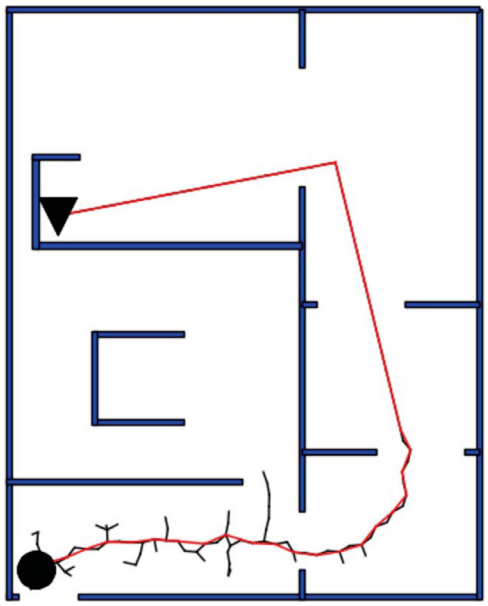

Fig. 10. Direct-DRRT* result.

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

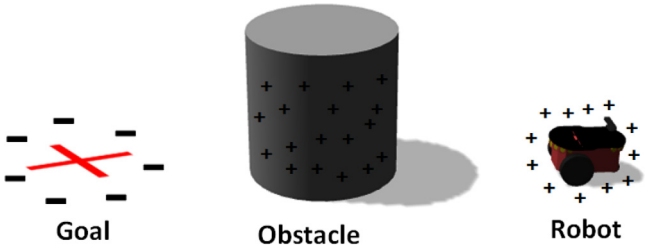

Fig. 11. Exemplification of the APFs.

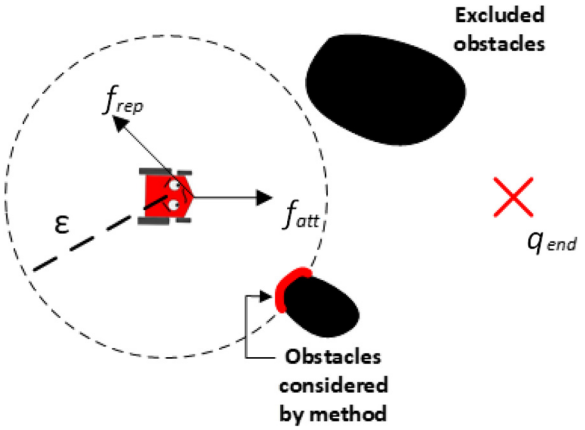

Fig. 12. APFs behaviour.

Finally, objects are transformed into rectangles as presented in Fig. 13(e). Then, they are saved in the new map as in Fig. 13(f). In this sense, the offline path planner can take into account the new obstacles in the next execution.

After defining the online and offline stages, the methodology should analyse how to improve the calculation of the current robot's localisation. An issue addressed in this work consists of the robot's current position estimation ($\mathbf{X_t}$) given by the perceived observations. This research performs this process by fusing the sensory information through the EKF. This algorithm is the localisation method. The EKF is divided into two steps: prediction and update. In this work, the prediction estimates the states using the odometry sensor. The update step corrects the state found in the prediction phase. The system, which EKF acts on, is specified in Eqs. (16) and (17) [31], and it defines the current robot's pose ($\mathbf{X_t}$) given the observations ($\mathbf{Z_t}$) made by the depth camera.

$$\mathbf{X_t} = g(u_t, \mathbf{X_{t-1}}) + \mathbf{E_o} \tag{16}$$

$$\mathbf{Z_t} = h(\mathbf{X_t}) + \mathbf{E_c} \tag{17}$$

It is necessary to use the odometry sensors to obtain the pose $\mathbf{X_t}$. The h function corresponds to the camera observation model. Both functions are non-linear and were defined in Sections 3.3 and 3.4. The variables $\mathbf{E_o}$ and $\mathbf{E_c}$ correspond to Gaussian noises of the odometry and observation, respectively. The variable u_t is the control unit.

The prediction phase analyses the odometry sensors that are performed using the robot's state information at a previous time. Since the models are non-linear, it is necessary to use the Jacobian functions to apply the EKF.

