Indian Journal of Science and Technology, *Vol 10(14), DOI: 10.17485/ijst/2017/v10i14/112175, April 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Global Path Planning for Mobile Robots using Image Skeletonization

Fernando Martinez Santa*, Edwar Jacinto Gomez and Holman Montiel Ariza

Universidad Distrital Francisco Jose de Caldas, Bogota D.C; fmartinezs@udistrital.edu.co.in

Abstract

Objectives: To propose an algorithm for global optimized path planning for mobile robots on static and observable environments, based on image skeleton technique and using some other image processing algorithms. **Methods/Analysis:** The proposed geometric scheme is supported on the skeletonization of image of free space into a navigation environment, as a strategy to limit the number of possible paths from a starting to end points for a mobile robot. This approach, considerably simplifies the selection of appropriate paths, allows defining lines (edges) that can be utilized by graphs for the specification of paths. Several simulation and real test were done to prove the efficiency of the algorithm. For the real test, a prototype differential platform was build using a LEGO Mindstorms NXT robotics kit. **Findings:** The strategy demonstrated excellent performance and robustness over about 50 simulations and experimental tests, finding always a feasible and short path., but taking in average 6.4 seconds to find the solution. **Novelty/Improvements:** The proposed algorithm showed to work better than other tested algorithms like Voronoi and Bug, especially when the navigation environment is a maze.

Keywords: Deep Search, Hybrid Scheme, Image Skeletonization, Navigation, Path Planning, Robotics

1. Introduction

One of the most interesting issues in robotics in the last years, both by its complexity, as its complexity as its practical applications, has been without a doubt, path planning for mobile robots.[1-3] This area has strong implications as in autonomous process development for industrial applications, as in commercial prototypes design able to interact in real environments.

The strategies that have been used to analyze and try to solve the issue can be classified in two types. First category is centered in globally observable environments, it means, in environments for which a reading of the environment's architecture is done previously, which is joined to the robot data, in order to design a navigation strategy.[4,5] The second one is centered in the local sensor readings of the robot, that in general, induces reactive navigation strategies modeled as hybrid systems.[6,7] Each of these strategy groups allows to apply different approaches for solving the problem of planning the path. In the first case, when it is possible to sense the entire environment, to design geometry-based strategies for defining the path is possible. In the second case, when the sensing is local, the strategy has to react continuously to the local conditions. Even though most of these strategies are formulated for only one robot, most of them allow to self-organize groups, which integrates some degree of distributed processing.[8] Within the set of geometric strategies, there are classic schemes that are widely known, such as Voronoi Diagrams, artificial potential fields and the regions segmentation.[9-11] In general, each of these strategies have as advantage their simplicity and implementation ease, but entail serious problems such as: convergence, robustness, optimum path and definition of feasible paths for the robot. Most of them start from the concept of a central control unit, in charge of the environment sensing and the direct handling of the robot. Also, in order to watch the real robot dimensions inside the algorithms, they start

**Author for correspondence*

with an image dilation process applied to the obstacles, taking into account a safe distance defined from the robot dimensions, which guarantees the not existence of collisions. In this way, the trajectory computing is easily done, considering the robot as a dimensionless point.

The research shown in this paper proposes a geometric algorithm that pretends to maintain the features such as low computing cost and structure simplicity, and at the same time, remove the convergence, robustness and optimum path problems. The algorithm is a hybrid scheme supported in the skeletonization process of environment image. The skeletonization algorithm simplify the environment information, specially the related one with the free-obstacle space configuration, defining on it a slim graph, where its points are at the maximum possible distance of near obstacles and borders. Thus, the searching space is reduced to a finite set where the optimum sub/optimum path is selected.

This research was done between different groups of robotics and control theory of Universidad Distrital (Bogotá, Colombia) as part of the research in robotics at Technological Faculty, in order to reformulate the control area in collaboration with the School of Computing and Information Sciences de la Florida International University (USA).

