

Navigating Mobile Robot in Finite Environment Using Potential Field Merging

Adel Djellal^{1,2}, Dr. Rabah Lakel¹ and Randa Chergui¹

¹ Department of Electronics

Badji Mokhtar University-Annaba

² Department of Electronics and Telecommunication

Kasdi Merbah University-Ouargla

rabah.lakel@univ-annaba.dz¹, djellal.adel@univ-ouargla.dz²

Abstract

Navigation can be defined as moving a robot to a defined objective and avoiding undesirable situations (collision and unsafe conditions) simultaneously. In this work, a navigation technique based on Potential Field has been used to plan an optimal-safe path to pilot a mobile robot in known environment.

The proposed techniques are based on extracting path by merging information from two potential maps: security potential, pulling back robot from obstacles, and slope potential attracting robot to target. Three planning techniques are proposed; weighting technique that gives equal importance to safety and nearest path yet changes importance to highlight attraction in conflict situations. Degraded technique, which is more favorable for cases where security is not often important, so superior weight is given to nearest path. The last technique is to integrate paths from the two previous techniques to have safe paths in some situations and have nearest path in other situations.

Comparison between proposed techniques with addition to previous work (graph-based technique) has been done to discuss difference and possible optimizations. The proposed techniques have shown interesting results especially for complex environments with curved obstacles, which is not resolvable with graph based technique. Tests and implementation has been done using Matlab simulations and a real mobile robot with a bird's-eye view camera to discuss techniques and propose future work.

Keywords: Robotics, Navigation, Path planning, Obstacle avoidance

1. Introduction

For several decades, mobile robotics has encountered a spectacular progress. In fact, the robot generations are quickly succeeded giving each time more and more performances in matter of perception, autonomy and decision making. This progress was an inevitable result of developments made in mechanics, microelectronics and computer science. Research in robotics has a goal to design and make machines capable of evolving and react with the environment such that accomplishing different tasks for which they are created.

Two cases of navigation can be cited: online and offline. The first is based on the idea of selecting a path according to the present knowledge, then move in this path to have more information. Based on the new findings, the path is then replanned [7]. Some works are based on this case, and via representing the environment with graphs [15], to solve the pursuit evasion problem [3, 4, 11]. The second case is based on proposal of

having to know the entire building and generate a global scheme where each starting point will take the robot to the goal. Some works [11] use the graph theory to generate an area graph of the building to facilitate the navigation for the robot in such way to find the evader. This work is based on the second case using potential field technique to find the optimal path.

A mobile robot operating in a known or unknown environment is often confronted with several problems such as:

- **The path planning:** The main objective of a mobile robot is to make a movement by connecting a source point to a destination port. The execution of this task is performed by a technique based on the generation of a map of potential shadow. From point destination, brought to zero potential, increasing diffusion potential as wavefront will scan all available navigation cells. Moving the robot from the source point to the destination will be based on descent potential without local minimum potential
- **The obstacle avoidance:** Detection and obstacle avoidance behavior as a base on almost all mobile robots. It aims to allow the robot to navigate his environment safely. For the environment considered in our application (labyrinth) such that constraint can be translated by the movement of the robot in the middle of corridors (safest distance to walls). Especially when the walls are a real danger such as high voltage lines. For this purpose a second potential map is established where all obstacles (walls) carried at potential (-1) diffusing a growing wave potential and modeling the environment as ridgeline simultaneously.

This paper is organized as follows:

Firstly, the planning techniques are presented to have general idea on possible techniques for path planning. Secondly, the used method on this paper is explained where potential field propagation is described, chain coding and path planning are presented, and finished by highlighting the proposed planning techniques. Moreover, the implementation is done in two parts, simulation to compare proposed techniques with previous work for different environments, and experimental work on a real mobile robot to validate proposed techniques on real world. Finally, a conclusion completes this paper where results are discoursed while proposing future work.

2. Motion Planning Techniques

The presented techniques below can be used for holonomic and Nonholonomic robots [13, 17]. In our work, it has been used for a differential mobile robot which is a case of holonomic mobile robots.

2.1. Trapezoidal Decomposition

This technique is based on the decomposition of space using a discrete presentation of the robot environment. An exact decomposition leads to a set of polygons dividing the free space. With this representation, a graph is generated linking different adjacent polygons of the environment [12, 14].

