Al-Khwarizmi Engineering Journal, Vol. 7, No. 4, PP1-16 (2011)

Al-Khwarizmi Engineering Journal

# Path Planning Control for Mobile Robot

**Nabeel K. Abid Al- Sahib  Amenah A.H. Salih**

*Department of Mechatronics Engineering/ Al-Khwarizmi College of Engineering/University of Baghdad*

Email:dr-nabeelalsahib@hotmal.com

(Received 16 February 2011; Accepted 20September 2011)

## Abstract

Autonomous motion planning is important area of robotics research. This type of planning relieves human operator from tedious job of motion planning. This reduces the possibility of human error and increase efficiency of whole process.

This research presents a new algorithm to plan path for autonomous mobile robot based on image processing techniques by using wireless camera that provides the desired image for the unknown environment . The proposed algorithm is applied on this image to obtain a optimal path for the robot. It is based on the observation and analysis of the obstacles that lying in the straight path between the start and the goal point by detecting these obstacles, analyzing and studying their shapes, positions and points of intersection with the straight path to find the nearly optimal path which connects the start and the goal point.

This work has theoretical part and experimental part.

The theoretical part includes building a MATLAB program which is applied to environment image to find the nearly optimal path .MATLAB - C++.NET interface is accomplished then to supply the path information for C++.NET program which is done for programming the pioneer mobile robot to achieve the desired path.

The experimental part includes using wireless camera that takes an image for the environment and send it to the computer which processes this image and sends ( by wireless connection ) the resulted path information to robot which programmed in C++.NET program to walk according to this path.

So, the overall system can be represented by:

Wireless camera – computer – wireless connection for the mobile robot .

The experimental work including some experiments shows that the developed mobile robot (pioneer p3-dx ) travels successfully from the start point and reach the goal point across the optimal path (according to time and power) which is obtained as result of the proposed path planning algorithm introduced in this paper.

***Keywords:****Path planning algorithms, Pioneer P3-DX mobile robot, Image processing, Minimum time and power.*

## 1. Introduction

Mobile robots must know where they are and specify a suitable path to the goal to navigate reliably in indoor environments; this is the task of navigation and path planning to enable the robot to reach the goal point with optimal path without any knowledge about the environment, by using wireless camera as feedback sensor. So it is obvious that an efficient control of the mobile robot needs:

• A local level based completely on the information of different sensors covering the close circle of the robot.

• A high-level for path planning using a global description of the world with possibly incomplete and/or imperfect knowledge.

This allows having a safe navigation to modulate continuously the path and eventually to update the map.

This work deals with global path planning in which the robot is sent to completely unknown environment (unknown environment's dimensions, obstacle's shape, position and dimension, number of obstacles, distribution of obstacles or any other information), and as soon as the robot reachs to this environment, a wireless camera will send it information as an image by wireless connection to

computer, the computer finds the required path and the information of this path will send to the robot to do it's task to reach the goal point in an approximately optimal path [1].

## 2. Pioneer Mobile Robot

The platform used in work is Pioneer3 DX which is a family of two wheels mobile robots, the newest Pioneer3 DX is used for university research and the platform shares a common architecture and foundation with all other MOBILE ROBOT platforms including AmigoBot™, PeopleBot™ V1, Performance PeopleBot™ and PowerBot™ mobile robots.

The robot is built completely with a strong aluminum body; a balanced drive system (two-wheel differential with free caster wheel), reversible DC motors, motor-control and drive electronics, high-resolution motion encoders, and battery power, all are managed by an onboard microcontroller and mobile-robot server software. The software includes foundation Advanced Robotics Interface for Applications (ARIA) and AR Networking

Pioneer3 robot has a variety of expansion power and I/O ports for attachment and close integration of a client PC, sensors, and a variety of accessories, all are accessible through a common application interface to the robot's server software [2].

The mobile platform robot (Pioneer3 DX) used is shown in figure (1).

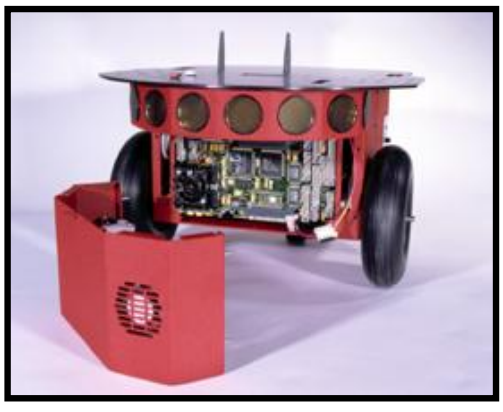


**Fig.1. Pioneer Mobile Robot [2].**

## 3. OAC Path Planning Algorithm

This work present an algorithm for planning the path for the robot which is based on detailed analysis and observation of the obstacles lying in the supposed straight line path between start and goal point.

When the robot detect on obstacle in its straight direction, the path will be out of this direction and it tends to turn around the obstacle (either clockwise or counterclockwise) until it reach the straight direction again.

The phenomena that the path follows it to achieve this movement result from observing the obstacle shape, its dimensions and the points of intersection with straight direction that connects start and goal point.

This comes from detailed analysis for these obstacles and all possible cases of intersection that could be found the upper and lower forms of paths.

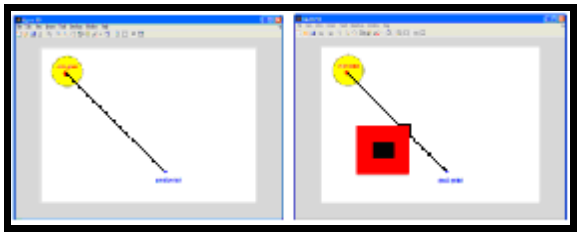
This phenomenon is based on the observation, analysis and conclusion, we can call the algorithm proposed here and which depends on the principles the observation, analysis and conclusion based algorithms. For this reason this algorithm can be called (OAC) algorithm according to the first letters of its principles.

## 4. OAC Algorithm Explanation and Definition

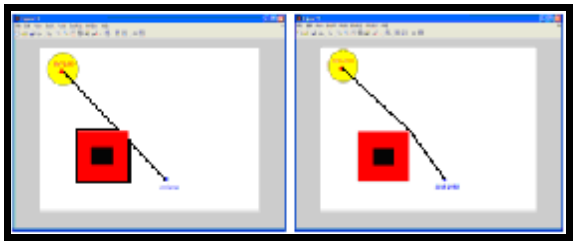
In this research, it is intendended to find the optimal or nearly optimal path that connects the start and goal point without collision with the obstacles which are found in its direction .If we suppose that there is no obstacle in the straight direction between the start and goal points the path will be purely straight line as shown in fig. (2-a). If an obstacle is located between the start and goal points, it will intersect with this straight line. At this point this straight path must change its direction in order to avoid collision with the obstacle, this is done by making the path turning around the obstacle (adjacently with edges either clockwise or counter clockwise) until it exceeds the obstacle and return again to its straight direction towards the goal point , this behavior is always repeated with any other obstacle located an its straight direction . Finally two paths resulted from this behavior:

- The upper path : is the path that resulted as a combination between two merged paths, the first one is represented by a straight line (in the no-obstacle area) and the other represents the surrounded part of the path that is surrounding the obstacles in clockwise manner as shown in figure(2-b).