The update phase only exists if the camera has detected some AR Code. Otherwise, the prediction is considered the best sensor state estimation. In cases where the observation phase does not occur, the uncertainties related to the robot's state increase. This work does not present the EKF equation due to the vast existing literature applied to it. Details can be found in ref. [9].

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

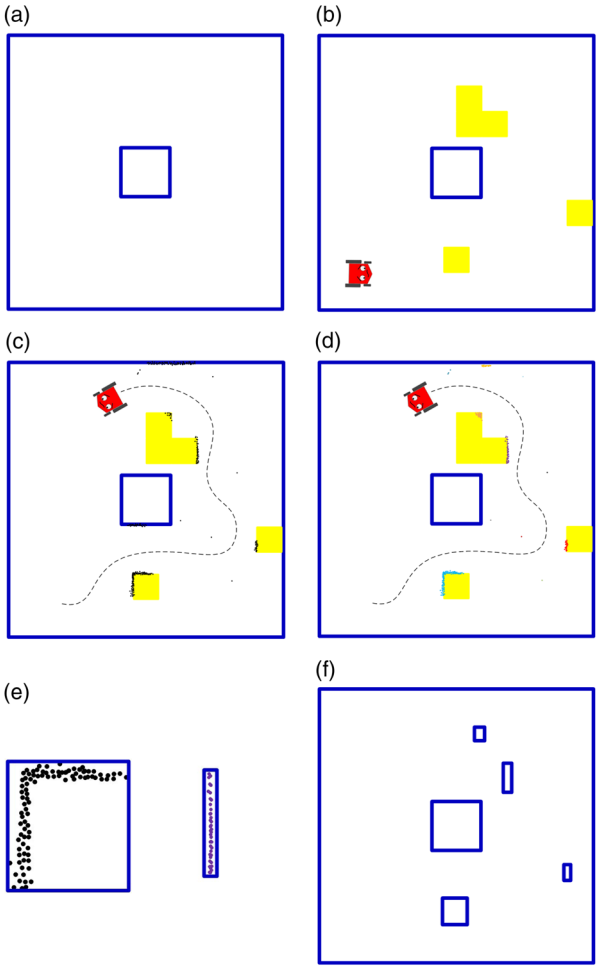

Fig. 13. Mapping process: (a) environment map; (b) dynamic obstacles in the environment; (c) movement and mapping performed by the mobile robot; (d) clustering process; (e) rectangles transformation; and (f) updated map.

Fig. 14. Environment layout for the experiments.

5. Results and Discussions

The experimentation methodology consists of moving the robot from a starting position to an ending point. In the experimentation design, 19 AR Code markers were spread throughout the environment. During the missions, the robot is subjected to different conditions to verify the autonomous capability to reach the final position safely. Figure 14 exhibits the environment.

This research uses the P3DX robotic platform [39]. The robotic system relies on a sonar and encoder sensors. The navigation is given by a pair of wheels with differential traction. The entire system is managed by the framework ROS. The RosAria package interfaces the actuators and sensors

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

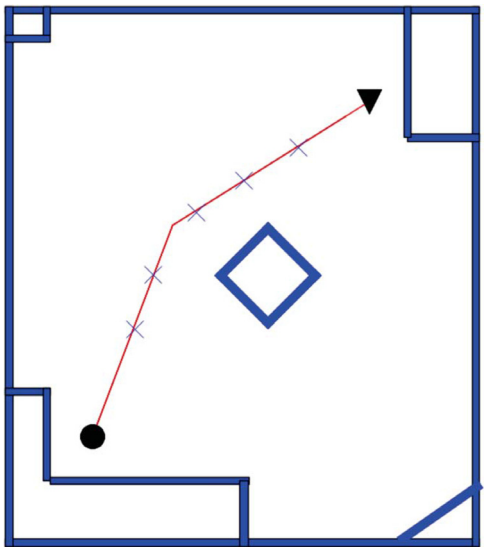

Fig. 15. Result for the Direct-DRRT*.

presented in the P3DX. For the methodology validation, three practical experiments were carried out: (1) the autonomous mission with dynamic obstacles; (2) the robot kidnapping; and (3) robot lost.