2.2. Potential Fields

This method has been first introduced by O. Khatib in 1986 [10] for manipulators and mobile robots [9]. It is very different from other techniques because it is not based on a geometric reasoning. Furthermore, it is not complete which means that it is possible that a

solution exists and the system does not find when it is stuck in a local optimum [1]. The proposed technique in this work is an extension for the Potential Field Technique.

2.3. Voronoï Diagram

The Voronoï Diagram [12] gives, for a plane polygonal environment, the lines of equal distance from the obstacles. These lines allow the robot to navigate hypothetically safely. i.e. as far as possible from the obstacles.

2.4. Visibility Graph

Parsons firstly used this technique in 1976 [15] to solve the pursuit-evasion problem. Recent works [2, 11, 3, 4] used visibility graph to solve pursuit evasion problem in known and unknown environments using Depth and Breadth First Search (DFS and BFS respectively), and in Shakey [18], the first mobile robot.

The technique is based on the idea of connecting the starting point with all summits of polygonal obstacles that are visible from the robot, and do the same thing with the visible summits, etc. At the end when all points are connected, the path for start to end can be found.

3. Used Method

The used approach is interested to create a 2D grid of the environment. It is a global technique that allows obtaining a path plan based on Potential Field (section 2.2) technique and with no risk to the robot to be stuck in local optimum.

3.1. Potential Propagation Algorithm

Initially, a negative potential is given to the segments that are not accessible (obstacles), and a positive potential (very high) to the free space (Figure 1)

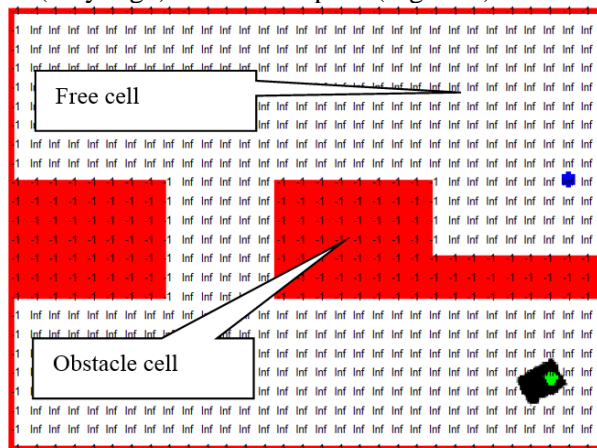


Figure 1. Potential Allocation

3.1.1. Security Potential Propagation: The propagation technique works as follows:

- Starting from obstacle segments (in this example, with potential equals "-1"), the potential of all neighbor cells (with high potential, in this example "inf") are given potential incremented by 1.
- Then, the cells that are modified (with potential up by 1), will spread their potential to the neighbor cells.
- Spreading the potential is repeated until there is no free space with infinite potential (Figure 2).

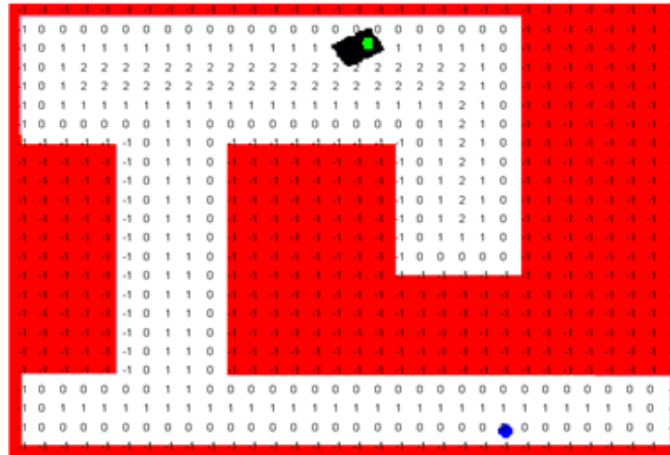


Figure 2. Potential Spreading

3.1.2. Front Wave Propagation

The basic idea of this algorithm is to find the fastest path to the target. This is achieved by assigning a negative potential border cells (not available for navigation), a large positive potential for available navigation cells and a potential equal to zero at the target cell.

