Journal of Computer Science and Cybernetics, V.24, N.4 (2008), 358–373

DYNAMIC PATH PLANNING FOR A MOBILE ROBOT USING IMAGE PROCESSING

NGUYEN TRUONG THANH[1], NITIN V. AFZULPURKAR[2]

[1]*Space Technology Institute, Vietnam Academy of Science and Technology*
[2]*Asian Institute of Technology, Bangkok, Thailand*

Abstract. The aim of this work is to develop a path planning algorithm for a mobile robot in a dynamic environment. The algorithm uses visibility graph method with A* search for global path planning. On the other hand, the algorithm uses a new potential field method with predictive collision ability for local path planning. The image processing algorithm is implemented to get the information of environment. A user-friendly path planning software, which is called PMADE v1.0, is developed. PMADE v1.0 is utilized for image processing of robot environment, path planning in a dynamic environment, simulation of robot navigation, and data export to real robot.

Tóm tắt. Mục đích của nghiên cứu là phát triển một thuật toán lập đường đi cho robot di động trong một môi trường động. Thuật toán sử dụng phương pháp đồ thị trực quan (visibility graph) với giải thuật tìm kiếm A* để lập đường đi toàn cục. Bên cách đó, thuật toán sẽ sử dụng phương pháp trường thế năng (potential field) mới với khả năng dự báo va chạm để lập đường đi địa phương. Thuật toán xử lý ảnh được thực thi để thu nhận thông tin về môi trường hoạt động của robot. Phần mềm lập đường đi có tên là PMADE v1.0 đã được phát triển. PMADE v1.0 dùng cho xử lý ảnh về môi trường hoạt động robot, lập đường đi trong môi trường động, mô phỏng quá trình di chuyển của robot và truyền dữ liệu điều khiển robot thực.

1. INTRODUCTION

The popularity of autonomous mobile robots has been rapidly increasing because of their new emerging application areas, from room cleaning, material transportation systems to space explorations. However, the development of a satisfactory path planning algorithm that will enable the mobile robots to navigate safely in dynamic environments is still an open research problem.

In an indoor environment such as an office, a hospital or a manufacturing plant, the situation of work space for mobile robot changes dynamically with moving obstacles. To develop an path planning for robot on this, some limitations are defined. Workspace of mobile robot is a flat and 2-dimension environment. Obstacles are simple polygon shape and single color to be identified easily by cameras. The non-stationary obstacles move along linear path with constant velocities. For map building, only an overhead camera above the center of the workspace is used to observe the whole environment surface. For future, we can combine more cameras in larger environments. No kinematic and dynamic constraints limit the motion of the robot.

In this paper, a path planning algorithm for mobile robot in a dynamic environment by visibility graph and potential field methods is proposed. In the next section, a software packet called PMADE is developed to implement the algorithm by image processing for getting the environment information. Finally, some experiments and results are given and discussed.

2. THE PROPOSED PATH PLANNING ALGORITHM

For navigation of mobile robot in dynamic environments, the path planning problem is divided hierarchically into two problems: global static path planning (1) and local dynamic path planning (2). When the environment is viewed globally, small environment changes can be neglected. So, only stationary obstacles are taken into consideration, and the shortest path to the final goal will be planned. In this research, "Roadmaps method" will be applied to the global path planning. To the contrary, when a local view of the environment is taken, even a small change should be considered to avoid collisions. Therefore, the local dynamic path plan deals with not only stationary obstacles but also moving obstacles in detail. "Potential method" will be applied to the local path planning [1].

This approach is developed because it is suitable for path planning in dynamic indoor environment. On the other hand, it is simple to implement by simplistic mathematics formulas. This approach can also be used for real-time control.

2.1. Global static path planning

In the first level of the path planning, the sub-goals are setup along the shortest path to the final goal in consideration of only stationary obstacles in the map. First, the shortest path is searched based on the "Visibility graph method" under the condition that assures no collision with stationary obstacles. In the "Visibility graph method", all vertexes of stationary obstacles are extracted for maps, and connected with each other as the path. As the path graph shows all possible routs to the goal, we can search the shortest path by A^* searching method. The vertexes on the shortest path are defined as the sub-goals.