2. Problem Formulation and Assumptions

A robot (nota point, with a defined area in R^2 $\mathbf{R^2}$) is placed in a point p_i inside a navigation environment $W \subset R^2$ flat, connected and compact. Within this navigation environment, $\mathcal{O}$ is defined as the obstacle set, in which, each obstacle $O \in \mathcal{O}$ is closed with a limit(border)connected, analytic by segments and with finite length. Also, the obstacles in $\mathcal{O}$ are disjunct pairs (mutually disjunct) one another (not share common points) and finite, it is shown in Figure 1. The limit or border of W is denoted by ∂W, and the limits of each obstacle are considered part of ∂W. The free space in which the robot can navigate is denoted by E, which is contained in R^2, and is defined as $W - \mathcal{O}$. In accordance with the configuration of W and $\mathcal{O}$, if at least one path in E exists, which allows the robot to move from the initial point p_i to a destination point p_t, then the path planning strategy must define it. Additionally, in the case of existing multiple paths, the strategy has to define an optimum path (one of the shortest) and safe for the robot movement.

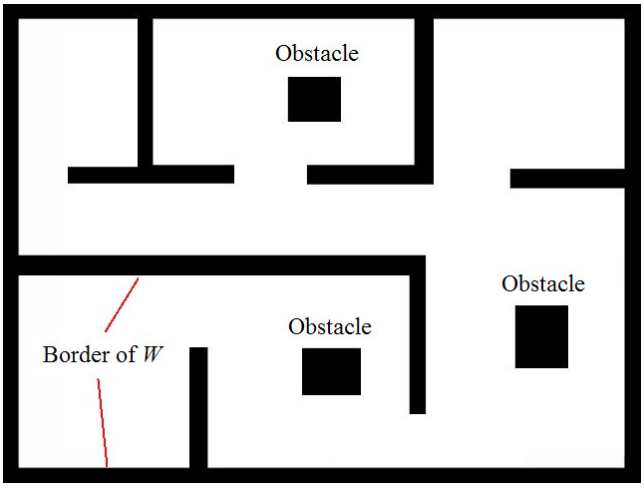

Figure 1. Navigation environment with formulation details. Obstacles (inner black regions) inside the navigation environment W.

Like in a lot of schemes, the environment is divided in a finite set of connected regions, over which, a discrete transition system is defined Figure 2.[12,13] These regions are bounded by the environment limits ∂W and the resultant edges of the skeletonization process applied to the free space E. These edges establish the possible navigation paths, and are defined as equidistance points of ∂W.

The regions that decompose W, are generated or induced by the obstacles in W. Each edge of skeleton is denoted by Y, and its extremes are common with ∂W or are part of an edge or node of the skeleton. The set of all edges of the skeleton is denoted by Γ. Each $\gamma_i \in \Gamma$ is the image of an injective curve and rectifiable defined between limits of environment ∂W and edges of skeleton in E. The regions are denoted by r, and the set of all regions as R.

The robot is considered small regarding W and the regions $r \in R$. Also, the size of robot is considered limited by a circle of diameter d in R^2. This robot must move from p_i to a destination point p_t navigating through E, it means, avoiding the obstacles. Additionally, it is desired that the selected path is as short as possible, and not has small turn angles that make difficult the handing (safe for the robot). The navigation algorithm must plan the navigation path so that, the required edges γ_i are selected,

in order to perform a safe and efficient path between the points p_i and p_t, if and when the geometry allows the pass of the robot. The environment will be completely observable by the control central unit, which communicates with the robot to transmit the navigation path.

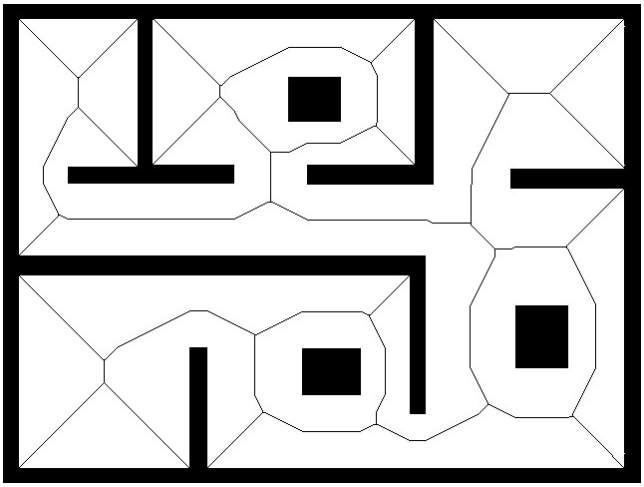

Figure 2. Generated image skeletonfor the free space E .