The operation can be summarized in the following steps:

- 1) Initialize $W_0 = I$ (target cell) for $i = 0$
- 2) Initialize $W_{i+1} = \Phi$
- 3) for each cell in W_i , look for the four neighboring cells, increment potential value 1 and insert in W_{i+1} .
- 4) if $W_{i+1} = \Phi$ then finish else $i = i + 1$ and go to step 2.

At the end, any point of the environment has the potential shown in equation 1, with i is the Euclidian distance between the current cell and the target cell.

$$Pd(cell) = i \tag{1}$$

The values shown in Figure 3 are the distances from the target (blue circle).

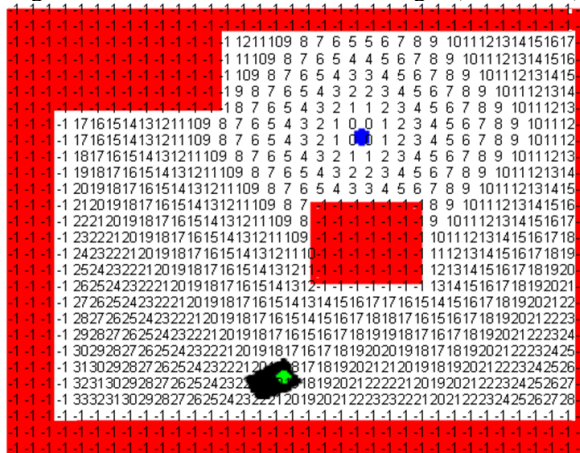


Figure 3. Descent Potential Allocation

3.2. Chain Coding

The eight Freeman directions [8] is an alphabet consisting of eight letters (0; 1; 2; 3; 4; 5; 6; 7) that allows to write a sentence corresponding to the path from the robot cell to the destination cell. Figure 4 shows path coding using Freeman 8 directions.

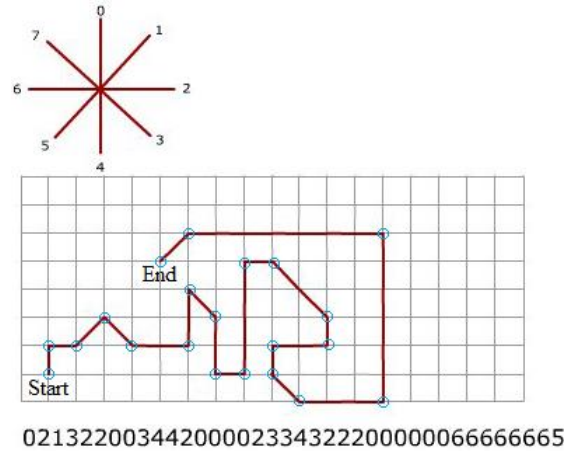


Figure 4. Freeman Coding with 8 Directions [6]

Chain coding was used in experimental work (Section 4.2) to simplify control algorithm (Algorithm 1) such that each chain of repeated letter is replaced by one letter and the occurrence of recurring character. Consequently, the word will be containing only points where the robot changes its direction. For example, in Figure 4, the simplified word would be [021320342023432065] and occurrence vector is [11112212141211361]. Important points that give directions are shown in Figure 4 with blue circles.

3.3. Path Smoothing

When the path contains direction changing over short distance, e.g. [06060] according to the eight Freeman directions, and to respect the robot dynamics, the path is smoothed as follows:

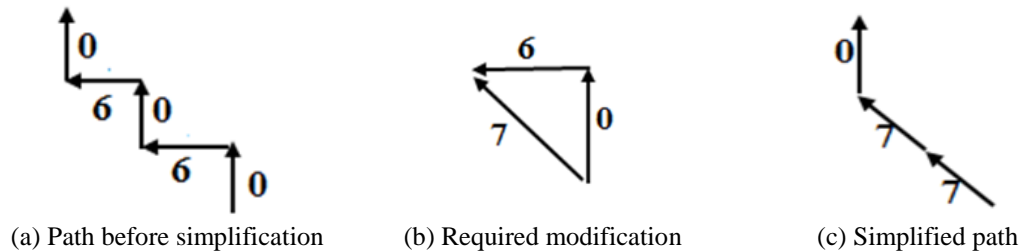


Figure 5. Smoothing Path Example

Path smoothing is not detailed in this work but will be developed in further work by using Genetic Algorithms and other techniques to optimize path subject to robot dynamics.

