

# Towards Optimal Robot Navigation in Domestic Spaces\*

Rodrigo Ventura and Aamir Ahmad

Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST),  
Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal {yoda, aahmad}@isr.ist.utl.pt

**Abstract.** The work presented in this paper is motivated by the goal of dependable autonomous navigation of mobile robots. This goal is a fundamental requirement for having autonomous robots in spaces such as domestic spaces and public establishments, left unattended by technical staff. In this paper we tackle this problem by taking an optimization approach: on one hand, we use a Fast Marching Approach for path planning, resulting in optimal paths in the absence of unmapped obstacles, and on the other hand we use a Dynamic Window Approach for guidance. To the best of our knowledge, the combination of these two methods is novel. We evaluate the approach on a real mobile robot, capable of moving at high speed. The evaluation makes use of an external ground truth system. We report controlled experiments that we performed, including the presence of people moving randomly nearby the robot. In our long term experiments we report a total distance of 18 km traveled during 11 hours of movement time.

## 1 Introduction

Domestic spaces are significantly different from laboratory environments and office floors. Most of the robot navigation and path planning algorithms developed so far assumes the latter and are inherently designed for such controlled environments. People's homes vary not only with the cultural aspects of a country but also depending on individual choices. Consequently, it is quite difficult to generalize home environments. However, if the robot navigation algorithms accounted for certain recurring features in people's homes, e.g., presence of randomly moving humans, pets, unknown object, displaced pieces of small furniture as well as hard-to-perceive surfaces, robot motion would be perceived more natural in people's homes. In this context, it becomes necessary for such robots to not only execute an optimal path from a given start to goal pose in the environment but to also avoid previously-unmapped obstacles and randomly-moving people and pets.

In this paper we present a novel design for the motion of such robots in home-like environments by coupling an optimal path planning strategy with a navigation algorithm that inherently avoids previously-unmapped obstacles and randomly moving people. To this effect, we integrate Fast Marching Method [1] for path planning and dynamic window approach [2] to compute the motion velocity commands for the robot in real time that automatically avoids previously-unmapped static and dynamic obstacles in the environment. The novelty of this paper consists in the way these two methods are

---

\* Work supported by FCT projects PEst-OE/EEI/LA0009/2013 and FP7-ICT-9-2011-601033 (MONarCH).

integrated and implemented on a real robot. In addition, we performed experiments where the robot was expected to execute safe motion within human-occupied spaces. In order to evaluate the performance of our technique, we also implemented a ground truth system that estimates the robot's actual path during the experiments. Comparing the actual executed path by the robot with the optimal planned path will provide an insight into how well our proposed method behaved in real-time application.

The rest of the paper is structured as follows. Sec. 2 describes related work with this paper, followed by Sec. 3 where we describe the path planning, guidance and obstacle avoidance as well as their integration. This is followed by the robot description, details of the ground truth evaluation system and results of real robot experiments in Sec 4. We conclude with a remark on future work in Sec. 5

## 2 Related work

Domestic service robots operating in domestic spaces require various challenging functionalities, e.g., navigation, perception and manipulation, to accomplish a variety of tasks [3]. The human factor, in the context of developing such functionalities, is the most essential one. Consequently, new methods for robot navigation in the presence of humans are being studied extensively. Authors in [4] present a thorough survey of such methods where they identify comfort, naturalness and scalability as the three key issues addressed by the existing human-aware robot navigation methods so far. However, in most such methods, time and energy efficiencies of the robotic systems often get suppressed.

In [5] a method for human-centered navigation is presented which is based on various heuristics, such as, to maintain a given distance to the robot, to keep humans within certain visibility cone and to alter the robot velocity w.r.t. the human motion in the environment. In a similar direction of work, authors in [6] explore the possibility of robots learning the paths traversed by the humans, which is subsequently used by the robot's own path planner and obstacle avoidance system which predicts human trajectories. Such methods not only tend to become computationally heavy but also their reliance on human-centric heuristics might not hold true universally. To circumvent the need for heuristics some works have employed visual perception for real time feedback and continuous re-planning. Such a method has been used in [7] to solve complex human-robot cooperative tasks that includes navigation during the accomplishment of the whole task. However, visual perception in itself is a challenge that can seriously affect the optimality and robustness of the whole solution.

Some works, e.g., [8] and [9], have explored planning in 3D representations of environment maps leading to much robust navigation that could inherently account for the robot's full height and complete traversability in the environment.