This global path planning method consists of the following three planning steps. The planned results are handed to the second level planning, or local dynamic path planning.

2.1.1. Setting sub-goal candidate points

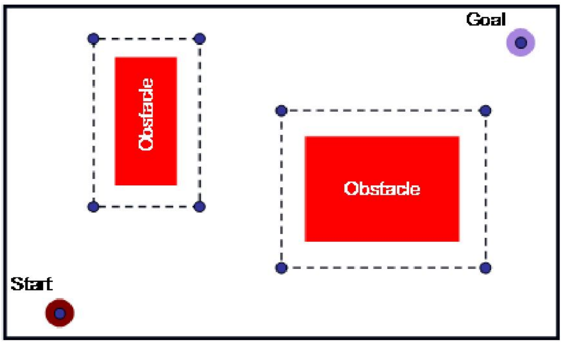


Fig. 1. Setting sub-goal candidate point

The dangerous areas, where the robot might collide with obstacles, are set around each obstacle based on the map's information. All the vertexes of the area are sub-goal candidate points where the robot should pass by before the final goal (Fig. 1). The dangerous areas are defined as the area, where the distance from the circumference of each obstacle is within a certain length. The length should be long enough and do not collide with the obstacles; i.e., with the radius of the robot body.

2.1.2. Drawing up a visibility graph

A path network, whose starting point is the robot start point and whose finishing point is the goal of the robot, is drawn up by connecting sub-goal candidate points using Visibility graph method. This method allows drawing a line connecting two points, when the line crosses neither obstacles nor the dangerous areas.

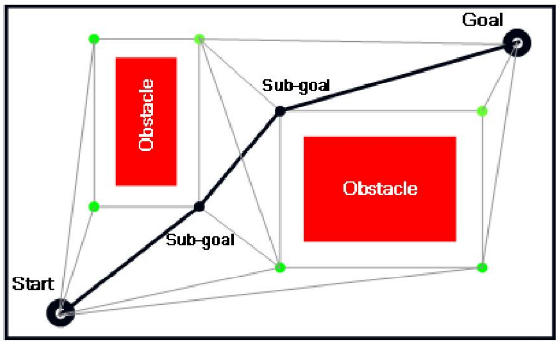


Fig. 2. Visibility graph

2.1.3. Searching the shortest graph

The obtained visibility graph shows all possible paths from the starting point to the goal. The shortest path to the goal is searched using A^* searching method. The sub-goal candidate points on the obtained shortest path are defined as the sub-goals towards the final goal. An example is shown in Fig. 2. In this figure, points $G1, G2$ are selected as sub-goals. The algorithm for global path planning is shown as follows.

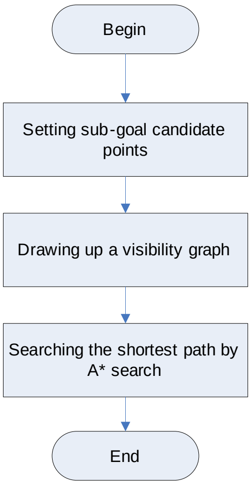

Begin
Setting sub-goal candidate points
Drawing up a visibility graph
Searching the shortest path by A* search
End

Fig. 3. Global path planning algorithm

2.2. Local dynamic path planning

At the second level of the planning, the local dynamic path planning makes a practical path to the next sub-goal. For this planning, we apply the Potential method to avoid known obstacles in the map information. When the Potential field method is applied to the local dynamic path planning, we need the distance between the robot and the obstacle. The information is gotten through the environment interfaces such as a CCD camera and distance sensors. Since this method uses only distances between the robot and the obstacles, it requires small amount of calculation. Therefore, the method is suitable for real-time robot control. However, this method sometime causes parallel run phenomena and stationary dead-lock phenomena (local minima). The parallel run phenomenon is an undesired situation where the robot and the other moving obstacle run in parallel under the influence of repulsive potential.

A new potential, or prediction potential, is introduced to decrease the undesired phenomena mentioned above. If the robot seems to pass the crossing point earlier than the obstacle, the prediction potential becomes attractive and vice versa. This prediction potential prevents the two moving things to reach the crossing point at the same time.

