VOL. 14, NO. 18, SEPTEMBER 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

HYBRID FREE-OBSTACLE PATH PLANNING ALGORITHM USING IMAGE PROCESSING AND GEOMETRIC TECHNIQUES

Fernando Martínez S., Fredy H. Martínez and Holman Montiel
Technological Faculty, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
E-Mail: fmartinezs@udistrital.edu.co

ABSTRACT

This paper shows a hybrid path planning algorithm based on image processing and geometric techniques, where the main idea is to obtain a free-obstacle path for a mobile robot over a known environment through a camera located in top of it. The algorithm uses image processing operations like dilation, skeletonization, image convolution among others, combined with simple geometric operations like distance between points and middle point calculation. Those operations had the goal to find some amount of navigable points including the starting point to the ending one. The A* algorithm was used to select the shortest combination of navigable points to take the mobile robot through, avoiding all the obstacles in the environment. Some reduction rules were implemented into the proposed algorithm with the purpose of decreasing the number of key points and/or navigable points and thus optimizing the density of the navigable network reducing the total computing time. Finally, the result of testing this approach over a total of 8 different navigation environments are displayed, comparing the number of obstacles and their relationship with the computing time.

Keywords: path planning, image processing, mobile robots, geometrical techniques.

1. INTRODUCTION

Path planning for mobile robots keeps being one of the most researched issues in robotics, due to constantly new challenges appear, such as robot swarm navigation (Agudelo, Moreno, & Hernández, 2017; Gómez, Giral, & Sarmiento, 2016) or 3D navigation as the used for UAVs (Chen, Luo, Mei, Yu, & Su, 2016; Doğançay, 2012; Roberge, Tarbouchi, & Labonte, 2013). A lot of different approaches have been tested for a lot of different navigation targets, basically they are classified in two big branches: strategies that use on-board sensors, and the ones that use external sensors. Mainly, cameras located at the top of the scenery are used as external sensor in the second group, obtaining global images of the robot navigation environment. Based on those images several approaches have been proposed, for instance geometrical algorithms based on digital image processing (Bagyaveereswaran, Rajagopal, & Anitha, 2016; Barrero, Robayo, & Jacinto, 2015) as visibility graphs (Contreras, Martínez S., & Martínez S., 2015; Ganguli, Cortes, & Bullo, 2009). As well as, Voronoi diagrams have been widely used for single robot navigation (Sangeetha et al., 2015) and multiple or swarm robot navigation (Breitenmoser, Schwager, Metzger, Siegwart, & Rus, 2010; Herrero & Martínez, 2012; Mohseni, Doustmohammadi, & Menhaj, 2016; Saeedi, Paull, Trentini, Seto, & Li, 2012; Wei, Mao, Guan, & Li, 2017), and their implementation variations as image skeletonization (Sangeetha et al., 2015). Most of those implementations are combined with selection algorithms as A*, in order to choose the shortest path (Mujtaba & Singh, 2017; Nawaz, Khan, & Lee, 2014). In ARMOS research group at Universidad Distrital, some navigation and path planning strategies have been explored, specifically this paper shows the proposed approach based on digital image processing and geometric techniques, finding the middle points between obstacles and the robot, in order to obtain a navigable network and finally to find the shorter safe navigation path.

2. METHODOLOGY

The proposed approach starts from a pre-processed binary image of the robot navigation environment taken from de up side, where the background is shown in black and the obstacles in white, as shown Figure-1, where the red points represent the starting (P_i) and the ending (P_t) points. Then starting from the binary image, it is necessary to take into account the robot size; this one is done by means of applying a dilation operation over the binary image in order to obtain an obstacle-dilated image version. After, an eskeletonization algorithm is applied not to the free navigation space (similar to Voronoi diagram) but to each dilated obstacle. After that, over the obstacle-skeleton image, the end-points are found to be used to calculate the middle points among obstacles. Finally, a unique path is found using A* algorithm.