More recently, some very successful approaches, e.g., [10] and [11], have made significant efforts in integrating efficient path planning and obstacle avoidance methods for navigation of domestic service robots in home environments. Apart from being robust and reliable, one very interesting property of these integrated methods is that they automatically account for human presence in the environment and do not depend on any heuristics. In this paper we not only subscribe to the aforementioned property but also focus on the optimality of the overall navigation of the robot in a home-like environment, in addition to being robust and reliable.

### 3 Navigation

Navigation in this context is understood as the capability of a robot to move autonomously in the environment with the goal of reaching a pre-specified final pose. The time taken by the robot to execute this task should be minimal, while avoiding collisions with obstacles as well as maintaining a certain clearance to them.

In this paper we take the classical approach of dividing navigation into self-localization and guidance, assuming knowledge of a map of the environment. We also assume that unmapped static or moving obstacles may appear in the environment, while the robot is expected to deal with them in an appropriate way. We further assume an existing self-localization system, possibly (but not necessarily) based on data fusion of odometry and range sensor matching with the map.

The guidance problem is approached as a two step process. First, given a goal location, the robot plans its path from the current pose to the goal pose. And second, the plan is executed by the robot, in real time, while avoiding unmapped obstacles. These two steps are described in the following two sections.

#### 3.1 Optimal path planning

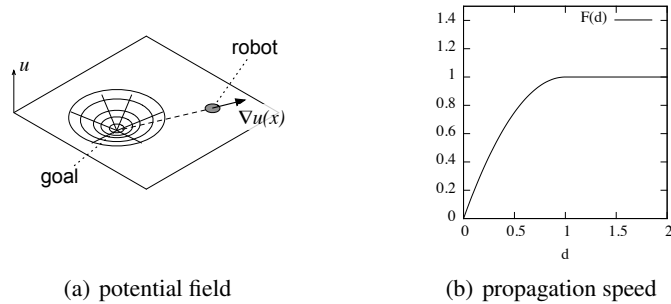
The path planning problem consists of determining a path for the robot to traverse the environment, given a map and a goal pose. Rather than explicitly planning for a path, we take a potential field approach. This potential field should have the property that, for any given robot location, the path resulting from following the gradient descent is the optimal path to the goal, while maintaining a certain clearance to the obstacles in the map. Such fields can be obtained using a Fast Marching Method approach applied to optimal path planning [1], which we closely follow. The process is explained next.

This potential field is obtained by considering, for each point  $x$  within the free region  $\Omega \subset \mathbb{R}^2$  of the map, the minimal time it takes for a wave to propagate from the goal location  $x_g$  to the current robot position. The computation of this time for each point  $x$  in the free region  $\Omega$  results in a field  $u(x)$ . Thus, the set of points that satisfy  $u(x) = T$  corresponds to the set of points belonging to the wave front at time  $T$ . This representation of a set is also known as *level set* [12]. It is well known that the path resulting from solving the ODE  $\dot{x} = -\nabla u(x)$  from an initial  $x(0) = x_0$  results in the optimal path from  $x_0$  to the initial wave front [1]. We set this wave front to an arbitrarily small ball  $\Gamma \subset \Omega$  around the goal location  $x_g$ . The relation between the goal, the field, and its gradient at the robot location is shown in Fig. 1(a).

Given an initial wave front  $\Gamma \subset \Omega$ , the field  $u(x)$  is the solution of the Eikonal equation

$$\begin{aligned} |\nabla u(x)|F(x) &= 1 \\ u(\Gamma) &= 0 \end{aligned} \tag{1}$$

where  $x \in \Omega$  is the free space of robot position,  $\Gamma \subset \Omega$  the initial wave front, and  $F(x)$  is the wave propagation speed at point  $x$ . The specification of this speed allows the resulting path to maintain a certain clearance to the mapped obstacles, since the optimal path tends to keep away from areas with lower propagation speeds. Note that this speed is anisotropical, that is, independent from the propagation direction of the wave front. It should also be noted that this speed is completely unrelated with the



**Fig. 1:** Aspects of the path planning method: (a) illustration of the field  $u(x)$  resulting from FMM, together with the gradient over that field starting at an arbitrary position; (b) plot of the wave propagation speed  $F$  as a function of the distance  $d$  to the nearest obstacle (for a  $C = 1$ ).

actual robot speed, which is specified elsewhere. The wave propagation speed is only used to promote clearance from mapped obstacles.