The mobile robot is coupled with a depth and RGB cameras that recognise the artificial markers spread around the environment for the robot's self-localisation. The depth camera assists in obstacle avoidance and maps the environment during the mission. The OpenNI 2 package enables the use of this sensor with the ROS framework. An on-board computer is responsible for the planning, localisation, and mapping of the environment.

Firstly, in the experiments, the offline algorithm Direct-DRRT* is executed to obtain a set of sub-goals so the robot can move from the initial to the final position. Figure 15 demonstrates this process. In this figure, the initial position is symbolised by ● ($x_f = 1.2$, $y_f = 1.5$) and the final position ($x_f = 5.0$, $y_f = 6.1$) by ▼. The markers in × are the sub-goals that the robot should move through to reach the final goal.

Three tests were performed with the sub-goals obtained by the Direct-DRRT*. The first case (**Case A**) explores the obstacle avoidance performance. The second one (**Case B**) analyses the robot's autonomous capacity during the kidnapping problem. The last case (**Case C**) tests the robot's ability to detect local minima in the environment and to complete the mission successfully.

Note that there is an impossibility of asserting the robot's position at all points during the missions. However, it is possible to affirm the initial and final positions that the robot reaches. The position given by the EKF method and odometry are compared in this work.

5.1. Case A: Dynamic obstacles

Case A presents a dynamic obstacle that was not previously considered on the map. The algorithm Direct-DRRT* is not able to avoid the obstacle. Thus, the APF method is applied. The obstacle is detected by a depth sensor and the APF controls the robot. This new obstacle was placed purposely on the top of the first sub-goal. Note that the robot detects the respective obstacle and then goes directly to the next sub-goal due to the localisation algorithm. The sensor detects the obstruction and the algorithm acts to avoid the collision. It is possible to observe that the localisation method reached a value closer to the odometry response. Figure 16 shows the obtained results of this case. The values of the final positions are given in Table I. This mission can be seen in the respective URL.[1]

5.2. Case B: Robot kidnapping

As previously mentioned, the second experiment addresses robot kidnapping. The set of sub-goals is the same as in the first case, that is, Case A. For a higher degree of autonomy, the robot should be

[1]https://youtu.be/muZ4PXzHC6I

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

Table I. Final positions for Case A.

Localisation	$\mathbf{x_f(m)}$	$\mathbf{y_f(m)}$
Odometry	4.837	6.147
Kalman	4.973	6.060
Real	4.960	6.100

Table II. Final positions for Case B.

Localisation	$\mathbf{x_f(m)}$	$\mathbf{y_f(m)}$
Odometria	−2.68	2.025
Kalman	4.983	6.080
Real	4.950	6.140

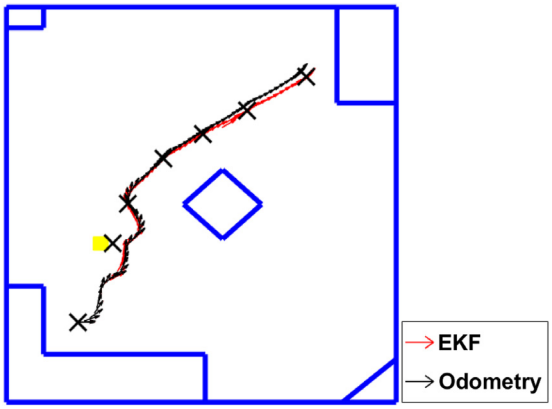

Fig. 16. Results for Case A.

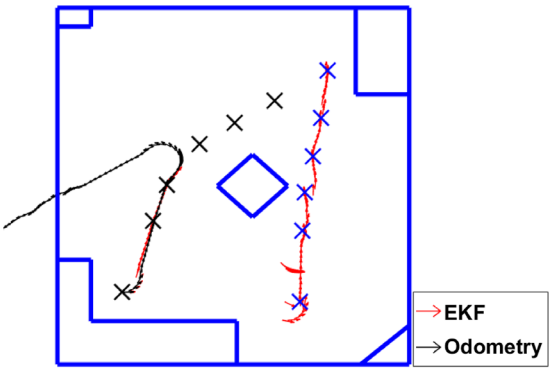

Fig. 17. Results for Case B.

self-localised to continue the mission. After the kidnapping problem detection by the robot, a new set of sub-goals are obtained through the Direct-DRRT*.