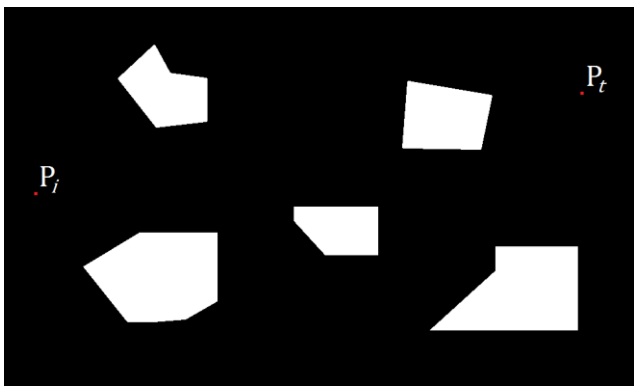

Figure-1. Input pre-processed binary image of the navigation environment.

2.1 Obstacle dilation

In order to take into account the robot size and avoid any kind of collision against the obstacles, the input image (black background and white obstacles) is passed through a binary-image dilation process. This one is a morphological binary operation based on bi-dimensional discrete convolution between the original image and a convolution matrix, as shown in equations (1) and (2), where I is the input image, M_c is the convolution matrix and $I_d$is the result. The resultant image shape depends on the chosen convolution matrix, in this case a disk shape was selected with a radius equivalent to the robot one (turned into pixels). The Figure-2 shows the resultant dilated image.

$$I_d(x,y) = I(x,y) * M_c(x,y) \quad (1)$$

$$I_d(x,y) = \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} M_c(i,j) I(x-i, y-j) \quad (2)$$

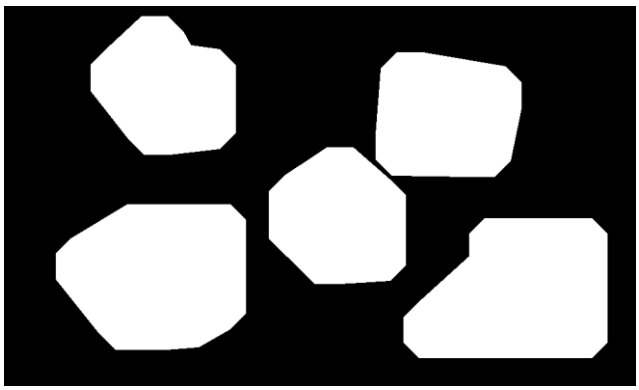

Figure-2. Obstacle-dilated binary image.

2.2 Obstacle skeletonization

From the dilated-obstacle image is obtained the skeleton of each obstacle, through applying the skeletonization algorithm. This one consists of doing iteratively but controlled, erosion operations (similar to dilation but with a different convolution M_c), the main idea is to remove pixels from que obstacles border in each iteration, to make the obstacles thinner. With the purpose of avoiding the obstacle shape disappear, this erosion process is not applied over pixels where another erosion could be applied (from a different border). Finally, a thin line in the middle of each obstacle is obtained, which represents the central or middle line of the obstacle (skeleton). In order to display this skeleton easier, it was shown as al line of 3 pixels wide (see Figure-3).

The ending points of the obstacle skeletons can be used as key points for the navigation, due to represent the extreme points that surround each dilated obstacle.

2.3 Extreme points of the obstacles

Starting from the skeleton image, the end-points were found using also a bi-dimensional convolution operation through a counting 3x3 convolution matrix M_t, which consist of e 3x3 matrix full of '1's. After doing the convolution, the pixels with 2 as a result (the ones with only 2 neighbors) are considered as end-points E_p, as shown in Figure-4 (highlighted as squares of 3x3 pixels), after applying equation (3).

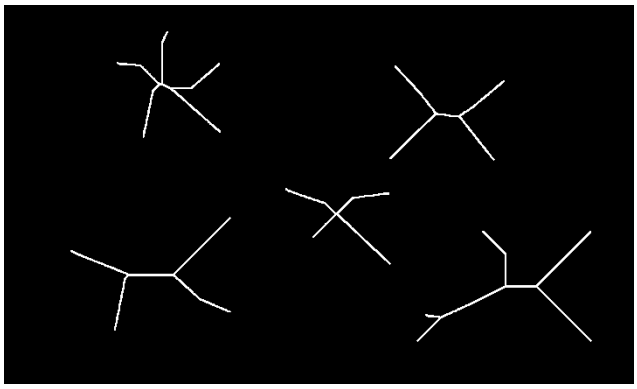

Figure-3. Skeleton of the dilated obstacles.

$$\forall\, n \in (I_s(x,y) * M_t(x,y)) : n = 2 \Rightarrow n \in E_p \quad (3)$$

Those skeleton end-points represent the extreme points of each obstacle, considering the maximum radius of the robot (by means of using the dilation of the obstacle image), so these extreme points surround each obstacle holding an appropriated free-collision distance (see Figure-5). Depending on the obstacle shape complexity the resultant skeleton can have from 4 to 8 branches (Martinez & Jacinto, 2017), which would produce the same amount of extreme points.