The field  $u(x)$  that solves the Eikonal equation has two key properties: (1) it shows no local minima, and (2) the gradient descent path is optimal, given a wave propagation speed function  $F(x)$ , *i.e.*, it is the smooth path  $\gamma(\tau)$  that minimizes the integral

$$\int_{x_0=\gamma(0)}^{x_g=\gamma(L)} F(\gamma(\tau)) d\tau \quad (2)$$

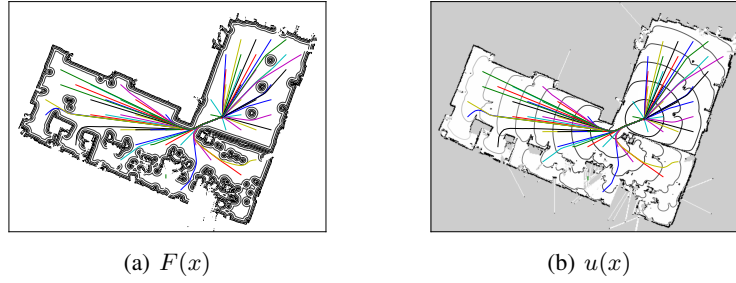
where  $\tau$  is the arclength parameterization of  $\gamma$ , that is  $\|\dot{\gamma}\| = 1$ , and  $L$  is the total length of the path [1].

To solve numerically the Eikonal equation we employ a standard Fast Marching Method: given a discretization of the map in an occupancy grid, we supply to this method the region of free space  $\Gamma$ , the propagation speed function  $F(x)$ , and the goal point, and in return we obtain a numerical approximation to the solution of the Eikonal equation on the grid points. The speed function  $F(x)$  is obtained by reducing the speed for the points near the mapped obstacles. Let  $d(x)$  denote the euclidean distance to the nearest mapped obstacle point. Then,

$$F(x) = \begin{cases} -\frac{1}{C}d(x)^2 + 2d(x), & d(x) < C \\ C, & \text{otherwise.} \end{cases} \quad (3)$$

where  $C$  stands for a threshold distance beyond which the wave propagation speed is constant (see Fig. 1(b)). Therefore, the smaller the  $C$ , the lower the clearance of the resulting path from mapped obstacles will be. This approach is quite similar to [13], however, we allow for a plateau in  $F(x)$  beyond a clearance distance  $C$ . This allows the robot to get closer to mapped obstacles, and thus resulting in shorter paths than in [13].

The resulting field  $u(x)$ , which is obtained once after a goal position is given, will then be used for the actual navigation. This navigation will aim at following a gradient descent of  $u(x)$ , while deviating from obstacles. Examples of several gradient descents over such a field, for the real scenario used for experimentation (Sec. 4), can be found in Fig. 2.



**Fig. 2:** Example of the propagation speed function  $F(x)$  and the field  $u(x)$  for the real scenario used for experimentation, as contour plots, together with a set of gradient descents from poses laid on a grid (within the scenario).

### 3.2 Guidance and obstacle avoidance

The goal of guidance is to compute in real time the robot actuation, in terms of motion velocity, given a FMM field  $u(x)$  embedding the optimal path to the goal. We solve this problem by taking a Dynamic Window Approach (DWA) [2,14]. That is, given the robot's current velocity, pose and available sensor data, DWA computes the next motion velocity command. It is done by formulating a constrained optimization problem over a discrete set of candidate velocity commands.

The outline of the algorithm is the following:

1. generate a set of candidate linear velocity commands
2. discard the velocity values beyond a specified maximum absolute value
3. discard the velocity values which could lead to a collision, that is, the robot is unable to stop, at the maximum de-acceleration, in time before hitting an obstacle
4. compute an evaluation value for each candidate by weighting three contributions: (i) progress towards the goal, (ii) clearance from obstacles, and (iii) absolute speed
5. select candidate maximizing the evaluation value
6. compute angular velocity based on the direction of the selected linear velocity, such that the robot front tends to be aligned with the motion direction.

This algorithm follows closely the DWA as initially proposed in [2], except for novel methods for both computing the clearance, taking into consideration the robot shape, and the progress, based on the potential field obtained from FMM. Next, we will describe each one of the algorithm steps in detail.