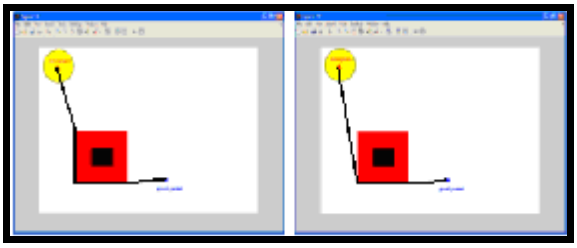
- The lower path : is the path resulted as a combination between two merged paths, the first one is represented by a straight line (in the no-obstacle area) and the other represents the surrounded part of the path that is surrounding the obstacles (in the area that contains obstacles) in counter clockwise manner as shown in fig. (2-c ).



**(a) The supposed straight line b) The upper path**



**(c) The lower path (d) Modified upper path**



**(e) Intermediate Lower Path (f) Modified Lower Path**

**Fig .2. All the Possible Paths.**

**Note**: Red point represents the start point.

Blue point represents the goal point.

Black rectangular represents the obstacle.

Red area represents the clearance

(Explained later).

Yellow circle represents the robot radius.

As shown in figures (2-b, c), the upper and lower paths are not the optimal path. In order to find the optimal path, we follow the following steps:

- Categorize the indiviual lines that form the upper and lower paths into two groups:

**Group A**: it is the group of all segments (individual lines) of the upper or lower path found from straight line equation (the parts of the path in space or for obstacles area).

**Group B**: it is the group of all segments (individual lines) of the upper or lower path forming a surrounded path around the obstacles.

- Finding the Resultant: If any line from group A is followed by a line (or successive lines) from group B, they will be replaced by the resultant of them. In opposite, if any line (or successive lines) from group B is followed by any line from the group A, they will be replaced by the resultant of them.
- Finding Tested path: it is the path that results from finding successive resultants for each n-line (from the upper or lower paths), so the tested path is combined from these individual resultants. This path is called according to (OAC) algorithm the tested path because it must submit to a specific test to judge if it is the optimal ,nearly optimal or not.

An important condition must be satisfied when finding each individual resultant that it must be checked (according to environment's image), if there is an obstacle intersects with it on or not.

The specific test is summarized in the following stages:

- If there is an obstacle in the direction of the resultant, this resultant must resolved to its original components, these components are recognized as forbidden lines (referring to its inability to be replaced with its resultant).
- If the direction of this resultant is free-obstacle this resultant is kept and takes its position in the tested path and it is known as unforbidden lines.
- The resulted path from step 1 and 2 can be retested again by finding the resultant for only unforbidden lines (while keeping forbidden lines).
- Step 3 continues until there is no resultant to be found (mostly this occurs one or two times).

Finally, we will get two new paths, the first resulted from the upper path which is called modified upper path and the other resulted from the lower path which is called modified lower path. These two paths with addition to original upper and lower path and all the resulted intermediate paths are examined and the results of the examination are compared with each other in order to determine the optimal path.

For example, the upper and the lower paths which are shown in fig. (2-b, c) are submitted to the previous test. So, the upper path will converted into modified upper path as shown in fig. (3-d) and the lower path will be converted into intermediate lower path and then into amodified lower path as shown in fig.( 2-e,f).

## 5. The Specific Examination for Finding Optimal Path

The following examination is done on all the resulted paths from (OAC) in order to find the optimal path which is accomplished by checking all the resulted paths with respect to:

- **Length of each path**: The length of the path is considered an important factor in determining the best path because the robot will consume less time to reach the goal point when the path is shorter.
- **Number of segments**: The number of segments of the resulted path is associated with number of turns that the robot must do to access the desired path. It is considered an important factor in finding the best path because the robot's motors will consume less power to reach the goal point when the path contains less number of segments, because it requires less number of turning to change its direction to follow the next segment direction.

### 5.1 Clearance Factor (C)

It is worth mentioning that the size of the obstacles which are submitted to (OAC) algorithm is larger than their actual size because a factor which is called clearance factor is added to their dimensions in order to prevent the robot from penetrating the obstacles or being in contact with them.

The clearance ( c ) is:

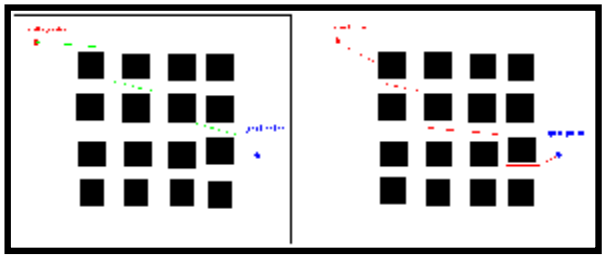
$$c = R + t + s$$

where:

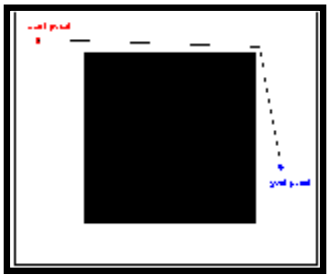
R = radiuas of the plateform robot .
t = thickness of the wheels of the robot.
s = safety factor .

#### 5.1.1 The Effective Role for Factor (S)

Factor ( s ) is assumed by ( OAC ) algorithm in order to ensure that the robot when it is turning around the obstacle to follow the specific planned path, it will never penetrate the obstacles or be in contact with their edges, so it must be added to the obstacle's dimensions. The value of this factor is free and is left for the programmer of the robot who decides this value with respect to the environment, robot dimensions, the obstacles' dimensions and their distribution within the environment which the robot must navigate within it. According to the properties of the environment, pioneer mobile robot and the obstacles which are used in this experimental work, the value of factor ( s ) ranging ( from 5 to 10 cm) and this is considered its normal value in most of other cases. The free selection of the factor ( s ) value gives it an importance which clearly appeared in some complex environments that contain large number of close obstacles. This type of environments is called crowded environments. In this environments the distances between the obstacles are enough for the robot to navigate between them according to the planned paths, but these paths are not practical, because they have large number of curves and large length as shown in figures ( 3 – a ,b). So, with clever selection of factor ( s ), we can conclude these distance and merge these obstacles with each other to form a single irregular obstacle which can be regulated ( by converting it into single rectangular obstacle ) and then this obstacle is submitted to ( OAC ) algorithm to find the optimal path as shown in figure ( 3.c) .



**(a) Modified upper path for small s** **(b) Modified lower path for small s**



**(c) Optimal path for large s**

**Fig.3. Crowded Environment.**

## 6. Obstacle Regulation

To make the (OAC) algorithm capable of being applied for all obstacles, it is suggested that their shapes must be converted into regular shapes (it is a supposed term that refer to the specific shape which can be submitted to (OAC) algorithm which is the rectangular shape only while all other shapes such as atriangle is considered irregular shape), so any other shape must be converted into a rectangular shape. We regulate the obstacle's shape by building a MATLAB subprogram which is part of (OAC) algorithm.