The importance of the AR Codes in the camera field of view comes from the possibility of properly correcting the robot's localisation. This means that in case the robot is deliberately moved to a new place without markers, the mission's success may be compromised. Moreover, the robot can enter in a slow-speed mode to prevent long-distance travels in unknown locations as the uncertainties of the EKF method increase. This allows more time for improving the localisation.

Figure 17 shows the path results given by the odometry and the EKF. The end positions reached by each of the methods can be found in Table II. The practical experiments with the results of Case B are accessed in the respective URL.[2]

[2]https://youtu.be/lWIRNjyQgaU

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

Table III. Final positions for Case C.

Localisation	$\mathbf{x_f(m)}$	$\mathbf{y_f(m)}$
Odometry	5.41	5.73
Kalman	4.97	6.080
Real	5.05	6.220

Fig. 18. Local minima given by the new obstacles.

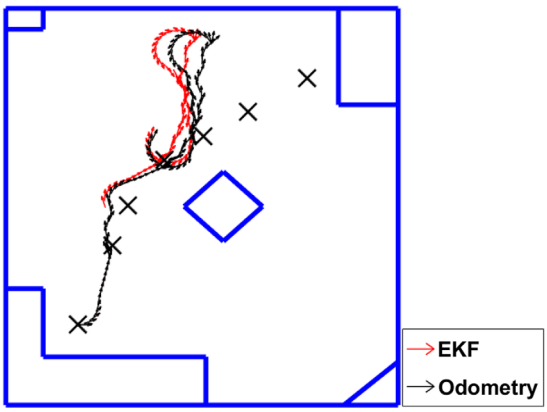

Fig. 19. Mobile robot in a local minimum.

5.3. Case C: Changes in the robot's environment (robot lost)

The last case tests the robot's skills in dealing with major modifications in the environment. These changes should be updated on the map since a simple deviation of obstacles is not possible. The obstacles were allocated in such a way that the robot could be trapped in a local minimum during the mission as presented in Fig. 18. Thus, the robotic system should perceive that the slow mode is activated to reach the respective sub-goal. Then, the robot should stop the mission and update the map. Therefore, a new set of sub-goals must be obtained from the Direct-DRRT*.

The positions obtained by the odometry and the EKF until the robot realises that it is trapped in a local minimum are presented in Fig. 19. The map is updated from a cloud of points, which is stored throughout the mission as shown in Fig. 20(a). The new map is presented by Fig. 20(b).

Given a new map, the mobile robot should execute the algorithm Direct-DRRT* for finding a new path to reach the final position. The result is shown in Fig. 21. The travelled path to reach the final position is shown in Fig. 22. The details are presented in Table III. The practical experiments with the results of Case C are accessed in respective URL.[3]

[3]https://youtu.be/R408gVJGuto

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

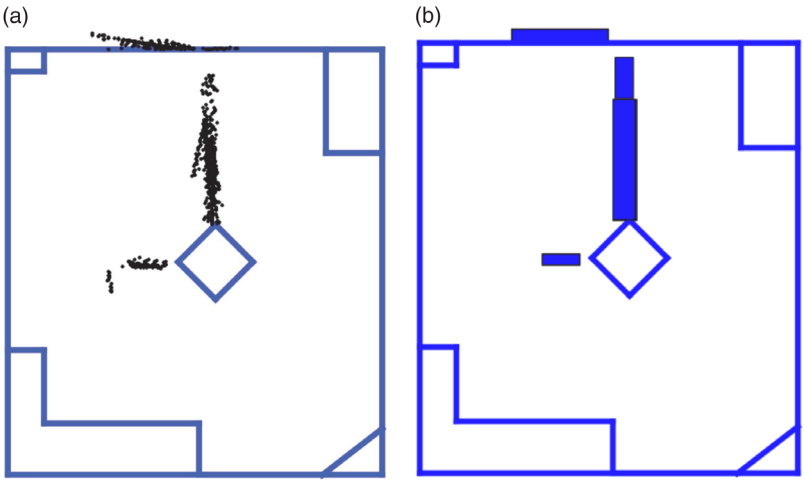

Fig. 20. The mobile robot is lost: (a) Cloud of points. (b) New map.