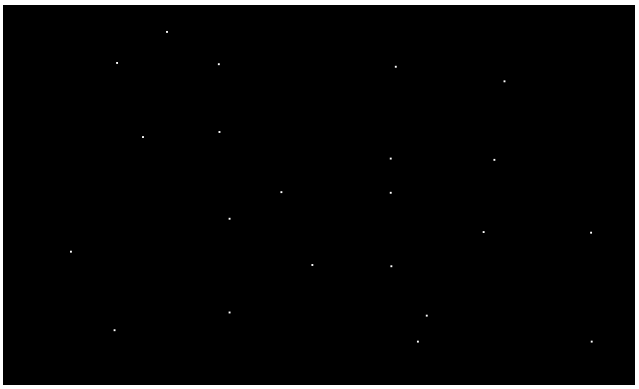

Figure-4. Extreme points of the skeleton.

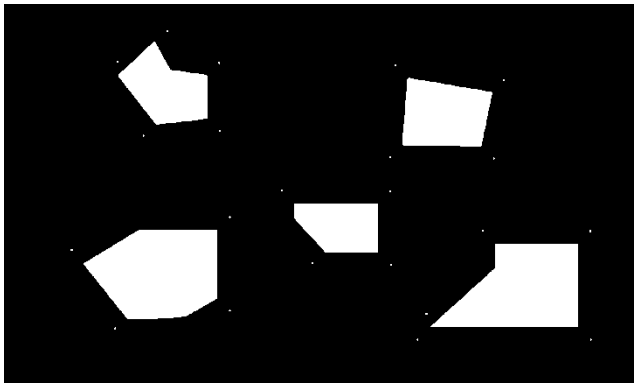

Figure-5. Key points of each obstacle.

2.4 Key navigation points

Based on the extreme points of each obstacle the key navigation points were calculated as the middle points between obstacles, following the next steps:

a) By each couple of obstacles in the environment, a maximum of 2 extreme points are selected, taking into account that the line generated by each couple of points does not cross any obstacle and at the same time it is as short as possible (nearest points).
b) The middle point between each couple of extreme points selected in the step 1 is calculated.
c) The starting and the ending points are added to the rest of navigation key points.
d) The middle point between the starting point and the nearest extreme point of each obstacle is calculated following the same crossing and shortness conditions of the step 1.
e) The middle point between the ending point and the nearest extreme point of each obstacle is calculated following the same conditions of the step 1.

The resultant navigation key points are shown in the Figure-6 in green, an in red are shown the starting and the ending points.

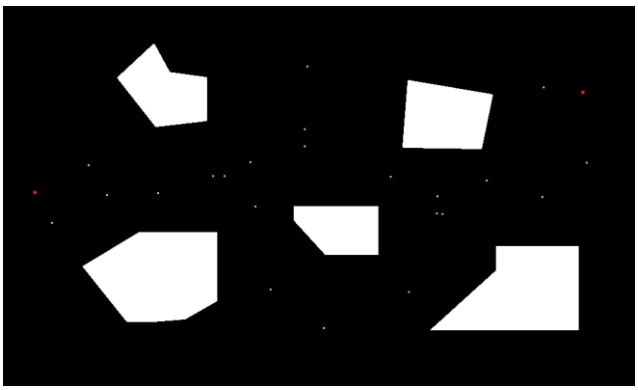

Figure-6. Global navigation key points (green), including the starting and the ending point (red).

2.5 Navigation network

The obtained key points were joint to obtain a navigation network, it means a network of all the possible paths that connect the starting and ending points passing through the key ones. First, all the possible connections were done (excluding the lines that passes through obstacles), but in order to reduce the final path selection time, the next steps were followed:

a) By each key point (including starting and ending points) the first 4 nearest key points are detected and connected by lines to the current key point.
b) By each generated connection it is necessary to discard the lines that cross any obstacle.