3. Methodology and Design

In the proposed algorithm, the path is designed starting from digital image processing. The image corresponds to the sensing of *W*, information captured using a digital camera. By means of applying specific filters, a segmentation of the navigation environment *W* is done into regions $r \in$ R.This segmentation use the skeletonization of the obstacle-free space E , process that looks for representing E with a graph where its edges are composed by points, which maintain the maximum distance to ∂W . A post-analysis of these edges allows defining the navigation path Figure1 and 2.

The image is initially subjected to a basic process of obstacles and free space E identification. Each image is turned into gray scale and after into binary (Black & White), using adequate thresholds to the lightness level. Once the obstacles are clearly identified from the background, a dilation or obstacle growing operation is applied in order to identify the safe area for the robot navigation. This dilation process increases the obstacles size for the processing in a factor of $\frac{d}{2}$, this guarantees that the robot will not collide with the obstacles in spite of the path has been computed for only one point.

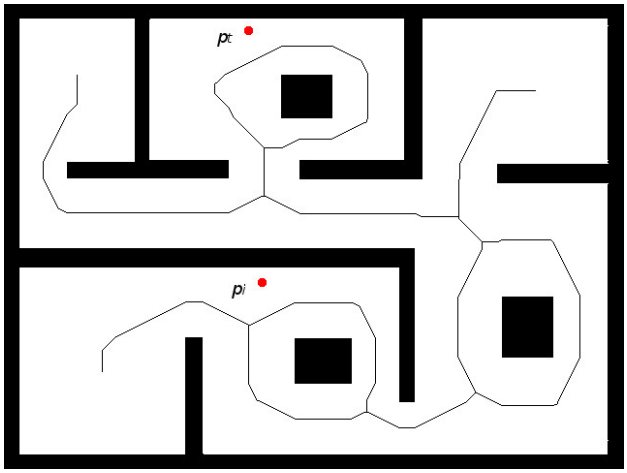

Figure 3. Navigation environment with skeleton plus initial pointp$_i$and final point p$_t$. the edges γ that had common points with ∂W have been removed.

Skeletonization algorithm can be seen as a morphologic operation of controlled and iterative erosion. A repetitive erosion over the image is done, until the object gets slim and separated regions by lines are formed (edges, $\gamma_i \in \Gamma$). That erosion is done, using a structuring element or kernel matrix, which is square and smaller than W .The skeletonization process produces navigable and safe edges (equidistant to the obstacles, and without possibility of collision due to dilation). Subsets of $\gamma_i \in \Gamma$ can be part of navigable and safe paths along W . Although, some edges $\gamma_i \in \Gamma$ take the robot to collide with ∂W .Also, two edges are missing, those that connect the graph to the point's $p_i$andp_t Figure 2. Before starting the selection process of the path, it is necessary to eliminate from the graph the edges those produce collisions, those that have common points with ∂W .The applied filter for that, looks for the graph points that intercept with ∂W (extreme points of graph) and go forward erasing the edge until reach a vertex Figure 3.

Para detecting both the graph pixels that intercept ∂W and the pixels of a vertex, the eight neighbor pixels are checked for each graph pixel. The set of eight neighbors is defined for a pixel $p(x, y)$, as shown in the Equation 1.The amount of active neighbor pixels is defined as shown in Equation 2.

$$N_8(p(x,y)) = \begin{Bmatrix} p(x-1,y+1), p(x,y+1), p(x+1,y+1), \\ p(x-1,y), \qquad\qquad p(x+1,y), \\ p(x-1,y-1), p(x,y-1), p(x+1,y-1) \end{Bmatrix} \quad (1)$$

$$\forall\, p(x,y) \in I:\ \ p(x,y) = 1 \ \rightarrow\ k = \sum_{j=1}^{8} N_8\big(p(x,y)\big)_j \qquad (2)$$

The pixels with $k = 2$ are edge points, the ones that have $k = 1$ are points that intercept to ∂W , and the ones that have $k = 3$ or more are vertexes. To join the initial point p_i and the final point p_t with edges to the graph, the Euclidean distance is computed between these points to each point of the edges of graph. This allows finding the shortest distance, line that is turned into a new edge of graph Figure 4.