## 7. MRL and MRS

As explained previously, the (OAC) algorithm gives us a number of paths. The modified upper and the modified lower path (the optimal path is one of them) are resulted from achieve modifications on the upper and lower paths respectively. This modification is done either in the lengths of these paths or on the number of the individual lines (segments) which form them. We suggest two formulas to find the value of these modifications for each case study:

Modification rate with respect to path length (MRL);

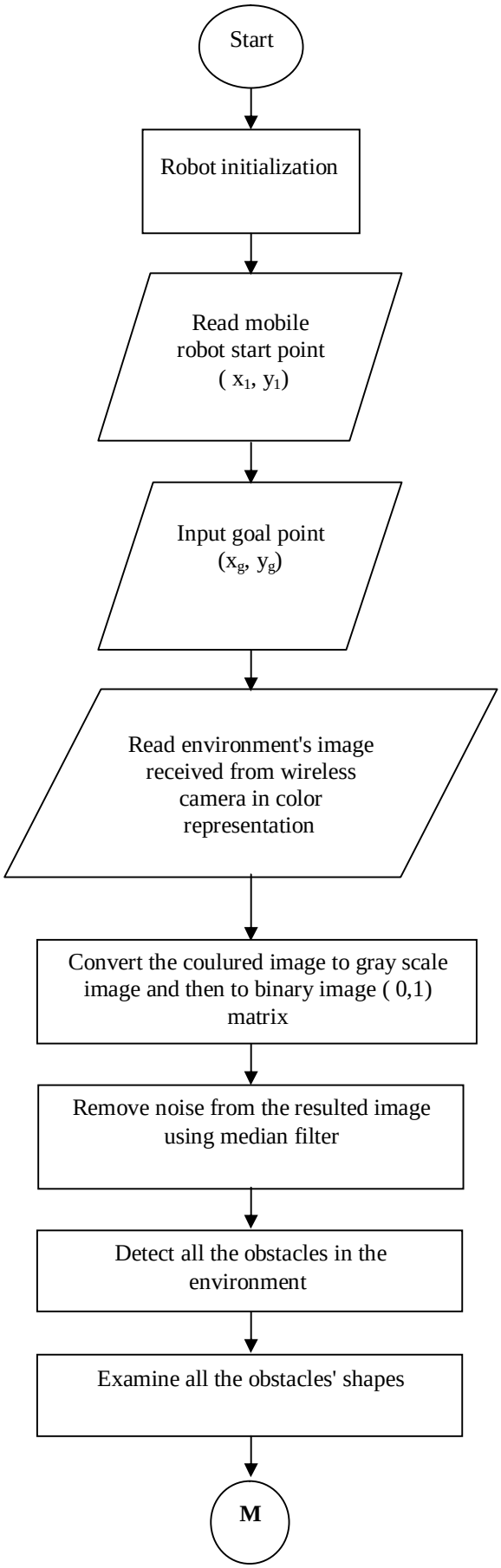
$$(\text{MRL}) = \frac{\text{Largest path length} - \text{smallest path length}}{\text{Largest path length}} * 100\ \%$$

Modification rate with respect to no. of segment (MRS);

$$(\text{MRS}) = \frac{\text{Largest path no. of segment} - \text{smallest path no. of segment}}{\text{Largest path no. of segment}} * 100\%$$

## 8. OAC Flow Chart

Figure (4) shows the flowchart of both MATLAB and C++.NET programs to implement OAC algorithm with all the required details.



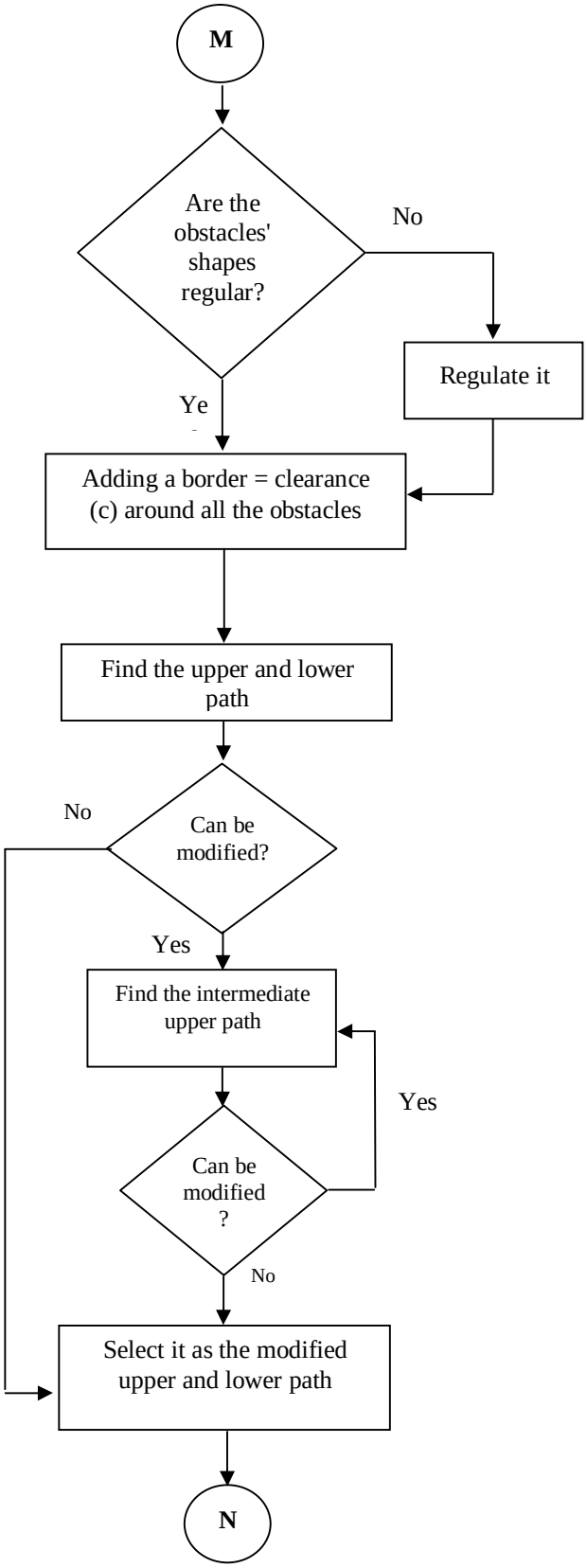



**Fig.4. The Program Flowchart**.

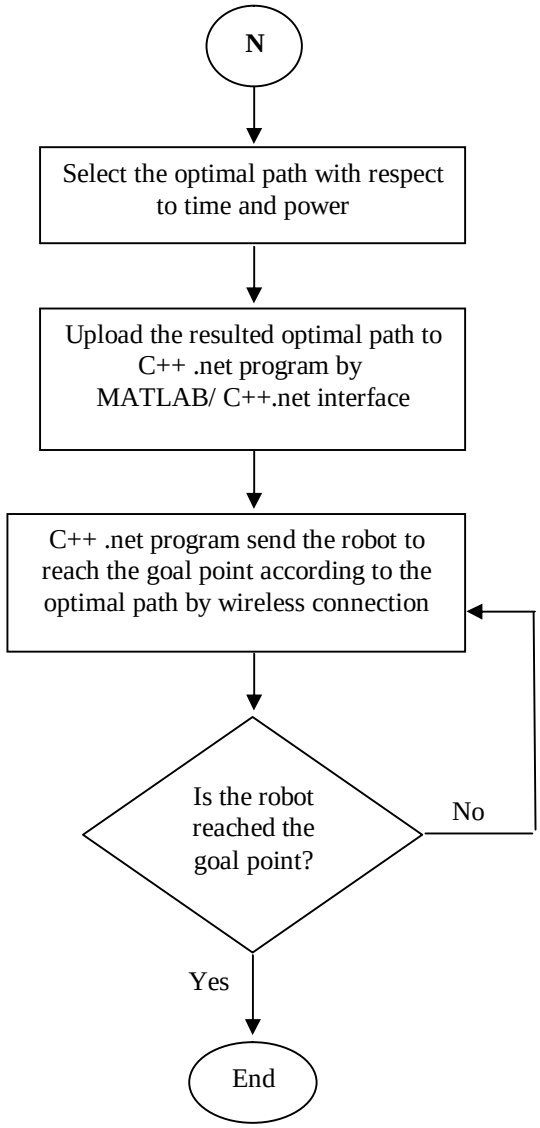



**Figure (4) Continued**

## 9. Simulation and Experimental Work

The work in this paper can be divided into two parts:

- Simulation: It enables us to see the original image, the processed image, the (OAC) algorithm steps and all the resulted paths .This is accomplished by MATLAB program.
- Experimental work: This part include the programming of the robot with C++.NET program by wireless connection in order to reach the goal point according to optimal path.

## 9.1. Simulation Using MATLAB

In order to get a complete the image about the overall work, and be able to see all the results before the order is being sent to robot, the process can be simulated so that we can see the original environment's image, the noised image, the converting steps of this image, the filtered image, the OAC algorithm steps and all the resulted paths and the optimal path.

MATLAB program which is built in order to achieve image processing and the (OAC) algorithm steps can also be used as a simulation tool to display all the above information to be clear for the programmer.

This is considered important requirement to enable the programmer to see the results, to modify the program if required and to explore the error if it is found and determine in which stage of the process it happened.

## 9.2. Experimental Work

The (OAC) algorithm required the environment's image to be processed to and find the optimal path instantaneously. This required a camera as shown in figure (5) ready in any moment to offer the image of environment. For this purpose a wireless camera is used here to accomplish this task.

After the image is sent to MATLAB program (by camera – computer wireless connection) and as soon as the approximately optimal path is found, the path information is processed by the built C++.NET program in order to program move according to this path until it reach the goal point without collision with obstacles found on its way . This task represents the experimental side of this work.

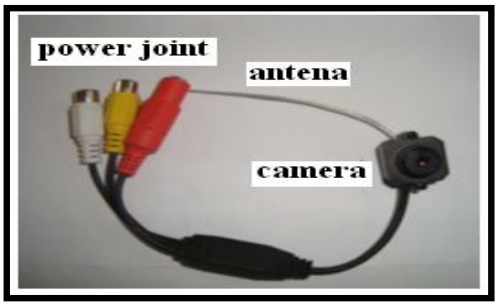



**Fig.5. Wireless Camera.**

In work, we use this camera by fixing it above any unknown environment, with a suitable height that offer the required image requirement and cover nearly all the area of the work which is nearly (2.4 m x 3.2 m) square area. The wireless camera is suited above the environment by using aluminum bar with length of (4 meter) as shown in figure (6) and the group is fixed on height of (5 meter).

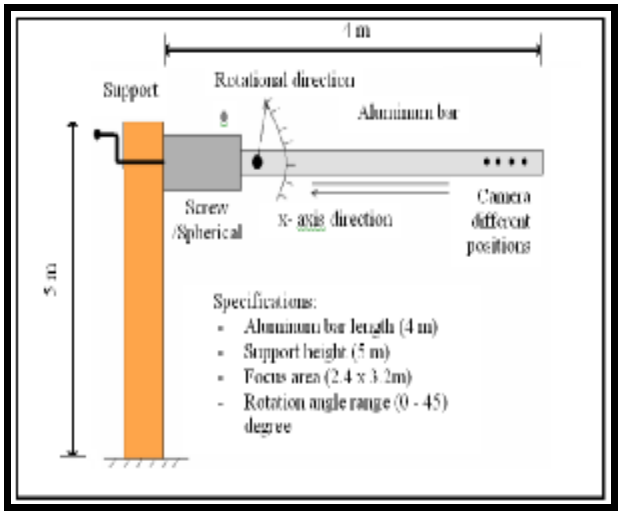



**Fig.6. Modeling Camera – Aluminum Bar Connection.**

## 9.3. Model Environment

The environment is chosen to be a rectangular area with dimensions (2.4 m x 3.2 m) depending on the wireless camera specification .The group of environment, wireless camera, pioneer mobile robot and obstacles can be modeled as in figure (7).

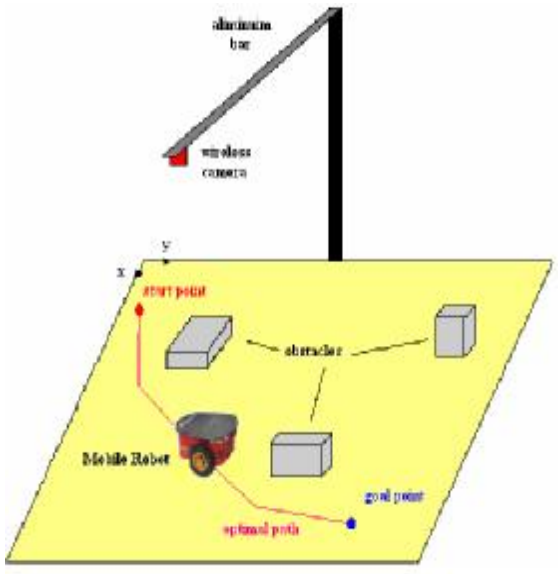



**Fig.7. Model Environment.**

## 10. Path Planning Cases Studies

A number of experiments were carried out (see table 1) as shown in figures (8-16), to test the ability of the mobile robot to reach its goal with approximately optimal path found according to (OAC) algorithm.

These experiments, represented by the cases studies explained in the next section, are done depending on the built MATLAB program to find the optimal paths and then these results are transferred by MATLAB – C++.NET Interface to the C++.NET program in order to program the pioneer mobile robot to take this path through its navigation from the start point to reach the goal point.

**Table 1,**
**Different Cases Studies**

| Index | Case Study | Description |
|---|---|---|
| 1 | Case study 1 | An environment with single obstacle |
| 2 | Case study 2 | An environment with two obstacles |
| 3 | Case study 3 | An environment with two irregular obstacles |
| 4 | Case study 4 | An environment with three obstacles |
| 5 | Case study 5 | An environment with three obstacles / for different start and goal points |
| 6 | Case study 6 | An environment with three obstacles including irregular shapes |
| 7 | Case study 7 | An environment with four obstacles |
| 8 | Case study 8 | An environment with six obstacles |
| 9 | Case study 9 | An environment with crowded obstacles |