The algorithms for making the three potentials will be shown below in detail; they are the attractive sub-goal potential, the repulsive obstacle potential and the attractive/repulsive prediction potential.

2.2.1. Searching the shortest graph

The attractive sub-goal potential around the current sub-goal is defined. This potential pulls the robot toward the current sub-goals. The attractive sub-goal potential can be simply expressed as [1].

$$U_a = \frac{1}{2}k_a\|q - q_{sg}\|^2, \tag{1}$$

where, $q = (x, y)$: current Euclidean coordinate of robot; qsg: Euclidean coordinate of sub-goal; k_a: an attraction constant value.

Thus, the attractive force is will be

$$F_a = -\nabla U_a = -k_a(q - q_{sg}). \tag{2}$$

The typical three-dimensional shape of the potential field is shown in Fig. 4. The robot will be pulled as if it falls into the current sub-goal.

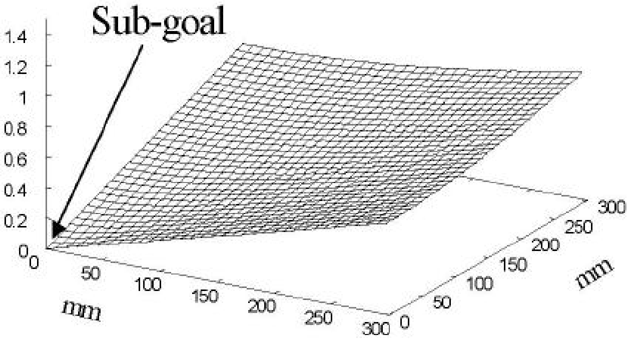


Fig. 4. Attractive sub-goal potential

2.2.2. Repulsive obstacle potential

The repulsive potential field around obstacles is defined. When the robot comes close to obstacles, the repulsive potential works as a force to make the robot away from the obstacles. The repulsive potential can be expressed as follows

$$U_r = \begin{cases} \frac{1}{2}k_r(\frac{1}{x_p} - \frac{1}{x_s})^2, & x_p \le x_s \\ 0 & x_p > x_s \end{cases} \tag{3}$$

where, k_r: a repulsion constant value; x_p: the shortest distance between the robot current point and the obstacle; x_s: a constant value and is defined as a safe distance around the obstacle.

The repulsive potential force will be $F_r = -\nabla U_r$. Thus, the magnitude of this force is defined by the following equation.

$$|F_r| = \begin{cases} k_r(\frac{1}{x_p} - \frac{1}{x_s})\frac{1}{x_p^2}, & x_p \le x_s, \\ 0 & x_p > x_s \end{cases}. \tag{4}$$

An example of the potential field is shown in Fig. 5. The parameter, that decides the power of the potential, is constant, and the value is decided based on the characteristics of the obstacles.

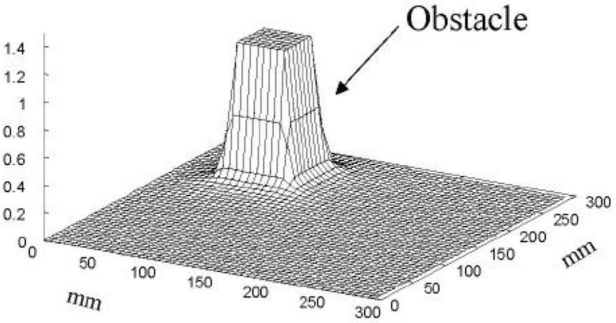


Fig. 5. Repulsive obstacle potential

2.2.3. Attractive/repulsive prediction potential

This potential field is formed around the crossing point of the robot path and the obstacle path to avoid simultaneous approach to the point. If the robot is predicted to pass the crossing point earlier than the obstacle, the predictive potential is attractive, and vice versa. If the situation is opposite, the potential is repulsive. When the approaching times of the both objects are almost the same, the power of the potential should be strong to strength the acceleration or the deceleration. Besides, the closer the robot is to the crossing point, the stronger the predictive potential becomes. Under the assumption of uniform linear movement, the predicted arriving point of the obstacle is predicted when the robot reaches the crossing point (Fig. 6).