3.4. Mobile Robot Model

3.4.1. Definition

The mobile robot is a mobile robot actuated by two independent wheels and optionally having a number of free wheels ensuring its stability. The model of Differential wheeled robot type is given in Figure 6. The loose wheels are omitted since they do not occur in the drive. This type of robot is very popular because of its simplicity of construction and interesting kinematic properties.

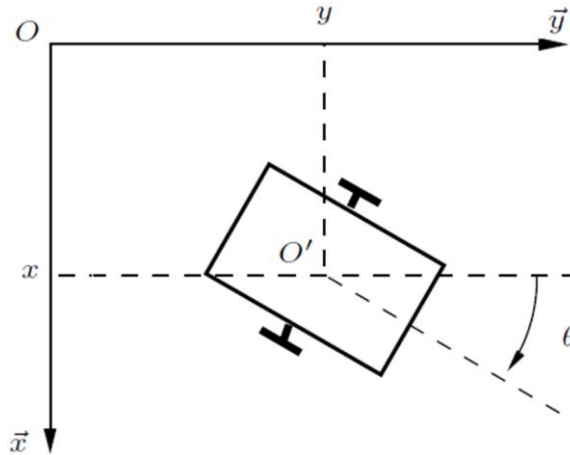


Figure 6: Unicycle-type mobile robot

3.4.2. Robot Modeling

Since the robot have two independent wheels that are controlled by a constant speed, the model used in the work neglects friction and is based on the following motion equations:

- 1) Translation (*forward=backward*)

$$\begin{cases} x_1 = x_0 \pm del \times deltaRun \times \cos(\theta_0) \\ y_1 = y_0 \mp del \times deltaRun \times \sin(\theta_0) \\ \theta_1 = \theta_0 \end{cases} \quad (2)$$

- 2) Rotation (*right=left*)

$$\begin{cases} x_1 = x_0 \\ y_1 = y_0 \\ \theta_1 = \theta_0 \mp del \times deltaRot \end{cases} \quad (3)$$

where:

- $[x_0; y_0; \theta_0]$: Current robot state.
- $[x_1; y_1; \theta_1]$: Next robot state.
- *del*: Movement delay.
- *deltaRun*: Translation constant.
- *deltaRot*: Rotation constant.

3.5. Path Planning

Because environments are labyrinths consisting of corridors and crossways, the evolution of the robot is according to one of eight directions shown on the figure below, called Freeman directions [8] that correspond to neighbors of the cell occupied by the robot. The trajectory of the robot is composed of successive movements according to Freeman directions.

From the corresponding cell at the current position of the robot, two 3x3 matrices are established. The first matrix includes the information about the Slope potential (*Pd*) of the eight neighboring cells of the cell occupied by the robot, while the second matrix includes the information about the safety potential (*Ps*). Consequently, each cell will have two potential values of its distance to the target cell $Pd(cell)$, and the its distance from obstacles $Ps (cell)$.

The elements of the matrix that contains the motion candidate directions are obtained by applying the following formula:

$$P(n, m) = Ps_1 \times ((Pd_0 - Pd_1) + (Ps_1 - Ps_0)) \quad (4)$$

Where Pd_0 and Ps_0 represent respectively the two potentials (Descent and safety) of the current position of the robot and Pd_1 Ps_1 represent respectively the two potentials (Descent and safety) of the cell (n, m) in the region of the current cell.

For each position the direction is decided using equation 4 (table 1).

The robot navigation can be divided into five phases:

- **Phase 1:** Search the crest with greater value.
- **Phase 2:** Evolution by lowering the potential on a line maximum value.
- **Phase 3:** Lane change with research line maximum peak.
- **Phase 4:** Tabbing between phases 2 and 3.
- **Phase 5:** Gradient descent towards the target and stop.