Let the initial set of candidate velocities for (discrete) time instant  $t$  be  $\mathcal{C}_0(t) = \{v^i(t)\}$ , for  $i = 1, \dots, N$ , expressed in the body frame. These candidates are assumed admissible, that is, they must comply with the kinematic constraints of the robot. In the case of an omnidirectional robot, as in the case of the targeted robot, we can independently control the motion velocity along these two directions. Thus, each candidate has the form  $v^i(t) = (v_x^i(t), v_y^i(t))$  representing the tangent and normal linear velocities. Otherwise, the actuation space has to be appropriately parametrized, e.g., in the of a differential drive, a possible parametrization is a linear and angular velocities pair. These velocities are chosen in a grid of values around the current robot velocity, within the acceleration limits of the robot. That is,  $\|v^i(t) - v^i(t-1)\| \leq A_{max}T$ , where  $A_{max}$

and  $T$  are the maximum linear acceleration and the period of the control loop (from here on, we will drop the dependence on  $(t)$  for the sake of clarity). The next candidate set  $\mathcal{C}_1$  contains the velocity candidates within the maximum linear velocity, that is

$$\mathcal{C}_1 = \{v \in \mathcal{C}_0 : \|v\| < V_{max}\} \quad (4)$$

where  $V_{max}$  is the maximum linear velocity.

Computation of both the clearance and the collisions, for each candidate, makes use of a 2-D point cloud obtained from the range sensors, e.g., a laser range finder. These points are here represented with respect to the body frame, being the union of the points perceived from all sensors:  $\mathcal{L} = \{p^j\}$ , for  $j = 1, \dots, M$ , where  $p^j = (p_x^j, p_y^j)$  are the point coordinates in robot body frame. Given a velocity candidate  $v^i = (v_x^i, v_y^i)$ , the point cloud is projected into a reference frame aligned with the candidate direction, that is, the tangent and normal unit vectors  $e_t^i$  and  $e_n^i$ , computed using

$$e_t^i = \frac{v^i}{\|v^i\|} \quad e_n^i = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} e_t^i \quad (5)$$

Thus, we get the projected points into this frame

$$p_t^{ij} = \langle p^j, e_t^i \rangle \quad p_n^{ij} = \langle p^j, e_n^i \rangle \quad (6)$$

where  $\langle \cdot, \cdot \rangle$  stands for the standard Euclidean inner product.

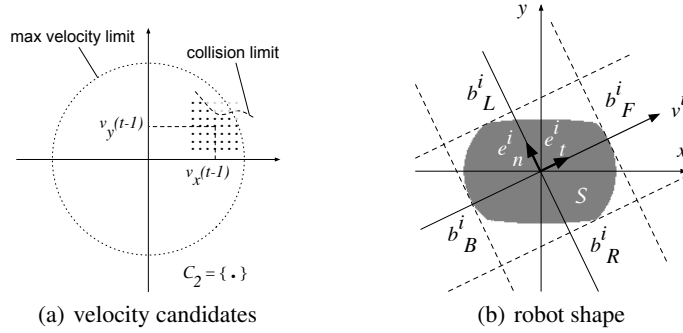
To compute the possibility of collision, we need to consider the physical space occupied by the robot, i.e., its shape. Rather than considering C-obstacles generated by the point cloud, which would require a computationally expensive convolution operation, we project the robot shape along its motion direction, and compare it with the obstacle point cloud. Let  $\mathcal{S}$  be the region of the space, on the robot body coordinate frame, occupied by the robot. Depending on the motion direction, the robot body will span a certain longitudinal and a certain lateral space. We approximate this span with a rectangular bounding box, aligned with the motion direction. That is, given a motion direction  $e_t^i$  associated with a velocity candidate  $i$ , a rectangular bounding box aligned to the  $(e_t^i, e_n^i)$  axes is determined. This bounding box is delimited by the points<sup>1</sup>  $(b_F^i, b_L^i)$ ,  $(-b_B^i, b_L^i)$ ,  $(b_B^i, b_R^i)$ , and  $(b_F^i, b_R^i)$ , expressed in the  $(e_t^i, e_n^i)$  frame. The relation between this bounding box, the robot shape, and a candidate velocity is shown in Fig. 3(b).

Assuming that the robot will be moving at speed  $\|v^i\|$  along direction  $e_t^i$ , given a maximum (de)acceleration  $A_{max}$ , the minimum stop time is  $T_{min}^i = \|v^i\|/A_{max}$  and the corresponding minimum stop distance is  $D_{min}^i = \|v^i\|^2/(2A_{max})$ . Thus, any obstacle within a distance of  $D_{min}^i$  of the robot front, across its extent, is here considered to result in a collision. Any candidate with obstacles meeting this criterion are discarded at this point. It should be noted that this criteria is an approximation, as we are approximating the spatial span of the robot by a bounding box. Formally, the new candidate set  $\mathcal{C}_2$  is obtained using

$$\mathcal{C}_2 = \mathcal{C}_1 \setminus \left\{ v^i \in \mathcal{C}_1 \mid \exists_j -b_R^i \leq p_n^{ij} \leq b_L^i \wedge 0 \leq p_t^{ij} \leq b_F^i + D_{min}^i \right\} \quad (7)$$

The selection of these candidates, which take into account both the maximum robot speed and possibility of collision, is illustrated in Fig. 3(a).