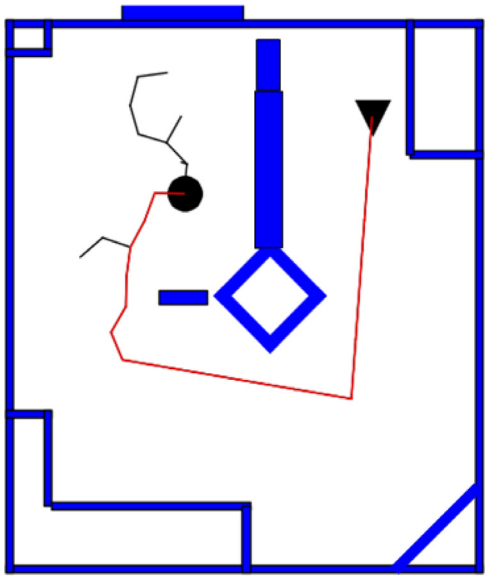

Fig. 21. New execution of the algorithm Direct-DRRT*.

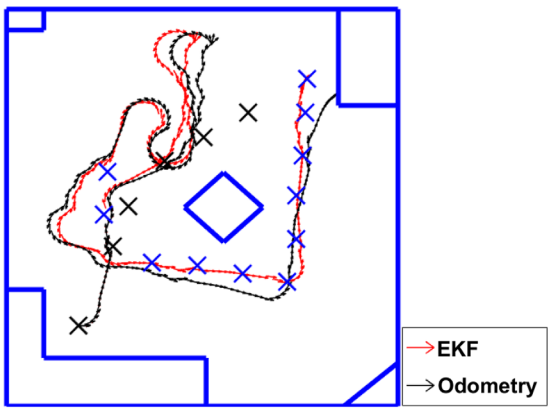

Fig. 22. Final path performed by the robot in Case C.

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press

6. Conclusions and Future Works

This research proposes an autonomous mission methodology for mobile robots involving localisation, mapping, and offline and online planners. The offline method is based on an algorithm that was previously proposed by the authors in ref. [6], namely, the Direct-DRRT*. The online method is based on the standard APF algorithm for obstacle avoidance. The localisation method is based on EKF that fuses odometry with the depth camera.

In addition to the detailed model description, an extensive evaluation is reported. The proposed methodology has proved sufficiency to move the robot from a starting point to an ending one under different established and unexpected conditions. As a general result, this work consolidates practical tests involving previously methodologies proposed by the authors for localisation, found in ref. [9], and for path planning presented in ref. [6]. The model covers three experimental scenarios with an increasing degree of accuracy in respect with the real world. The camera model presented in Section 3.4 used homogeneous transformations and considered the angle that the AR Codes are scattered environment. The proposed modelling in ref. [9] showed ambiguity errors, so it was not possible to reuse it in the practical tests, because if the EKF converts to the wrong position, the success' mission would be compromised.

A few extensions are foreseen from this research work. Firstly, a study could be undertaken regarding the minimum and the maximum number of AR Codes for this kind of mission. Secondly, an investigation could be initiated regarding the dispersion associated with the artificial markers throughout the environment, aiming at optimising them.

Acknowledgements

We would like to thank the Brazilian Federal Agencies CAPES, CNPq, FAPEMIG, UFJF, PPEE, ANEEL, and CTG Brasil for supporting this research.