The graph (Figure 7) shows the evolution of the robot between the four phases:

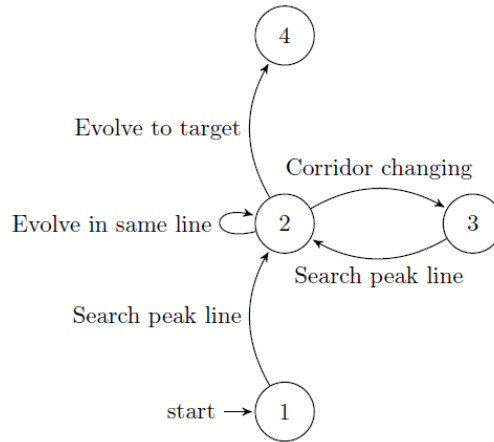


Figure 7. Navigation Graph

Matrix Ps $Ps(i, j) \geq P(n, m)$	Matrix Pd $Pd(i, j) \leq P(n, m)$	Candidates $P(n, m) = \text{equation 4}$	$\max(P(n, m))$
$\begin{matrix} 3 & 3 & 3 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{matrix}$	$\begin{matrix} 35 & 36 & 37 \\ 36 & 37 & 38 \\ 37 & 38 & 39 \end{matrix}$	$\begin{matrix} 9 & 6 & 3 \\ 2 & 0 & -2 \\ -1 & -2 & -3 \end{matrix}$	$\begin{matrix} 9 & 6 & 3 \\ 2 & 0 & -2 \\ -1 & -2 & -3 \end{matrix}$

Table 1. Decision Made Using Equation 4

3.6. Planning Techniques

3.6.1. Weighting Technique: Generally, the Robot navigates according to computed neighbors presented in section 3.5. However, some situations, where there is more than one Candidate in Table 1, the method of assigning weighting coefficients Ad and As respectively and lowering security potential is adapted. These two coefficients are chosen ways to respect the following constraint:

$$Ad + As = 1 \text{ and } As < Ad \quad (5)$$

This encourages the potential descent by applying the formula:

$$P(n, m) = Ps_1 \times (Ad \times (Pd_0 - Pd_1) + As \times (Ps_1 - Ps_0)) \quad (6)$$

Where Ad and As are the weights assigned to potential security and descent respectively

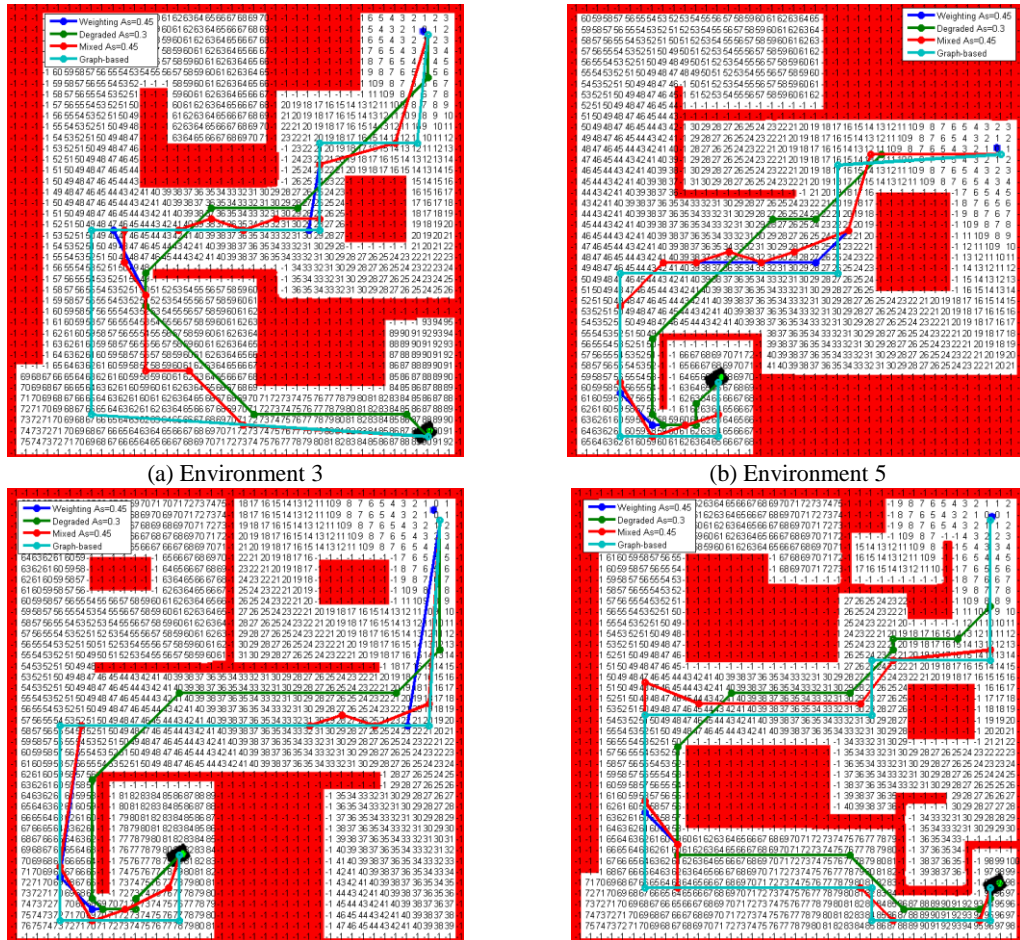
3.6.2. Degraded Mode: In the case where the required level of security is not very high, the weighting formula (equation 6) to determine the robot decision direction is applied. The absence of conflict in the choice of the direction to take will be paid by the evolution in the corridors on ridgelines value less secure.