<sup>1</sup> Subscripts F, B, L, and R stand for front, back, left, and right.



**Fig. 3:** Illustration of some aspects of the guidance method: (a) the velocity candidates form a grid (dynamic window) around the current velocity value; candidates are excluded when either the robot maximum speed or found to lead to a collision with a detected obstacle, resulting on a candidate set  $C_2$ ; (b) how the bounding box relates to the robot shape  $S$  and the candidate velocity  $v^i$ , both expressed on the robot body reference frame.

Now, the next step is the computation of the evaluation value for each candidate. This evaluation function is a weighted sum of three contributions:

$$V_i = a P_i + b C_i + c S_i \quad (8)$$

where  $a$ ,  $b$ , and  $c$  are positive weighting coefficients, and  $P_i$ ,  $C_i$ , and  $S_i$  quantify the progress to goal, the clearance, and the speed of each candidate, respectively.

Progress to goal is computed from the inner product between the FMM field gradient and the velocity candidate, both expressed in the world frame. The field gradient is numerically computed from the field values on a neighbor of a cell  $(x, y)$  as

$$Du = \begin{bmatrix} \frac{u(x+1,y) - u(x-1,y)}{2h} \\ \frac{u(x,y+1) - u(x,y-1)}{2h} \end{bmatrix}, \quad (9)$$

where  $h$  is the grid size and  $x$  and  $y$  are the robot position coordinates, scaled and discretized to grid indices. The candidate velocity is normalized to maximum speed  $V_{max}$ , so that its relative contribution to the evaluation function is independent from the maximum speed scale. The gradient is normalized into a unit vector, so that its contribution to the progress to goal is independent from its absolute value (which, following (1), is  $1/F(x)$ ). Thus, the progress to goal contribution is

$$P_i = \left\langle \frac{Du}{\|DU\|}, \frac{v^i}{V_{max}} \right\rangle \quad (10)$$

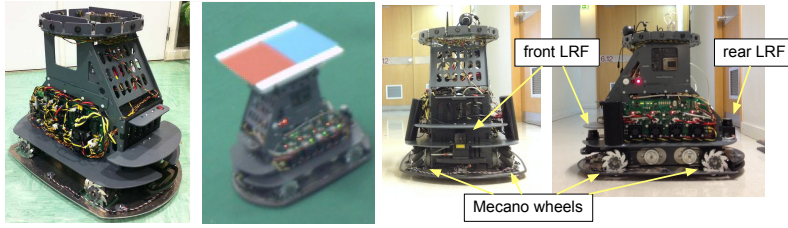
Clearance is computed, for each candidate, from the closest distance to an obstacle after moving along the candidate direction:

$$C_i = \min \|(p_t^{ij} - d_i, p_n^{ij})\| \quad (11)$$

where  $d_i = \|v_i\|T$  is the distance travelled along the candidate direction during the control loop period.

Finally, the absolute speed contribution is simply the absolute value of the candidate velocity

$$S_i = \|v^i\| \quad (12)$$



**Fig. 4:** MOnarCH robot platform: Exposed robot chassis and electronics (left-most). Robot with a color coded plate to facilitate its ground truth pose estimation (second from left). Side views of the MOnarCH robot platform, showing the position of each LRF (two images on the right side)

The optimal candidate results simply from the maximization of the evaluation values, that is

$$v^* = \arg \max_{v^i \in \mathcal{C}_2} V_i \quad (13)$$

Since the target platform is omnidirectional, we have complete freedom on the choice of the heading of the robot, provided that the direction of movement is covered by range sensors. This extra degree of freedom can be used, for instance, to convey expressiveness to the robot motion. Currently we are not exploiting this, and thus we use a simple heading controller, based on a feedback proportional gain with saturation

$$\omega = \max \{-\omega_{max}, \min \{K e_\theta, \omega_{max}\}\} \quad (14)$$

where  $K$  is the proportional gain,  $e_\theta = \arctan 2(v_y^*, v_x^*)$  is the angular deviation of the candidate velocity with respect to the robot heading, and  $\omega_{max}$  the specified maximum angular speed.