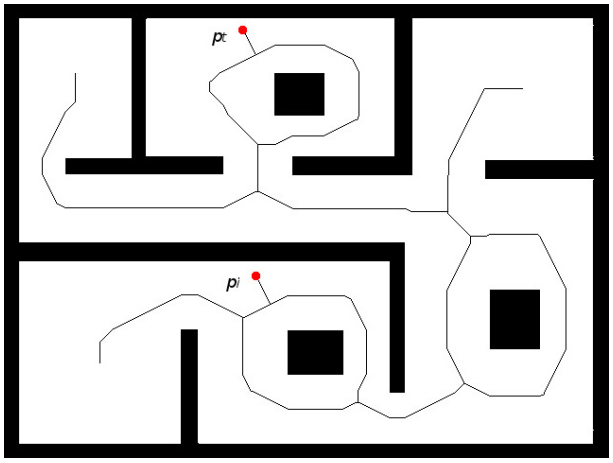

Figure 4. Skeleton image plus two edges that connect the point's p_i and p_t.

The next step in the process is the selection of the optimum/sub-optimum path. The ideal criterion is to find the shortest path with the largest turn angles. Due to the finite and small number of possible combinations of edges γ able to be part of the path between p_i and p_t, first, a deep search algorithm is implemented. This algorithm is not applied, in order to find the solution path, but finding all of the existent paths.

The algorithm starts from p_i, and go through the graph edges. When it finds a vertex, it explores one of the possibilities. If it successfully arrives to p_t, stores the path and starts again, changing to the next possible path in the last vertex. If it does not successfully arrive to p_t, also comes back to the last explored vertex and verifies another possibility, but without storing the path. The algorithm is iteratively executed until to explore all the possibilities of the graph, assigning to each edge of the path a weight proportional with its length. After, between the stored success paths, the path with the least summation of edges weight (shortest path) is selected Figure 5.

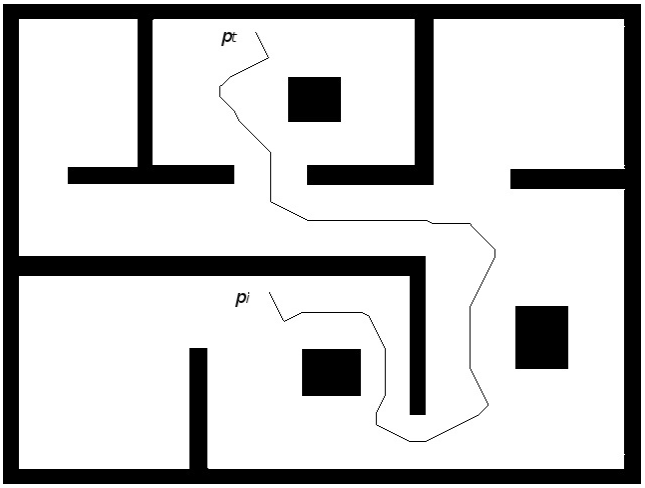

Figure 5. Shortest path identification. Between all of possible edge combinations, the one with lowest weight are selected.

This strategy guarantees the shortest path existent in the graph, but not the safest one (without small turn angles). In order to reduce the complexity of the path and make easy the robot navigation, a reduction of points of $\gamma_i \subset \Gamma$ (the ones that compose the total path) is proposed. The process is done by means of a decimation of coordinate's vector, defining a limited amount of points. The final point's number is defined as the 5%of the length of the largest edge of the graph Figure 6.The decimation algorithm uses the arithmetic measure of the coordinates that is why the output points do not exactly belong to the path defined by the graph edges. It takes into account the displacement inertia, which makes soft the navigation path and makes that the robot does not exactly seek the middle point between the obstacles.

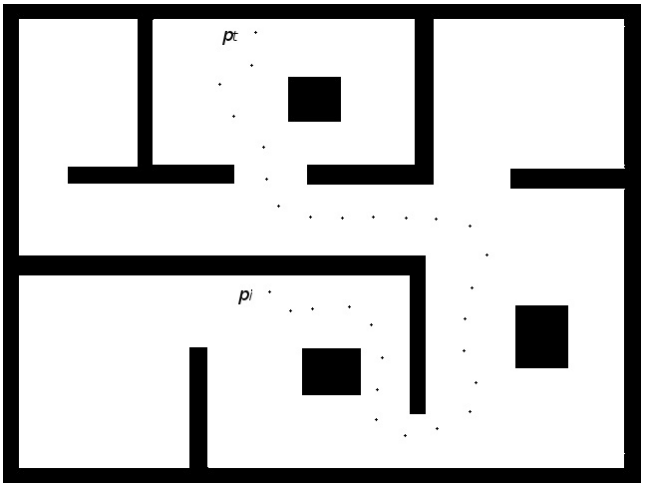

Figure 6. Resultant points after doing a decimation process over the obtained path.