3.6.3. Mixed Technique: With the real robot whose navigation is managed by technique that gives priority to the navigation in the corridors on peak lines with maximum value, it is associated a virtual robot evolving according to the degraded mode. If the difference between the two paths is constant, each robot operates on its own way. However, if the difference between the two paths begins to increase, the trajectory of the real robot will be controlled by degraded mode to maintain a relatively constant distance. The fact that the evolution in degraded mode can negotiate lane changes flexibly, real robot operates in lines with the maximum safety and takes the path of the virtual robot to detect early the end of a corridor and make the change in the best conditions.

4. Implementation

4.1. Simulation

Figure 8 shows some simulation examples to compare between the proposed techniques and a previous work on graph-based safe path [5].



(c) Environment 6

(d) Environment 7

Figure 8. Simulation Results for the Two Techniques

A comparison of the two proposed techniques was made on several environments and the results are shown in the following two tables and bar graphs: the first (Table 2 and Figure 9) shows the accumulated P_{total} and the second (Table 3 and Figure 10) comparing the number of iterations.

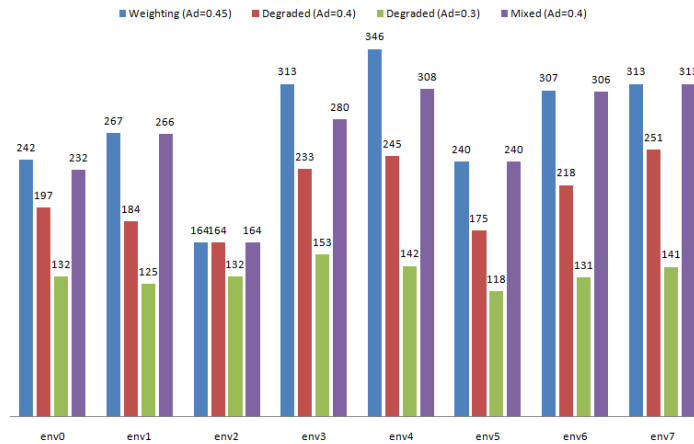


Figure 9. Accumulated P_{total} for Different Environments

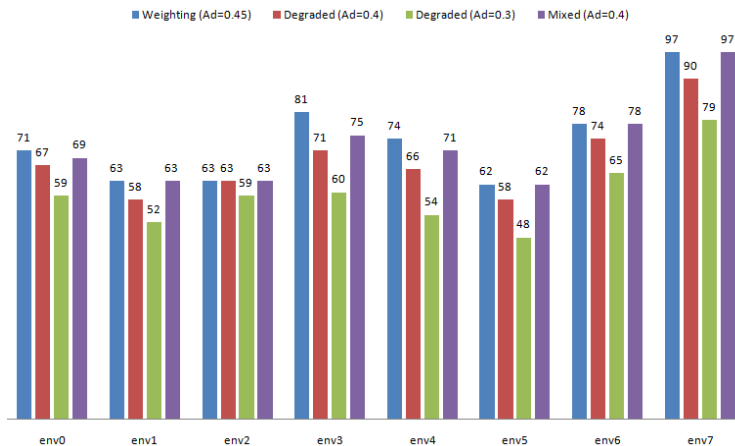


Figure 10. Number of Iterations for Different Environments

Table 2. Accumulated Value of P_s (P_{total}) for Different Environments

	Env0	Env1	Env2	Env3	Env4	Env5	Env6	Env7
Weighting (Ad = 0.45)	242	267	164	313	346	240	307	313
Degraded (Ad = 0.4)	197	184	164	233	245	175	218	251
Degraded (Ad = 0.3)	132	125	132	153	142	118	131	141
Mixed (Ad = 0.4)	232	266	164	280	308	240	306	313