## 4 Experiments and Results

### 4.1 Robot Details

The path planning and navigation algorithms described in this paper were implemented on a 4-wheeled omni-directional robot platform (Fig. 4). This robot has been specifically developed for an ongoing European FP7 project: MOnarCH<sup>2</sup>. In addition to various other sensors and actuators as described in [15], it is equipped with two laser range finders (LRFs) which are used for mapping, navigation, and obstacle avoidance (as shown in Fig. 4). In particular, we use two LRF together for the guidance method: the point cloud  $\mathcal{L}$  consists of the union between the latest scan of each LRF, with the appropriate coordinate transformation to the common robot body reference frame.

The platform runs Linux Ubuntu and the integration middleware is the widely used ROS framework<sup>3</sup>. For mapping we use the gmapping package<sup>4</sup> and for localization we use the AMCL package<sup>5</sup>, both available out of the box from the base ROS installation. The former is an implementation of a Rao-Blackwellized particle filter [16], while the latter implements an Augmented Monte Carlo Localization (AMCL) method [17].

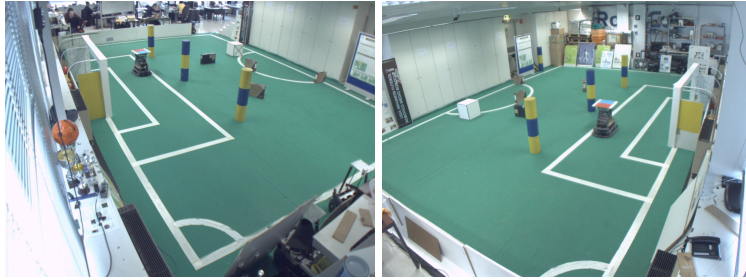
<sup>2</sup> Project reference: FP7-ICT-2011-9-601033

<sup>3</sup> <http://www.ros.org>

<sup>4</sup> <http://wiki.ros.org/gmapping>

<sup>5</sup> <http://wiki.ros.org/amcl>





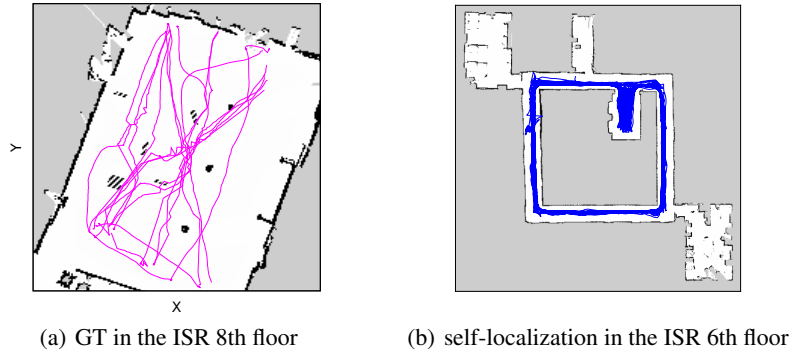
**Fig. 5:** Testbed in the 8th floor of ISR with the MONarCH robot during the experiment from the left and right cameras of the GT system. The blue/yellow poles were known obstacles included in the map, while all the other objects on the field were previously unmapped.

## 4.2 Ground Truth Evaluation

In order to evaluate the performance of our navigation algorithm, i.e., to infer how well the DWA-based navigation performs w.r.t. the path planned using the FMM (gradient descent along the potential field, as explained in subsection 3.1), it is necessary to compare the FMM-planned path with the actual path taken by the robot during the experiments. For the actual path taken by the robot, one cannot use the self-localization information as estimated by the robot itself because this information is one of the inputs to the DWA-based navigation algorithm. Therefore, re-using the self-localization information to evaluate the performance of navigation will be incorrect. Consequently, it is imperative that one must obtain the ground truth (GT) poses (position and orientation) of the robot during the experiment from a totally external GT system. To this end, we developed such a GT system as described further.

The hardware of the GT system to evaluate the pose of MONarCH robot is the same as the one described in [18]. It consists of two gigabit ethernet cameras in a stereo baseline approximately 13m apart. They are connected to a machine with Quad Core Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz, 8GB RAM, running a Linux operating system. The model of the camera is Basler acA1300-30gc with a maximum acquisition frame rate of  $\sim 25$  frames per second (fps) and a resolution of  $1294 \times 964$  pixels (1.2 megapixels). The robot and the GT system were time-synchronized using the network time protocol (NTP). In [18], this hardware was used to evaluate the 3D-position GT of a spherical-shaped object of known color and size therefore making the use of the stereo cameras. However, in this work we use this hardware to construct a GT system to estimate GT poses of the MONarCH robot. For this purpose, we placed a bi-colored plate (stuck with two adjacent, colored, A4-sized markers) on top of the robot such that the center of the plate was directly above the robot's center of mass and the line joining the center of each colored markers on the plate was aligned with the robot's heading. This arrangement can be visualized in Fig. 4 (second from left). As this robot moves on a fixed plane, we could safely assume that the height of the robot's plate above the ground plane was constant throughout the experiment. This height was pre-measured and used further in the GT pose evaluation of the robot.