4. Results

The navigation strategy was evaluated using a total of 30 sceneries, 20 of them caught in the laboratory with real prototypes, and 10 generated by software. Most of the environments had geometric obstacles, which were defined as target of the initial profile. However, the performance tests considered also mazes. The experimental prototype used a differential platform with only one free-turn wheel, built from a LEGO Mindstorms NXT robotics kit. Figure 7 and 8 show two navigation cases solved for different initial and final positions of the robot.

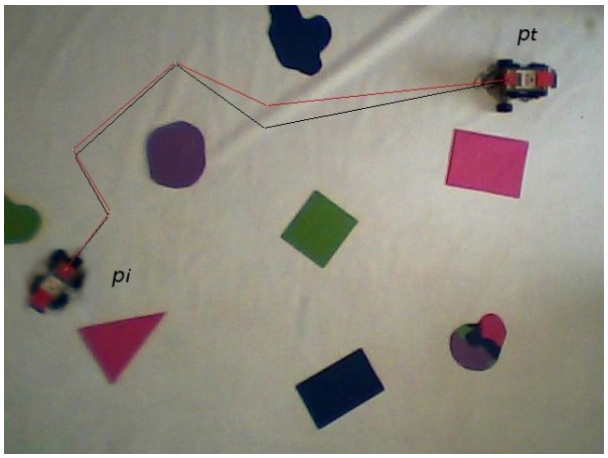

Figure 7. Experimental Test 1 The robot initial point p_i is located at the left-lower part and the final point p_t is located at right-upper part.

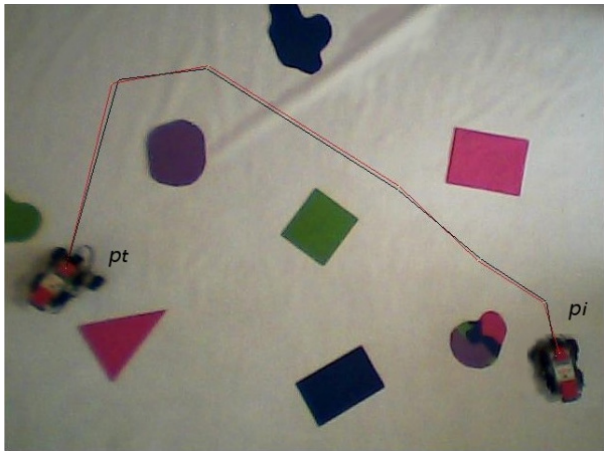

Figure 8. Experimental Test 2- The robot initial point p_i is located at the right-lower part and the final point p_t is located at left-lower part.

Along the 50 different tests (different environments, environment size and obstacle complexity), the proposed algorithm was able to find the 100% of the cases an adequate path, and in the most of cases, it found the one that would be consider as optimum (shortest). Especially, it showed an excellent performance in mazes, environments for which the previous evaluated strategies (Voronoi, Bug, Delaunay and other types of segmentation) always presented difficulties.[14,15] The algorithm also keeps a low computing cost, which allows the use of devices quite modest (between 5 and 8 seconds, using an Intel U7300 Dual Core with 4 Gigabytes of RAM running Windows 7 64 bits and Matlab R2012b), and standing out for its reliability (always find safe navigation paths).

5. Conclusion

The issue of planning robot paths is one of the most important tasks in robotics. In this sense, a solution algorithm is proposed, which is based on the skeletonization of navigation free space combined with a non-informed deep searching. The evaluation of the algorithm both by simulation and as a laboratory prototype confirm not only its capability to find global optimum/sub-optimum paths without falling into convergence problems, but also feasible to the robot, keeping a computing cost extremely low. Due to these features, as future work, to implement the algorithm for dynamic environments is proposed; in that case, it is necessary to update the path at the same time the robot is moving and the dynamics of the environment, like a reactive system.