Table 3. Number of Iterations for Different Environments

	Env0	Env1	Env2	Env3	Env4	Env5	Env6	Env7
Weighting (Ad = 0.45)	71	63	63	81	74	62	78	97

Degraded (Ad = 0.4)	67	58	63	71	66	58	74	90
Degraded (Ad = 0.3)	59	52	59	60	54	48	65	79
Mixed (Ad = 0.4)	69	63	63	75	71	62	78	97

4.2. Experimental Results

The used robot in the experiments was a POB-Bot [16], by POB Technology, The perception of the robot position has been done using a VGA Webcam (Figure 11). It can be seen that the robot is a dark circle with one bright circle at the front (figure 12); these are used for image processing to detect the robot position (big dark circle) and its direction (small bright circle). This new technique showed better results compared to previous works [5] where robot detection lacked of computation time and color perturbations. Algorithm 1 shows the robot control via computer, it is based on the idea that to go from a node to the next node they are visible from one to another because the environment is based on corridors.

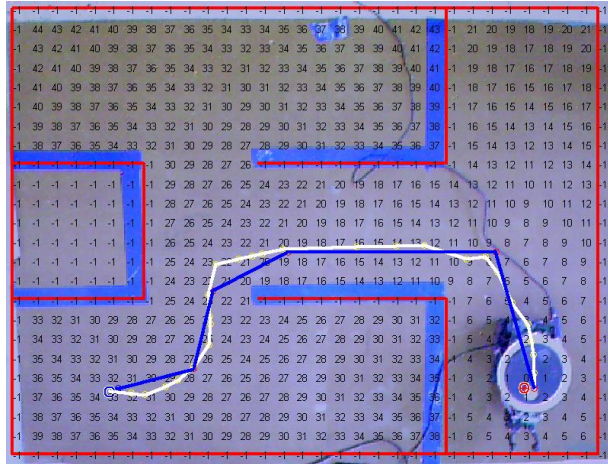


Figure 11. Robot Path for Experimental Environment 6

Algorithm 1: Robot navigation algorithm

- For each current node n_i
 - While distance from robot to $n_i > d_{max}$ do:
 - Take a camera snapshot
 - Detect robot position and direction
 - While robot is not facing n_i do:
 - Turn robot to face n_i
 - Take a camera snapshot
 - Detect robot position and direction
 - Move the robot
 - Stop the robot
-

Figure 11 shows a path planning example where white line is the robot path and red line is optimal path. It is visible that the robot did not follow the optimal way; this is due to image processing incertitude and communication delays.



Figure 12. Mobile Robot View from Above

5. Discussion

From Figures 9 and 10, it is noticed that the nature of the environment (*i.e.* its complexity: number of crossings, wide hallways, etc.) greatly affects the accumulated safety potential regardless of the used technique. However, weighting technique is very safe (bigger accumulated safety potential) yet very expensive in term of iterations (big number of iteration). In the contrary, degraded technique is faster to achieve goal yet does not get so far from obstacles (as A_s getting smaller, as path is nearer to obstacle). Though, mixed technique does not show big difference from first technique while real robot generally adapts weighting technique.

It can be concluded that weighting technique is very adapted to safe paths where it very important to robot to navigate as far as possible from walls (high voltage, sensitive components, etc.). Furthermore, degraded technique prioritizes cases where near path is more important than dangerous walls. However, Mixed technique needs more work to optimize decision between two decision techniques.

6. Conclusion

This paper deals with the path planning of an autonomous mobile robot, and its implementation by introducing the vision in feedback loop. The path planning is performed by environment mapping techniques, based on the Potential Fields technique that was firstly proposed by [10]. The major problem known in the original technique is its weakness in local optimum, so the robot can be stuck in regions where retracting force (obstacle) and attracting force (Target) are equal.