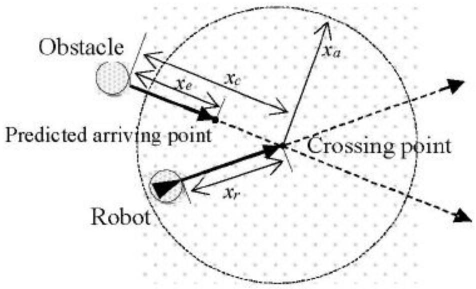
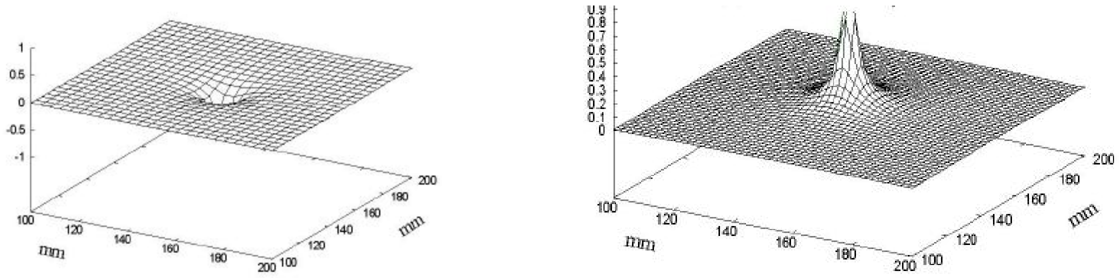


Fig. 6. Field of Predictive potential

The magnitude of the prediction potential force Fe is calculated by Equation 5 using the above notations [2].

$$|F_e| = \begin{cases} k_e \dfrac{1}{x_e - x_c}(\dfrac{1}{x_r} - \dfrac{1}{x_a}), & x_c \leq x_a,\ x_e \neq x_c \\ 0 & x_c > x_a \end{cases}. \tag{5}$$

An example of the potential field is shown in Fig. 7. If we use only traditional potential field method, the robot might move inefficiently because of the parallel run or stationary dead-lock phenomenon (local minima). The value of parameter is decided based on the characteristics of the obstacles.



(a) Attractive potential (b) Repulsive potential

Fig. 7. Attractive/repulsive prediction potential

2.2.4. Determining constant values for potential fields

In our work, the concept for calculation of potential field constant values is based on the maximum potential forces and the priority of potential forces. The maximum potential forces can be estimated by the limitation of robot velocity because the robot velocity is proportional to the synthesized potential force. The priority of each potential force depends on the potential field.

Due to the importance of attractive and repulsive potential fields, their priorities are higher than the priority of the prediction potential field. Thus, it is necessary to set the maximum attractive and repulsive potential force is bigger than the maximum prediction potential force.

2.2.5. Synthesized potential

We have introduced three kinds of potentials, and the robot is controlled based on the summation field of the three as shown in Equation 6. An example of the synthesized potential field is shown in Fig. 8.

$$U(q) = U_a(q) + U_r(q) + U_e(q), \tag{6}$$

where, $q = (x, y)$ is the coordinate of robot in the environment.

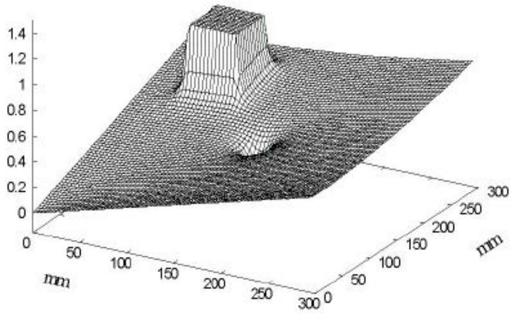


Fig. 8. Synthesized potential

Then, we have the control force for the robot generated by the synthesized potential field as follows:

$$F(q) = F_a(q) + F_r(q) + F_e(q). \tag{7}$$

Finally, to control the mobile robot, we set the robot speed (vx, vy) proportional to the force $F(q)$ generated by the field. The local path planning algorithm is shown in Fig. 9.