Finally, the reduced connection network is obtained as shown un Figure-7. This network is reduced due to the limitation of 4 connection by obstacle.

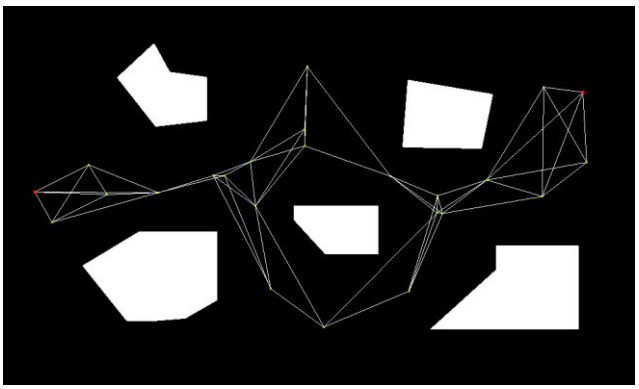

Figure-7. Global navigation network limited to 4 connections by key point.

2.6 Path selection

On the obtained navigation network, a selection algorithm was run in order to obtain the shortest path that connects the starting and ending points, in this case, the A* algorithm was applied to the navigation network, using the summation of distances between connected points as the algorithm goal. The resultant image is shown in Figure-8.

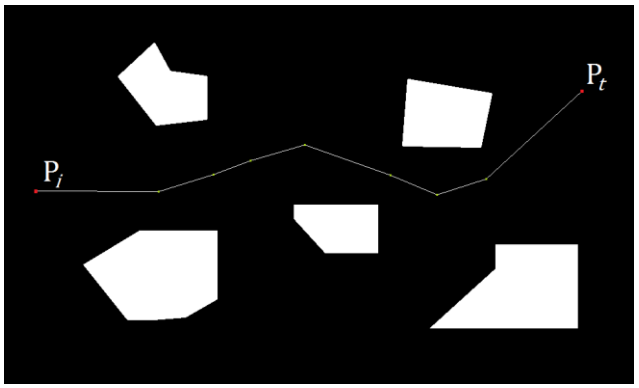

Figure-8. Resultant path.

3. RESULTS

The proposed algorithm was applied to a total of 8 different navigation stages, for which, always a valid path was found. All stages used from 3 to 6 obstacles by each one, those obstacles had simple shapes as shown in the figure sequence producing from 4 to 6 extreme points per obstacle.

Table-1. Simulation tests comparative.

Stage	Obstacle number	Extreme points	Navigation points	Time [seg]
1	3	14	15	1,84
2	3	13	15	1,74
3	4	16	17	2,28
4	4	18	20	2,9
5	5	22	22	3,76
6	5	23	24	4,24
7	6	27	26	5,29
8	6	26	28	5,47

Table-1 shows a summary of the 8 simulation tests realized. It displays the number of obstacles, the total extreme points, the total points of the navigation network and finally the computing time of all the algorithm.

The resultant time behaves exponentially and depends on the number of obstacles in the environment. This time was reduced a maximum of 87% reducing the complexity of the navigation network. This was carried out applying some strategies of reduction of key points and finally the navigation points. First, just two connection points between each couple of obstacles were selected, second, only one middle point between the starting and ending points and the nearest obstacle was calculated, and finally, with the purpose of computing the navigation network, the steps described in the section 2.5 were applied instead of doing all the possible connections among the key points. In conclusion, the density of the navigable network (Figure-7) was highly optimized to reduce the input information to the selection algorithm A*, and thus reducing the total computing time.

The proposed algorithm improved the performance and final path shape (distance to the obstacles) regarding the previous works like the exposed in (Martinez & Jacinto, 2017), taking advantage in a better way of the free navigable space of each environment.

4. FUTURE WORK

This approach only takes into account key points from obstacles and the starting and ending points, but according with the results the selected paths prefer to go far to the environment borders, discarding some interesting results. Then it is possible to improve the algorithm using the environment corners and other points from the border as key points.

Finally, it is important to test the proposed algorithm on real mobile robots, for which it would be interesting to previously test it on environments with walls, it means, environments based on architectonic plans.