However, the proposed technique allows obtaining an optimal path from the starting point to the target point without getting trapped while selecting paths and ensuring maximum safety. The image processing was not well discussed in this work. However, the robot detection was made using a circle detection technique which showed better results compared to prior techniques [5].

The proposed techniques were compared with previous work on graph-based safe path [5] to be able to see good and bad points in the current work. Presented techniques have shown interesting results for specific use (section 5). However, this work can be extended to study the movement of a robot in dynamic (presence of moving obstacles) or partially known environments, integrating a data fusion system combining global information provided by the camera, and the local information provided by robot sensors and also the trajectory optimization techniques consistent with the robot kinematics and dynamics.

References

- [1] B. Bayle, "Robotique mobile", Ecole Nationale Supérieure de Physique de Strasbourg (2011).
- [2] A. Djellal, "Conception of multiagent system applied on mobile robots", Master's thesis, Badji Mokhtar University - Annaba (2009).
- [3] A. Djellal and R. Lake, "Using graph search technique to solve the pursuit-evasion problem in unknown environment", In International Conference on Industrial Engineering and Manufacturing ICIEM'10 (2010).
- [4] A. Djellal and R. Lakel, "Using graph theory to search an evader in unknown environments", In Conférence Internationale sur le Traitement de l'Information Multimédia (CITIM'2012) (2012).

- [5] A. Djellal, R. Lakel, B. Lakehal, and I. Belazzoug, "Controlled robot with RF system in finite known environment", In International Congress on Telecommunication and Application'14 (ICTA'14), Bejaia, Algeria (2014).
- [6] C. Dolci, D. Salvini, M. Schrattnner, and R. Weibel, "Structures for data compression", Geographic Information Technology Training Alliance (GITTA) (2010).
- [7] T. Ersson and X. Hu, "Path planning and navigation of mobile robots in unknown environments", Intelligent Robots and Systems, vol. 2, (2001), pp. 858-864.
- [8] H. Freeman, "On the encoding of arbitrary geometric configurations", IRE Transactions on Electronic Computers, (1961), pp. 260-268.
- [9] K. Houda, R. Lakel, and D. Messaoud, "Méthode globale de planification de trajectoire pour un robot mobile autonome basée sur les champs de potentiels fictifs", In Conférence Internationale sur la Productique (2005).
- [10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", International Journal of Robotic Research, vol. 5, (1986), pp. 90-98.
- [11] R. Lakel and A. Djellal, "Resolving the pursuit evasion problem in known environment using graph theory", Int. J. Bio-Inspired Computation, vol. 2, no. 6, (2010), pp. 434-439.
- [12] J.-C. Latombe, "Motion Planning", University of Stanford (2004).
- [13] J.-P. Laumond, "Nonholonomic Motion Planning for Mobile Robots", LAAS, CNRS, Toulouse (1998).
- [14] S. M. LaValle, "Planning Algorithms", Cambridge University Press (2006).
- [15] T. D. Parsons, "Pursuit-evasion in a graph", Theory and Applications of Graphs, (1976), pp. 426-441.
- [16] POB-Technology, POB-Bot User's Manual, POB-Technology (2008).
- [17] M. I. Ribeiro and P. Lima, "Motion planning", Instituto de Sistemas e Robotica (ISR) (2002).
- [18] C. A. Rosen, Nils J. Nilsson, and Milton B. Adams, "A research and development program in applications of intelligent automata to reconnaissance-phase I", Technical report, Stanford Research Institute, (1965).

Authors



Adel Djellal, Assistant Teacher at University Kasdi Merbah – Ouargla, Department of Electronics and Telecommunication. He received his Engineering degree on Automatics from Badji Mokhtar University – Annaba in June 2006 and his Magister degree on Automatics from the same university in June 2009. He is now preparing the Doctorate degree in the same university.



Rabah Lakel, Lecturer in Badji Mokhtar University, Electronics Department, Annaba. He is a member of Automation and Advanced Signals Laboratory (LASA) in Badji Mokhtar University.

Randa Chergui, Master degree student from Badji Mokhtar University - Annaba.