Images from both the GT cameras (Fig. 5) were captured during the experiment and post-processed. The process pipeline then consisted of color-based image segmen-



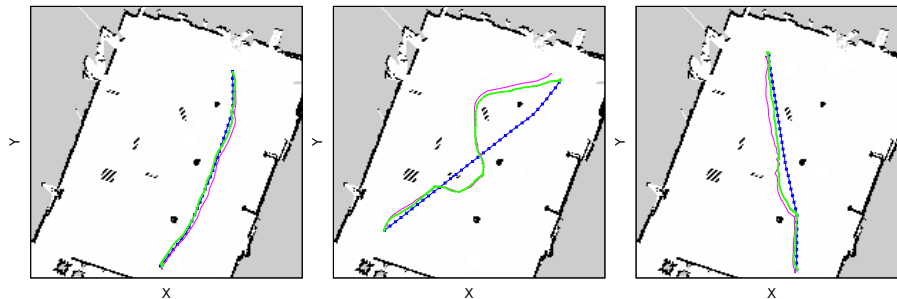
**Fig. 6:** Robot actual trajectories during experiments: (a) GT experiments, in the 8th floor of ISR which is also shown in the GT camera images of Fig. 5, and (b) a long 3 km test, according to the robot self-localization. The ragged lines on the left corridor correspond to a situation where the robot got lost and required manual re-localization.

tation (for the colors of the robot’s plate) and blob detection. The centers of the blobs, which denoted the centers of the colored markers on the robot’s plate, were transformed from image coordinates to the world coordinates using the camera parameters under the assumption that these centers were at a constant height above the ground plane (the ground plane is denoted by  $Z = 0$  plane in our world coordinate system). The camera parameters (intrinsic and extrinsic) were obtained prior to the experiments. Using simple geometry on the center of the colored markers, the position and orientation GT of the robot was calculated. Recalling that we had 2 cameras present in our GT system, we used one camera for either sides of the environment in which the experiment was carried out. As the height of the markers, and therefore the robot, was known a priori, only one camera would be required to estimate the robot’s pose GT. Nevertheless, we assume that between the two cameras, placed on either sides of the environment, the one closer to the robot could produce more trustworthy GT of the robot’s pose.

### 4.3 Experimental results

We performed two main experiments, one in a controlled environment using the GT system, and another one consisting of a longer duration run (1 hour and 50 minutes, about 3 km traveled distance). Both took place at ISR, in separate floors of the North Tower (IST). In these experiments,  $V_{max} = 0.75m/s$ ,  $A_{max} = 0.6m/s^2$ , and  $T = 50m.s$ . The trajectories for each of these experiments are shown in Fig. 6.

In the accompanying video (available at <http://youtu.be/QW40yepKtuY>), we show all the trajectories of the first experiment, overlaying one of the GT camera’s image stream with the 2D re-projection of the world coordinate estimates (robot’s actual (GT), self-localization and FMM-based poses). This video shows several actual trajectories of the robot, in an environment where only some of the obstacles are on the map, namely the three yellow/blue poles (shown with dark dots in the robot’s map in Figs. 6(a) and 7). The previously-unknown obstacles for the robot (all objects on the field in Fig. 5 other than the poles) are shown in Figs. 6(a) and 7 as dashed regions. In the latter part of the run, randomly walking people, totaling 5 persons, were introduced



**Fig. 7:** Three autonomous navigation tasks, showing the path planned using FMM in square-dotted blue and the actual trajectory, according to self-localization, in dotted green, and according to the ground truth system, in continuous magenta. From left to right: a trajectory without unmapped obstacles (1:14), a trajectory including unmapped obstacles (0:50), and a trajectory with moving people (2:41). We include in parenthesis the approximate time (in minutes) each trajectory begins in the accompanying video.

in the scenario. In Fig. 7 we show three of those trajectories, with different obstacle types.