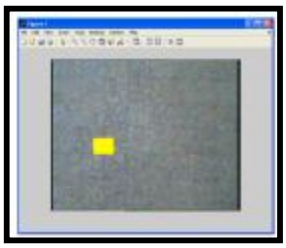


**Fig.8. Case Study 1**

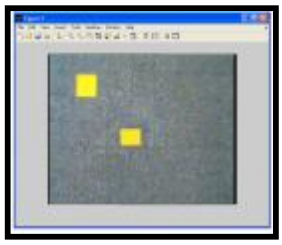


**Fig.9. Case Study 2**

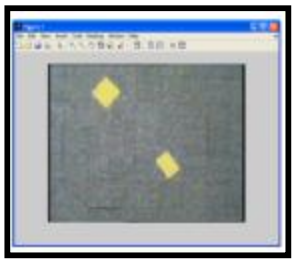


**Fig.10. Case Study 3**

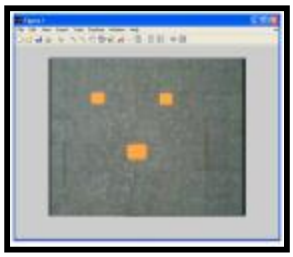


**Fig.11. Case Study 4**

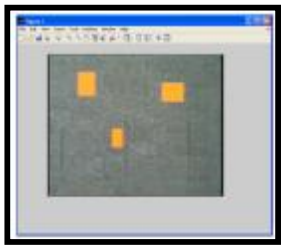


**Fig.12. Case Study 5**

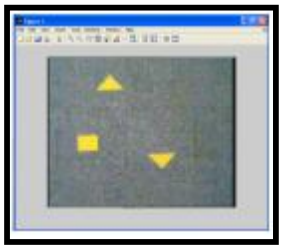


**Fig.13. Case Study 6**

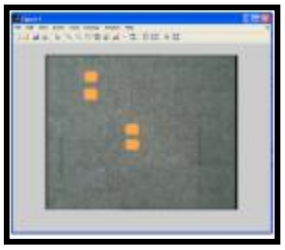


**Fig.14. Case Study 7**

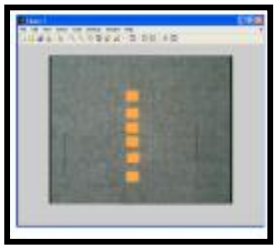


**Fig.15. Case Study 8**

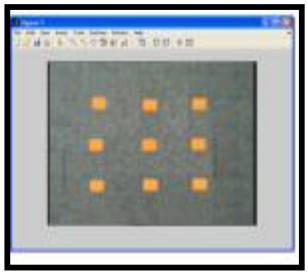


**Fig.16. Case Study 9**

One of these case studies (case study 6) will be explained in details as follow:

## 11. An Environment with Three Obstacles Including Irregular Shapes

The environment which is shown in figure (17) of dimensions (2.4m X 3.2m) contains three obstacles one of them is regular (rectangular 20 cm X 30 cm in area) and the others are irregular (triangle shape is considered irregular according to OAC algorithm principles of 20 cm height and 30 cm base), the start and goal point are (437, 53) and (159, 465) respectively.

This image is sent by wireless connection to the MATLAB program to apply OAC algorithm on it, so the results are as follows:

- Convert this colored image into a gray scale image as shown in figure (18).
- Convert this grayscale image into binary image (0, 1 representation) by selecting suitable threshold as shown in figure (19), note that the image is noised with pepper noise.
- Remov noise from the image and regulate the obstacle's shape. Since the image is highly contemplating with noise it is required to be applied to the filter twice in order to remove the noise completely as shown in figure (20).
- Add a border of thickness = c = 67 pixels (as previously explained in chapter 3) to ensure that the robot will not be in contact with any obstacle through its navigation according to the specified path as shown in figure (21).
- Detect the obstacle edges by building programming steps to find it as shown in figure (22).
- Find the upper path, note that the path is at a distance = c (represented by red color) from the obstacle in order to prevent the robot to touch obstacle and to enable it to move according to its path freely. From this path the intermediate and modified upper path are found as shown in figure (23).
- Find the upper path, note that the path is at a distance = c (represented by red color) from the obstacle in order to prevent the robot from touching the obstacle and to enable it to move according to its path freely. From this path the intermediate and modified upper path are found as shown in figure (23).

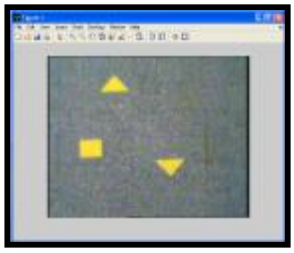


**Fig.17. the Environment Image**

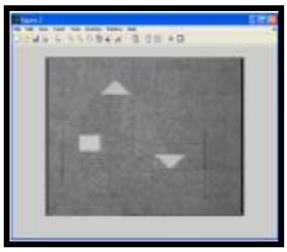


**Fig.18. The gray Scale Image**

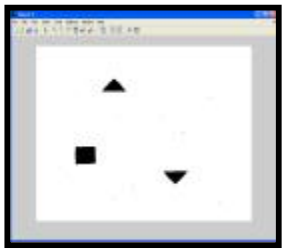


**Fig.19.Binary Image**

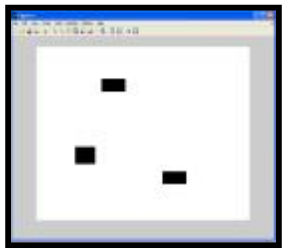


**Fig.20.The Filtered and Regulated Image**

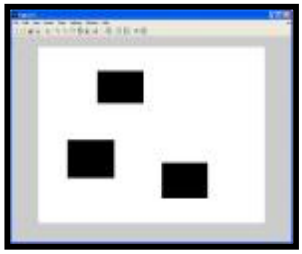


**Fig.21. The Resulted Image After Adding a Border = c**

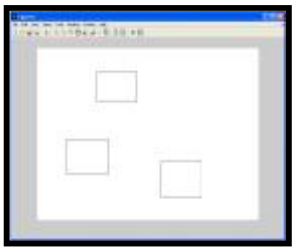


**Fig.22. Detect the Edges of Obstacle**

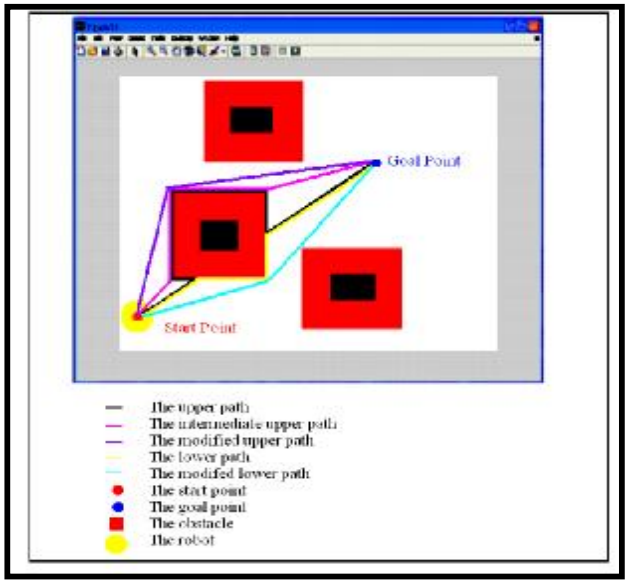



**Fig.23. The All Resulted Paths According to OAC Algorithm.**

As shown, five paths resulted in this case study and according to the criteria of determining the optimal path with respect to (OAC) algorithm the optimal path is the modified lower path because it is the smallest one in length and in number of segments, see table (1).