References

1. D. Lattanzi and G. Miller, "Review of robotic infrastructure inspection systems," *J. Infrastruct. Syst.*, 04017004 (2017).
2. A. A. Panchpor, S. Shue and J. M. Conrad, "A Survey of Methods for Mobile Robot Localization and Mapping in Dynamic Indoor Environments," *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, Vijayawada, India (2018) pp. 138–144.
3. M. Nosrati, R. Karimi and H. A. Hasanvand, "Investigation of the*(star) search algorithms: Characteristics, methods and approaches," *World Appl. Program.* **2**(4), 251–256 (2012).
4. S. Tao and Y. Yang, "Collision-free motion planning of a virtual arm based on the fabrik algorithm," *Robotica* **35**(6), 1431–1450 (2017).
5. F. Khan, A. Alakberi, S. Almaamari and A. R. Beig, "Navigation Algorithm for Autonomous Mobile Robots in Indoor Environments," *Advances in Science and Engineering Technology International Conferences (ASET)*, Dubai (2018) pp. 1–6.
6. F. O. Coelho, J. P. Carvalho, M. F. Pinto and A. L. Marcato, "Direct-DRRT*: A RRT Improvement Proposal," *2018 13th APCA International Conference on Control and Soft Computing (CONTROLO)*, Ponta Delgada, Azores (2018) pp. 154–158.
7. S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," http://msl.cs.illinois.edu/~lavalle/papers/Lav98c.pdf (1998). Accessed 1st August 2017.
8. P. Tomasello, S. Sidhu, A. Shen, M. W. Moskewicz, N. Redmon, G. Joshi, R. Phadte, P. Jain and F. Iandola, "Dscnet: Replicating lidar point clouds with deep sensor cloning," arXiv preprint arXiv:1811.07070 (2018).
9. F. O. Coelho, J. P. Carvalho, M. F. Pinto and A. L. Marcato, "Ekf and Computer Vision for Mobile Robot Localization," *2018 13th APCA International Conference on Control and Soft Computing (CONTROLO)*, Ponta Delgada, Azores (2018) pp. 148–153.
10. Z. Shiller, "Off-line and On-line Trajectory Planning," **In:** *Motion and Operation Planning of Robotic Systems* (Carbone G., Gomez-Bravo F. (eds)) (Springer, Cham, 2015) pp. 29–62.
11. D. Connell and H. M. La, "Dynamic path planning and replanning for mobile robots using RRT," arXiv preprint arXiv:1704.04585 (2017).
12. D. Connell and H. Manh La, "Extended rapidly exploring random tree–based dynamic path planning and replanning for mobile robots," *Int. J. Adv. Robot. Syst.* **15**(3), 1729881418773874 (2018).
13. Z. Du and S. Liu, "Asymptotical RRT-Based Path Planning for Mobile Robots in Dynamic Environments," *2018 37th Chinese Control Conference (CCC)*, Wuhan, China (2018) pp. 5281–5286.
14. T. Breuer, G. R. G. Macedo, R. Hartanto, N. Hochgeschwender, D. Holz, F. Hegger, Z. Jin, C. Müller, J. Paulus, M. Reckhaus and J. A. A. Ruiz, "Johnny: An autonomous service robot for domestic environments," *J. Intell. Robot. Syst.* **66**(1–2), 245–272 (2012).
15. T. Wisspeintner, W. Nowak and A. Bredenfeld, "Volksbot–a Flexible Component-Based Mobile Robot System," **In:** *Robot Soccer World Cup* (Bredenfeld A., Jacoff A., Noda I., Takahashi Y. (eds)) (Springer, Berlin, Heidelberg, 2005) pp. 716–723.