To assess whether the proposed approach is getting us closer to our goal of dependable autonomous navigation of mobile robots, we have been running long-term tests with the platform in the environment shown in Fig. 6(b). These tests consist in assigning random goal locations within the free space of the environment. At the time of writing, the platform has traveled a total of **18 km** during **11 hours** of movement. During this time, the platform lost localization a handful of times, mostly due to inaccuracies in odometry together with the range of the LRF being lower than half of the corridor length. However, we have no significant collisions to report. The only exceptions are light touching of persons' feet, explicable by the fact that typical feet lie entirely below the LRF scanning plane. That is, it only perceives the persons legs.

## 5 Conclusions and future work

This paper presented a novel design for dependable autonomous robot navigation, targeting domestic spaces, as well as populated public spaces. To this end, we employed an optimization based approach, combining a Fast Marching Method with a Dynamic Window Approach. The former provides optimal plan for the path, whereas the latter optimizes over a set of feasible robot actuation commands, providing guidance and obstacle avoidance. We evaluated this approach on a mobile robot platform, capable of high speed motion. Both short-term controlled experiments and long-term dependability tests were performed. The controlled experiments were evaluated using a ground truth tracking system to compare the actual trajectory with the one resulting from the Fast Marching Method.

As future work, we intend to continue long-term dependability tests, both in controlled environments as well as in populated indoor public spaces.

## References

1. Sethian, J.A.: Fast marching methods. *SIAM review* **41**(2) (1999) 199–235
2. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine* **4**(1) (1997) 23–33

3. Cakmak, M., Takayama, L.: Towards a comprehensive chore list for domestic robots. In: Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on. (March 2013) 93–94
4. Kruse, T., Pandey, A.K., Alami, R., Kirsch, A.: Human-aware robot navigation: A survey. *Robotics and Autonomous Systems* **61**(12) (2013) 1726 – 1743
5. Kruse, T., Kirsch, A., Sisbot, E., Alami, R.: Exploiting human cooperation in human-centered robot navigation. In: RO-MAN, 2010 IEEE. (Sept 2010) 192–197
6. Yuan, F., Twardon, L., Hanheide, M.: Dynamic path planning adopting human navigation strategies for a domestic mobile robot. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. (Oct 2010) 3275–3281
7. Stuckler, J., Behnke, S.: Following human guidance to cooperatively carry a large object. In: Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on. (Oct 2011) 218–223
8. Klaess, J., Stueckler, J., Behnke, S.: Efficient mobile robot navigation using 3d surfel grid maps. In: Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on. (May 2012) 1–4
9. Stueckler, J., Droschel, D., Grue, K., Holz, D., Kl, J., Schreiber, M., Steffens, R., Behnke, S.: Towards robust mobility, flexible object manipulation, and intuitive multimodal interaction for domestic service robots. In Rfer, T., Mayer, N., Savage, J., Saranl, U., eds.: RoboCup 2011: Robot Soccer World Cup XV. Volume 7416 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 51–62
10. Holz, D., Kraetzschmar, G., Rome, E.: Robust and computationally efficient navigation in domestic environments. In Baltes, J., Lagoudakis, M., Naruse, T., Ghidary, S., eds.: RoboCup 2009: Robot Soccer World Cup XIII. Volume 5949 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 104–115
11. Jacobs, S., Ferrein, A., Schiffer, S., Beck, D., Lakemeyer, G.: Robust collision avoidance in unknown domestic environments. In Baltes, J., Lagoudakis, M., Naruse, T., Ghidary, S., eds.: RoboCup 2009: Robot Soccer World Cup XIII. Volume 5949 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 116–127
12. Sethian, J.A.: *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge Press (1999)
13. Garrido, S., Moreno, L., Abderrahim, M., Blanco, D.: FM2: a real-time sensor-based feedback controller for mobile robots. *International Journal of Robotics & Automation* **24**(1) (2009)
14. Brock, O., Khatib, O.: High-speed navigation using the global dynamic window approach. In: Proceedings of the IEEE International Conference on Robotics and Automation. Volume 1., IEEE (1999) 341–346
15. Messias, J., Ventura, R., Lima, P., Sequeira, J., Alvito, P., Marques, C., Carri co, P.: A robotic platform for edutainment activities in a pediatric hospital. In: Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions. (2014) (accepted).
16. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics* **23**(1) (2007) 34–46
17. Thrun, S., Burgard, W., Fox, D.: *Probabilistic Robotics*. MIT Press (2005)
18. Ahmad, A., Xavier, J., Santos-Victor, J., Lima, P.: 3D to 2D bijection for spherical objects under equidistant fisheye projection. *Computer Vision and Image Understanding* **125**(0) (2014) 172 – 183