The above results represent the MATLAB simulation of this case. When the optimal path is sent for pioneer mobile robot, it successfully reaches the goal point without collision with any obstacle in the area as shown in figure (24).

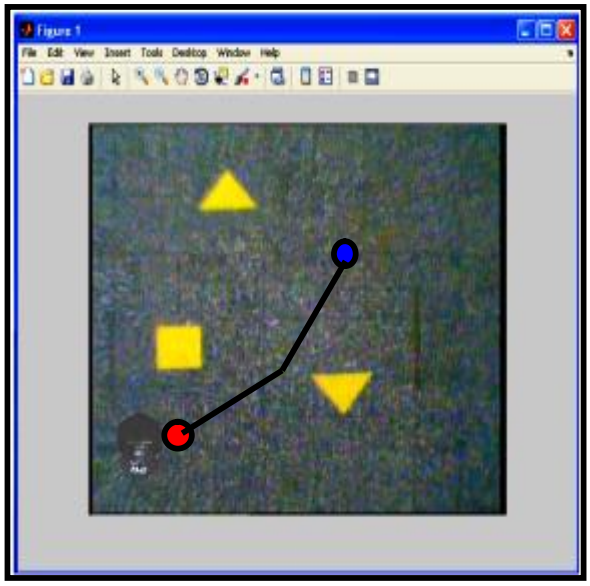


**Fig.24. The Robot Navigation According to the Optimal Path**

## 12. Results and Discussion

The results were for different cases by taking different images and comparing between the behavior of path and navigation of the robot in unknown environment, changing obstacles number and positions and their effectiveness on the resulted paths (specially the optimal path), time required and power consumed for each path.

The experimental work includes the following specifications:

1. The motion along the straight segments with constant speed (zero acceleration). This constant speed is 0.375 m / sec.
2. The motion at turning parts of the path with constant acceleration which is 0.1875 m / $sec^2$.
3. Radius of drive wheels r = 8.25 cm = 0.0825 m.
4. Robot width b = 425 mm = 0.425m.
5. Robot mass m = 9 Kg.
6. No-load speed V max = 24.7 m/sec.
7. The connection type is a wireless Ethernet-to-serial * is connected directly to robot's microcontroller; it works by automatically translating network-based Ethernet packet communications TCP/IP into streaming serial for the robot microcontroller and back again.
8. Battery = 12 volt.

According to (OAC) algorithm the table of results for all cases studies is shown in table (2) and the optimal paths for all cases studies are shown in table (3).

**Table 2,**
**Resulted for All Cases Studies in Details**

| Case study | No. of paths | Resulted path | total length cm | Time sec | No. of segment | K.E Joul |
|---|---|---|---|---|---|---|
| Case study 1 | 1 | Upper path | 244 | 9.5 | 4 | 0.563K |
| | 2 | Modified upper path | 240* | 7.5 | 2 * | 0.1875 K |
| | 3 | Lower path | 423 | 16.28 | 6 | 0.938 K |
| | 4 | Intermediate lower path | 320 | 11.53 | 4 | 0.563 K |
| | 5 | Modified lower path | 308 | 10.2 | 3 | 0.375 K |
| Case study 2 | 1 | Upper path | 304 | 14.11 | 7 | 1.125 K |
| | 2 | Modified upper path | 276 * | 9.36 | 3 * | 0.375 K |
| | 3 | Lower path | 628 | 26.75 | 11 | 1.875 K |
| | 4 | Intermediate lower path | 412 | 16 | 6 | 0.938 K |
| | 5 | Modified lower path | 356 | 12.5 | 4 | 0.563 K |

| | | | | | | |
|---|---|---|---|---|---|---|
| Case study 3 | 1 | Straight line path | 252 | 6.72 | 1 | zero K |
| Case study 4 | 1 | Upper path | 252 | 10.72 | 5 | 0.75 K |
| / a | 2 | Modified upper path | 200 | 7.33 | 3* | 0.375 K |
| | 3 | Lower path | 220 | 9.86 | 5 | 0.75 K |
| | 4 | Modified lower path | 184* | 7 | 3* | 0.375 K |
| Case study 4 | 1 | Upper path | 384 | 18.24 | 9 | 1.5 K |
| / b | 2 | Intermediate upper path | 304 | 14.1 | 7 | 1.125 K |
| | 3 | Modified upper path | 278 | 10.25 | 4 | 0.563 K |
| | 4 | Lower path | 764 | 28.4 | 9 | 1.5 K |
| | 5 | 1'st intermediate lower path | 304 | 12.1 | 5 | 0.75 K |
| | 6 | 2'nd intermediate lower path | 280 | 10.46 | 4 | 0.563 K |
| | 7 | Modified lower path | 276* | 9.36 | 3* | 0.375 K |
| Case study5 | 1 | Upper path | 212 | 9.65 | 5 | 0.75 K |
| /a | 2 | Modified upper path | 180 * | 6.8 | 3* | 0.375 K |
| | 3 | Lower path | 220 | 9.87 | 5 | 0.75 K |
| | 4 | Modified lower path | 188 | 7 | 3* | 0.375 K |
| Case study5 | 1 | Upper path | 192 | 9.12 | 5 | 0.75 K |
| /b | 2 | Modified upper path | 160 * | 6.27 | 3* | 0.375 K |
| | 3 | Lower path | 192 | 9.12 | 5 | 0.75 K |
| | 4 | Modified lower path | 160 * | 6.27 | 3* | 0.375 K |
| Case study5 | 1 | Upper path | 284 | 11.57 | 5 | 0.75 K |
| /c | 2 | Modified upper path | 236 | 8.3 | 3 | 0.375 K |
| | 3 | Lower path | 260 | 11 | 5 | 0.75 K |
| | 4 | Intermediate lower path | 224 | 9 | 3 | 0.375 K |
| | 5 | Modified lower path | 200 * | 6.34 | 2 * | 0.1875 K |
| Case study5 | 1 | Lower path | 384 | 17.24 | 8 | 1.3125 K |
| /d | 2 | Modified lower path | 328 * | 11.75 | 4 * | 0.563 K |
| | 3 | Upper path | 512 | 22.65 | 10 | 1.688 K |
| | 4 | Intermediate upper path | 424 | 16.31 | 6 | 0.938 K |
| | 5 | Modified upper path | 356 | 12.5 | 4 * | 0.563 K |
| Case study6 | 1 | Upper path | 396 | 15.56 | 6 | 0.938 K |
| / a | 2 | Intermediate upper path | 312 | 11.32 | 4 | 0.563 K |
| | 3 | Modified upper path | 300 | 9 | 2 * | 0.1875 K |
| | 4 | Lower path | 280 | 10.47 | 4 | 0.563 K |
| | 5 | Modified lower path | 260 * | 8 | 2 * | 0.1875 K |
| Case study6 | 1 | Upper path | 592 | 23.79 | 9 | 1.5 K |
| /b | 2 | Intermediate upper path | 420 | 15.2 | 5 | 0.75 K |
| | 3 | Modified upper path | 396 * | 12.56 | 3 * | 0.375 K |
| | 4 | Lower path | 604 | 24.11 | 9 | 1.5 K |
| | 5 | Modified lower path | 528 | 20.1 | 7 | 1.125 K |