ACKNOWLEDGMENTS

This work was supported by the District University Francisco José de Caldas, in part through CIDC, and partly by the Technological Faculty. The views expressed in this paper are not necessarily endorsed by District University. The authors thank the research group ARMOS for the simulations and tests done.

REFERENCES

Agudelo J. S. B., Moreno R. J. & Hernández R. D. 2017. Virtual Environment for Collaborative Robotic Agents.

Bagyaveereswaran V., Rajagopal N. N. & Anitha R. 2016. Image Based Autonomous Navigation for a Lander. International Journal of Applied Engineering Research. 11(4): 2424-2428.

Barrero A., Robayo M.& Jacinto E. 2015. Algoritmo de navegación a bordo en ambientes controlados a partir de procesamiento de imágenes. Tekhnê. 12(2): 23-34.

Breitenmoser A., Schwager M., Metzger J.-C., Siegwart, R. & Rus D. 2010. Voronoi coverage of non-convex environments with a group of networked robots. In Robotics and Automation (ICRA), 2010 IEEE International Conference on (pp. 4982-4989). IEEE.

Chen Y., Luo G., Mei Y., Yu J. & Su X. 2016. UAV path planning using artificial potential field method updated by optimal control theory. International Journal of Systems Science. 47(6): 1407-1420.

Contreras J. D., Martínez S., F. & Martínez S., F. H. 2015. Path planning for mobile robots based on visibility graphs and A* algorithm. In C. M. Falco & X. Jiang (Eds.), Seventh International Conference on Digital Image Processing (ICDIP15) (p. 96311J). International Society for Optics and Photonics. http://doi.org/10.1117/12.2197062

Doğançay K. 2012. UAV path planning for passive emitter localization. Aerospace and Electronic Systems, IEEE Transactions on. 48(2): 1150–1166.

Ganguli A., Cortes J. & Bullo F. 2009. Multirobot Rendezvous With Visibility Sensors in Nonconvex Environments. IEEE Transactions on Robotics. 25(2): 340-352. http://doi.org/10.1109/TRO.2009.2013493

Gómez E. J., Giral M. & Sarmiento F. H. M. 2016. Modelo de navegación colectiva multi-agente basado en el quorum sensing bacterial. Tecnura. 20(47): 29-38.

Herrero D. & Martínez H. 2012. Range-only fuzzy Voronoi-enhanced localization of mobile robots in wireless sensor networks. Robotica. 30(7): 1063-1077.

Martinez F. & Jacinto E. 2017. Navigable points estimation for mobile robots using binary image skeletonization. In Eighth International Conference on Graphic and Image Processing (ICGIP 2016) (Vol. 10225, p. 1022505). International Society for Optics and Photonics.

Mohseni F., Doustmohammadi A. & Menhaj M. B. 2016. Distributed receding horizon coverage control for multiple mobile robots. IEEE Systems Journal. 10(1): 198-207.

Mujtaba H. & Singh G. 2017. Safe Navigation of Mobile Robot Using A* Algorithm. International Journal of Applied Engineering Research. 12(16): 5532-5539.

Nawaz W., Khan K. U. & Lee, Y.-K. 2014. Shortest path analysis for efficient traversal queries in large networks. Proc. Contemporary Engineering Sciences. 7(16): 811-816.

Roberge V., Tarbouchi M. & Labonte G. 2013. Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. IEEE Transactions on Industrial Informatics. 9(1): 132-141. http://doi.org/10.1109/TII.2012.2198665

Saeedi S., Paull L., Trentini M., Seto M. & Li H. 2012. Efficient map merging using a probabilistic generalized voronoi diagram. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on (pp. 4419–4424). IEEE.

Sangeetha V., Vaithiyanathan V., Sivagami R., Divyalakshmi, K., Sundar, K. J. A. & Ahmed M. I. 2015. Skeletonisation Approaches for Determining Paths in an Image: A Review. Indian Journal of Science and Technology. 8(35).

Wei H.-X., Mao Q., Guan Y. & Li Y.-D. 2017. A centroidal Voronoi tessellation based intelligent control algorithm for the self-assembly path planning of swarm robots. Expert Systems with Applications. 85, 261-269.