6. Notation

Γ	Set of defined edges by the skeletonization process of environment free space E .
γ	Each edge of Γ . It is a curve that splits regions of the environment, defined between skeleton vertexes and/or points of ∂W .
d	Diameter of the smallest circle that contains the robot in R^2.
E	Free space in the navigation environment, in which the robot can move.
O	Obstacle in W .
$\mathcal{O}$	Set of obstacles.
p	One point in $W \subset R^2$.
R^q	The q-dimensional Euclidian space.

$\mathcal{R}$	Set of all regions r of W .
r	Region in W .Each region is defined by curves γ and intervals of ∂W .
W	Set that has an open inside connected with obstacles that represent inaccessible areas. Navigation area in the Euclidean space.
∂W	Limit or border of W .

7. Acknowledgments

This work was supported by Universidad Distrital Francisco José de Caldas specifically through "Centro de Investigaciones y DesarroloCientífico" (CIDC). The views expressed in this paper are not necessarily endorsed by Universidad Distrital. The authors thank the research group ARMOS for the evaluation carried out on designs and tests.

8. References

1. Jiang P, Ji Y, Wang X, Zhu J, Cheng Y. Design of a multiple bloom filter for distributed navigation routing. Transactions on Systems, Man, and Cybernetics. 2014 Feb; 44(2):254–60. Crossref
2. Vasile CI, Belta C. Sampling-based temporal logic path planning. Intelligent Robots and Systems; 2013 Jul. p. 4817–22.
3. Diller E, Floyd S, Pawashe C, Sitti M. Control of multiple heterogeneous magnetic microrobots in two dimensions on nonspecialized surfaces. IEEE Transaction Robotics. 2012 Feb; 28(1):172–82. Crossref
4. Xu L. Chow TWS. Self-organizing potential field network a new optimization algorithm. IEEE Transaction Neural Networks. 2010; 21(9):1482–95. Crossref PMid:20570771
5. Bhattacharya P, Gavrilova M.L. Roadmap-based path planning-Using the Voronoi diagram for a clearance-based shortest path. IEEE Robotics & Automation Magazine. 2008; 15(2):58–66. Crossref
6. Dogancay K. UAV path planning for passive emitter localization. IEEE Transactions on Aerospace and Electronic Systems. 2012; 48(2):1150–66. Crossref
7. Sujit PB, Lucani DE, Sousa JB. Bridging cooperative sensing and route planning of autonomous vehicles. IEEE Journal on Selected Areas in Communications. 2012; 30(5):912–22. Crossref
8. Martínez FH, Delgado JA. Hardware Emulation of Bacterial Quorum Sensing. International Conference on Intelligent Computing; 2010. p. 329–36. Crossref
9. Vachhani L, Mahindrakar AD, Sridharan K. Mobile Robot Navigation Through a Hardware-Efficient Implementation for Control-Law-Based Construction of Generalized Voronoi Diagram. IEEE/ASME Transactions on Mechatronics. 2011 Dec; 16(6):1083–95. Crossref
10. Masoud AA. Motion planning with gamma-harmonic potential fields. IEEE Transactions on Aerospace and Electronic Systems. 2012 Oct; 48(4):2786–801. Crossref
11. Cowlagi R V, Tsiotras P. Multiresolution motion planning for autonomous agents via wavelet-based cell decompositions.IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2012 Oct; 42(5):1455–69. Crossref PMid:22581136
12. Fainekos GE. Revising temporal logic specifications for motion planning.Robotics and Automation (ICRA), 2011 IEEE International Conference on; 2011. p. 40–5. Crossref
13. Kloetzer M, Belta C. Automatic deployment of distributed teams of robots from temporal logic motion specifications.IEEE Transactions on Robotics. 2010 Feb; 26(1):48–61. Crossref
14. Santa F M, Sarmiento FHM, Gómez E J. Using the delaunay triangulation and voronoi diagrams for navigation in observable environments. 2014 Dec; 18:81–7.
15. Barrero A, Robayo M, Jacinto E. Algoritmo de navegación a bordo en ambientes controlados a partir de procesamiento de imágenes. 2015; 12 (2):23–34.