| | | | | | | |
|---|---|---|---|---|---|---|
| Case study7 | 1 | Upper path | 376 | 17 | 8 | 1.3125 K |
| | 2 | Modified upper path | 308 * | 11.2 | 4 | 0.563 K |
| | 3 | Lower path | 429 | 20.44 | 10 | 1.688 K |
| | 4 | Intermediate Lower path | 376 | 13.03 | 4 | 0.563 K |
| | 5 | Modified lower path | 340 | 11.07 | 3 * | 0.375 K |
| Case study8 | 1 | Upper path | 384 | 14.24 | 5 | 0.75 K |
| | 2 | Modified upper path | 296 * | 9.9 | 3 * | 0.375 K |
| | 3 | Lower path | 456 | 16.16 | 5 | 0.75 K |
| | 4 | Modified lower path | 356 | 11.5 | 3 * | 0.375 K |
| Case study9 / small s | 1 | Upper path | 742 | 39.8 | 21 | 3.75 K |
| | 2 | Intermediate upper path | 404 | 20.77 | 11 | 1.875 K |
| | 3 | Modified upper path | 435 * | 14.66 | 5 * | 0.75 K |
| | 4 | Lower path | 720 | 39.2 | 21 | 3.75 K |
| | 5 | Intermediate Lower path | 476 | 22.7 | 11 | 1.875 K |
| | 6 | Modified lower path | 465 | 20.4 | 9 | 1.5 K |
| Case study 9 / large s | 1 | Upper path | 436 | 15.63 | 5 | 0.75 K |
| | 2 | Intermediate upper path | 429 | 13.44 | 3 | 0.375 K |
| | 3 | Modified upper path | 425 * | 12.33 | 2 * | 0.1875 K |
| | 4 | Lower path | 515 | 17.73 | 5 | 0.75 K |
| | 5 | Modified lower path | 483 | 14.88 | 3 | 0375 K |

**Note** : the star sign refers to the optimal selection.

**Table 3.**
**Optimal Path Information for All Cases Studies.**

| **Case study** | **Optimal path** | **Total length (cm)** | **time (sec)** | **No. of segment** | **K.E Joul** | **MLR %** | **MLS %** | **MSC** | **TS** |
|---|---|---|---|---|---|---|---|---|---|
| case study 1 | Modified upper path | 240 | 7.5 | 2 | 0.1875 K | 43 | 67 | 1 | 5 |
| case study 2 | Modified upper path | 276 | 9.36 | 3 | 0.375 K | 56 | 73 | 1 | 5 |
| case study 3 | Straight line path | 252 | 6.72 | 1 | ZERO | 0 | 0 | 0 | 1 |
| case study 4/a | Modified lower path | 184 | 7 | 3 | 0.375 K | 27 | 40 | 1 | 4 |
| case study 4/b | Modified upper path | 272 | 10.25 | 4 | 0.5625K | 65 | 67 | 2 | 7 |
| case study 5/a | Modified upper path | 180 | 6.8 | 3 | 0.375 K | 18 | 40 | 1 | 4 |
| case study 5/b | Modified upper & lower path | 160 | 6.26 | 3 | 0.375 K | 17 | 40 | 1 | 4 |
| case study 5/c | Modified lower path | 200 | 6.34 | 2 | 0.1875 K | 30 | 60 | 2 | 5 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| case study 5/d | Modified upper path | 328 | 11.75 | 4 | 0.5625K | 36 | 60 | 1 | 5 |
| case study 6/a | Modified lower path | 260 | 8 | 2 | 0.1875 K | 35 | 67 | 1 | 5 |
| case study 6/b | Modified upper path | 396 | 12.56 | 3 | 0.375 K | 35 | 67 | 2 | 5 |
| case study 7 | Modified upper path | 308 | 11.2 | 4 | 0.5625K | 28 | 40 | 2 | 5 |
| case study 8 | Modified upper path | 296 | 9.9 | 3 | 0.375 K | 35 | 70 | 1 | 4 |
| case study 9/ small s | Modified upper path | 400 | 14.66 | 5 | 0.75 K | 46 | 76 | 2 | 6 |
| case study 9 / large s | Modified upper path | 425 | 12.33 | 2 | 0.1875 K | 18 | 60 | 2 | 4 |

where

$$K.E_{loss} = \frac{R_a J^2 K_G^2}{K_t^2} \int_0^{t_f} \ddot{\theta}^2 dt$$

and $\frac{R_a J^2 K_G^2}{K_t^2}$

Constant depend on the DC motor parameters (K).

### 12.1. Modification step cycle (MSC)

It is the processing time (during programming) which is required for modifying the path from one form to another according to (OAC) algorithm (for example converting the lower into an intermediate lower path or into a modified lower path directly).

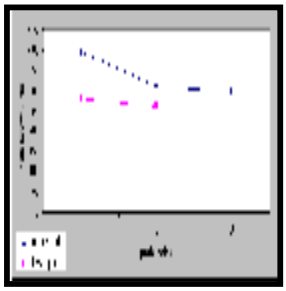
### 12.2. Selecting time (TS)

It is the processing time (during the programming) which is required for selecting the optimal path among all the resulted paths according to (OAC) algorithm.

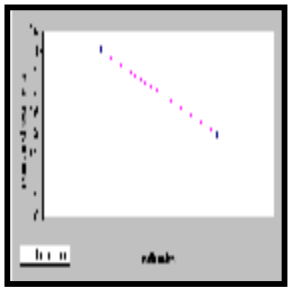
## 13. Length and Number of Segments of Resulted Paths

In all previous case studies, the resulted paths is different from each to other as a result of optimal path criteria which try to modify the upper and lower paths to get the optimal path either with respect to length or number of segments ( as explained in previous chapters).

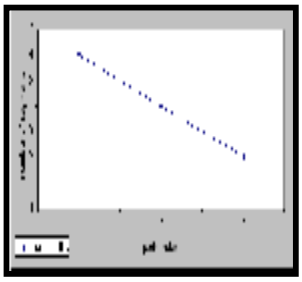
The modification level is different from one case to other depending on number of obstacles, position and distribution of them with respect to start and goal point's positions, size of obstacles …, etc. However the value of this level in all experiments is that the resulted paths are modified with respect to their length and number of segments. Figure (25) show how in case study 5, the length and number of segments of resulted paths are decreased with increasing the index of paths.



**(a) Path Length With Respect to Path Index**



**(b) Path No. of Segments with Respect to Lower Path Index**



**(c) Path no. of Segments with Respect to Path Upper Index.**

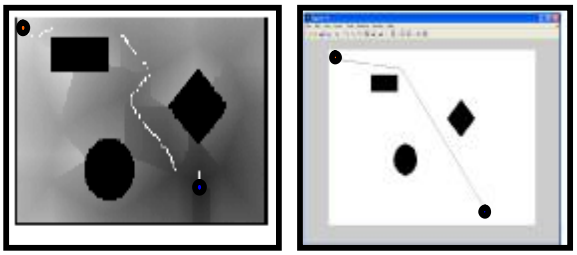
**Fig. 25. Path length and no. of Segments Variation for Case Study 6 .**

## 14. Conclusion

A proposed path planning algorithm which is called (OAC) algorithm is proposed in this work and it is applied to pioneer p3-dx mobile robot to obtain the optimal path ( in time and power ) in 2-D static environment. The building algorithm is based on the observation , analysis of the obstacles that intersects with the supposed straight path that connects the start and goal point and conclude the resulted paths and select the better one among them according to lengths and the number of the individual lines which form them. It is worthy to mention that this algorithm can be applied for dynamic environments by progranning the cammera to provide an instantaneous image for the environment for each spesific time and make the program repeat itself consequently in intervals. This algorithm requires less computations power than most of other algorithms, because the selection of the optimal power is applied maximally among six or seven paths. The optimal path selection is not submitted to known optimization criteria , but it is found by a teqnique that give us the better path which is not only the shortest one but also the less one in consuming power. This with addition to advantages of the (OAC) algorithm gives it enough strength that qualifies it to compete with all other path planning algorithm and excel on some of them.