16. A. H. Qureshi, K. F. Iqbal, S. M. Qamar, F. Islam, Y. Ayaz and N. Muhammad, "Potential Guided Directional-RRT* for Accelerated Motion Planning in Cluttered Environments," *2013 IEEE International Conference on Mechatronics and Automation*, Karlsruhe, Germany (2013) pp. 519–524.
17. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.* **5**(1), 90–98 (1986).
18. X. Xu, Y. Yang and S. Pan, "Motion Planning for Mobile Robots," **In:** *Advanced Path Planning for Mobile Entities* (IntechOpen, 2018).
19. D. S. Feirstein, I. Koryakovskiy, J. Kober and H. Vallery, "Reinforcement learning of potential fields to achieve limit-cycle walking," *IFAC-PapersOnLine* **49**(14), 113–118 (2016).
20. M. P. Faria, T. Mendonça, L. Olivi and A. Marcato, "A modified approach of potential field method for control of trajectory tracking and obstacle avoidance," *IEEE/IAS International Conference on Industry Applications (INDUSCON)*, Juiz de Fora, Brazil (2014).
21. P. Cui, W. Yan and X. Guo, "Path Planning for Underwater Docking Based on Modified Artificial Potential Field," *International Conference on Advanced Robotics and Mechatronics (ICARM)*, Osaka, Japan (2016) pp. 376–381.
22. D. Leite, K. Figueiredo and M. Vellasco, "Localização por kalman estendido aplicado a mapas baseados em marcos," **In:** *Simpósio Brasileiro de Automação Inteligente*, vol. 12 (2015).
23. P. Marin-Plaza, A. Hussein, D. Martin and A. D. l. Escalera, "Global and local path planning study in a ros-based research platform for autonomous vehicles," *J. Adv. Trans.* **2018**, 1–10 (2018).
24. F. Dayoub, T. Morris, B. Upcroft and P. Corke, "Vision-Only Autonomous Navigation Using Topometric Maps," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan (2013) pp. 1923–1929.
25. M. L. Fung, M. Z. Chen and Y. H. Chen, "Sensor Fusion: A Review of Methods and Applications," *2017 29th Chinese Control And Decision Conference (CCDC)*, Chongqing (2017) pp. 3853–3860.
26. H. M. La, R. S. Lim, B. B. Basily, N. Gucunski, J. Yi, A. Maher, F. A. Romero and H. Parvardeh, "Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation," *IEEE/ASME Trans. Mech.* **18**(6), 1655–1664 (2013).
27. H. M. La, N. Gucunski, K. Dana and S.-H. Kee, "Development of an autonomous bridge deck inspection robotic system," *J. Field Robot.* **34**(8), 1489–1504 (2017).
28. H. Sobreira, A. P. Moreira, P. Costa and J. Lima, "Robust mobile robot localization based on a security laser: an industry case study," *Indus. Robot Int. J.* **43**(6), 596–606 (2016).
29. P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, vol. 73 (Springer, Berlin, 2011).
30. I. F. Okuyama, M. R. O. A. Maximo, A. L. O. Cavalcanti and R. J. M. Afonso, "Nonlinear Grey-Box Identification of a Differential Drive Mobile Robot. *XIII Simpósio Brasileiro de Automação Inteligente (SBAI)*," Brazil (2017).
31. S. Thrun, W. Burgard and D. Fox, *Probabilistic robotics (intelligent robotics and autonomous agents)*, (MIT Press, Cambridge, MA, USA, 2005).
32. J. P. Carvalho, M. Jucá, A. Menezes, A. Marcato, A. D. S. Bessa and L. Olivi, "Landing a UAV in a dynamical target using fuzzy control and computer vision," *CBA2016* (2016) pp. 2636–2641.
33. J. Cashbaugh and C. Kitts, "Automatic calculation of a transformation matrix between two frames," *IEEE Access* (2018).
34. N. Rudy, Robot Localization and Kalman Filters on Finnding Your Position in a Noisy World *Ph.D. Dissertation* (Utrecht University, 2003).
35. M. F. Pinto, T. R. Mendonça, L. R. Olivi, E. B. Costa and A. L. Marcato, "Modified Approach Using Variable Charges to Solve Inherent Limitations of Potential Fields Method," *2014 11th IEEE/IAS International Conference on Industry Applications (INDUSCON)*, Juiz de Fora, Brazil (2014), pp. 1–6.
36. A. C. Woods and H. M. La, "A novel potential field controller for use on aerial robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **49**(4), 665–676 (2017).
37. S. Sharma, R. Sutton, D. Hatton and Y. Singh, "Path planning of an autonomous surface vehicle based on artificial potential fields in a real time marine environment," *COMPIT'17: 16th International Conference on Computer and IT Applications in the Maritime Industries*, Cardiff, UK (2017).
38. P. U. Lima, A. Ahmad, A. Dias, A. G. Conceição, A. P. Moreira, E. Silva, L. Almeida, L. Oliveira and T. P. Nascimento, "Formation control driven by cooperative object tracking," *Robot. Auto. Sys.* **63**(1), 68–79 (2015).
39. A. M. Robots, "Pionner 3-dx datasheet," http://www.mobilerobots.com/Libraries/Downloads/Pioneer3DX-P3DX-RevA.sflb.ashx (2011). Accessed in 09 August 2017.

https://doi.org/10.1017/S0263574719001206 Published online by Cambridge University Press