The advantages of ( OAC) algorithm are :

- The possibility of changing the start and goal points for any values and from any direction. This is considered a constraint for some path planning algorithms which do not have this ability.
- This algorithm can be applied for different shapes of obstacles **by creating new suggestion** which said that " all different shapes of obstacles can be regulated by converting them into square or rectangular shapes only ". This is considered a treatment for some path planning algorithms which is constrainted by the shape of the obstacles.
- The resulted optimal path is the shortest one in length as compared with the other paths which resulted acccording to OAC algorithm (and so in time that the robot required to accomplish this path ) when it compared with some path planning algorithms such as genetic algorithms and the potential field algorithms [3] as shown in figure (26-a).
- The resulted optimal path consummes less the power and it is required to change its direction in the curved parts in the resulted path ( because it is smooth and contains small number of curves ) as compared with potential algorithm as shown in figure (26-a ). This resulted path is considered more economic as shown in figure (26-b).



**(a) The Resulted Path according to Potential Field Algorithm [3]**

**(b) The Resulted Path According to (OAC) Algorithm**

**Fig.26. Comparison Between OAC Path and Potential Field Path**

## 15. References

[1] Steven M. cavalle, 'Planning Algorithm', Cambridge University Press 2006.

[2] Pioneer reference book, www.mobilerobotics.com. 2008.

[3] Adam Hoover, Bent David Olsen,' Path Planning For Mobile Robots Using a Video Camera Network ', in proc. Of IEEE/ASME Int'l Conf. on Advanced Intelligent Mechatronics, 1999.

[4] 'Computer Vision and Image processing ', Scott E. Umbaugh, Prentice Hall Press 1998.

[5] Kamran H. Sedighi, Kaveh Ashenayi, Theodore W. Manikas, Roger L. Wainwright, Heng-Ming Tai. , 'Autonomous Local Path-Planning For a Mobile Robot Using a Genetic Algorithm', Vol.2, 2004, Page(s): 1338 - 1345, IEEE.

[6] Anand Veeraswamy, 'Optimal Path Planning Using an Imported A* Algorithm for Homeland Security Applications ', February 2006, AIA'06: Proceedings of the 24th Lasted international conference on Artificial intelligence and applications.

[7] Andrew N. Hand, Jagruthi Godugu, Kaveh Ashenyi, Theodore W. Manikas, Roger L. Wainwrightb. 'Benchmarking Robot Path Planning ', ANNIE 2005.

[8] Mehmet Serdar Guzel, 'Mobile Robot Navigation Using a Vision Based Approach', 2003.

[9] 'Digital Image Processing' /second edition / Rafael C.Gonzalez / 1987.

[10] Amar Rezoug, Mohand Said Djouadi , 'Visual Based Lane Following for Non-holonomic Mobile Robot' , Vol. III (2008), Suppl. issue: Proceedings of ICCCC 2008, pp. 475-479 , Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 .

[11] Arvind Antonio de Menezes Pereira, 'Navigation And Guidance Of An Autonomous Surface' , Volume 60, Issue 2, March 2008 , on pages 133-143, Computers and Electronics in Agriculture.

[12] Ethar H.khalil, 'Application of Probability Recursive Method for Path Planning ', MSC. Thesis, 2008.

[13] 'Visual C++.NET', Michel Hyman, hungry minds, 2002.

[14] 'Problem Solving With C++'/ Walter Savitch, Addison Wesley / 1999.

[15] 'Visual C++.NET', H.M.Deitel , 2004 ,Prentice hall press.

# السيطرة على المسار المخطط لأنسان آلي متحرك

**نبيل كاظم عبدالصاحب آمنة عبد الهادي صالح**
*قسم هندسة الميكاترونيكس/ كلية هندسة الخوارزمي/ جامعة بغداد*
البريد الالكتروني:dr-nabeelalsahib@hotmal.com

**الخلاصة**

يقدم هذا البحث خوارزمية جديدة لتخطيط المسار المتوقع للروبوت المتحرك نوع p3-Dx pioneer ذو التحكم الذاتي القائم على تقنيات معالجة الصور باستخدام الكاميرا اللاسلكية التي توفر الصورة المطلوبة لبيئة مجهولة. يتم تطبيق الخوارزمية المقترحة على هذه الصورة للحصول على المسار الأمثل و الافضل للروبوت. وهو يقوم على رصد وتحليل العقبات التي تقع في الطريق المستقيم بين نقطة البداية والهدف من خلال الكشف عن هذه العقبات ، وتحليل ودراسة أشكالها ومواقعها ونقاط التقاطع مع الطريق المستقيم المفترض ، ومحاولة العثور على المسار الافضل الذي يربط بين نقطة البداية والهدف. ان تمثيل النظام العام يتضمن : (كاميرا لاسلكية - الكمبيوتر - اتصال لاسلكي للروبوت المتحرك ).

هذا العمل تضمن جزئين اساسيين, الجزء الاول ويشمل بناء برنامج بلغة (MATLAB) الذي يطبق على صورة بيئة لإيجاد الطريق الأمثل تقريبا وذلك من خلال التعشيق بين اللغتين البرمجيتين( C++.NET) - MATLAB ويتم تزويد معلومات المسار الى البرنامج الذي تم بناؤه بلغة( Visual C++.Net) الخاصة ببرمجة الروبوت (P3 - dx pioneer) لتحقيق المسار المطلوب.

اما الجزءالثاني يتمثل باستخدام كاميرا لاسلكية تأخذ صورة للبيئة الواقعية ثم يرسلها إلى جهاز الكمبيوتر الذي يعالج ، في دوره ، هذه الصورة ويرسل (عن طريق الاتصال اللاسلكي) معلومات المسار إلى الروبوت الذي تم برمجته بلغةVisual C++.Net ) لكي يسير وفقا لهذا المسار.

النتائج العملية بينت أن الروبوت (P3 - dx pioneer) تمكن من الحركة بنجاح من نقطة البداية و الوصول إلى الهدف عبر المسار الأمثل و الذي تم الحصول عليه كنتيجة من الخوارزمية التي تم بناؤها هنا , حيث تم ايجاد مسارين هما المسار العلوي والمسار السفلي ثم وفقا لهذه الخوارزمية تم تحديثهما لينتج منهما المسار العلوي الوسيط و المسار السفلي الوسيط ومن ثم الحصول على المسار العلوي المحسن و المسار السفلي المحسن , ومن خلال فحص هذين المسارين النهائيين نسبة للوقت و الطاقة يتم اختيار المسار الامثل للروبوت.