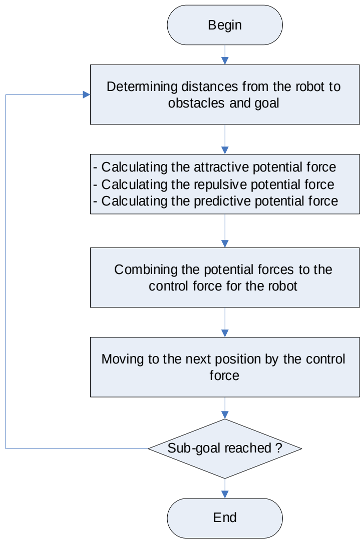

Begin
Determining distances from the robot to obstacles and goal
- Calculating the attractive potential force
- Calculating the repulsive potential force
- Calculating the predictive potential force
Combining the potential forces to the control force for the robot
Moving to the next position by the control force
Sub-goal reached ?
End

Fig. 9. Local path planning algorithm

3.3. Path planning algorithm in dynamic environments

In a dynamic environment, we combine the global static path planning using Visibility graph and the local dynamic path planning using Potential field to make the collision-free path for mobile robot. The robot moves via sub-goal on the global static path. If the distance between the robot and the moving obstacle (d_{RM}) is more than the threshold distance (d_T), the robot continues to move along the global path. The threshold distance is a safe distance that the moving obstacle does not endanger for the mobile robot. Otherwise, the new local collision-free path for mobile robot will created using Potential field method. The robot will move along local path to sub-goal. Then, the robot comes back to move in the global path. The movement of the robot is finished when the robot comes to the goal. The path planning algorithm is shown in Fig. 10.

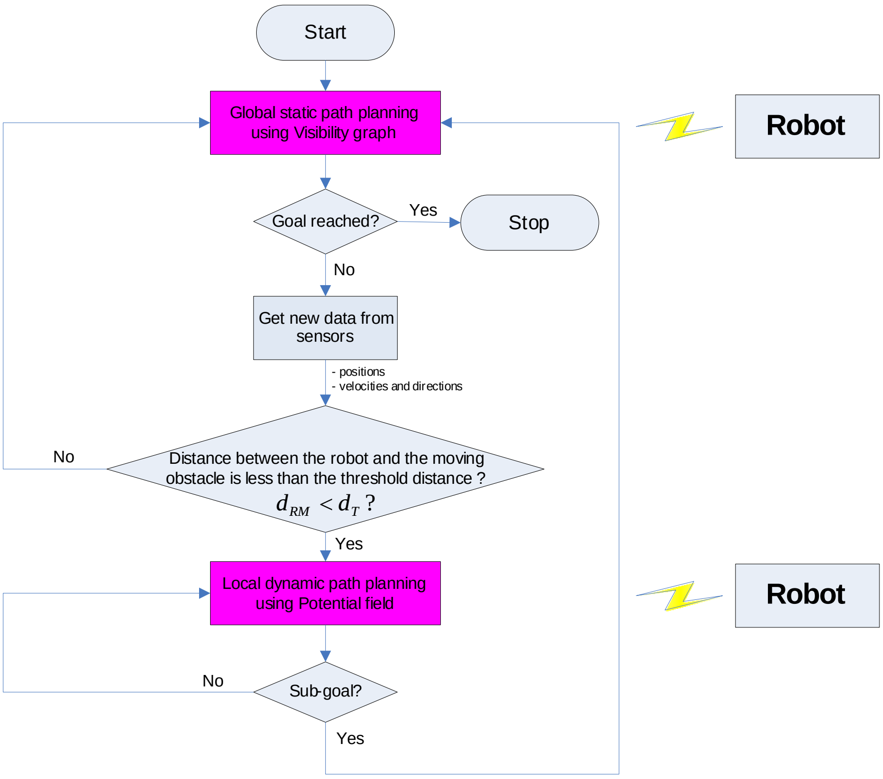

Start
Global static path planning using Visibility graph
Robot
Goal reached?
Yes
Stop
No
Get new data from sensors
- positions
- velocities and directions
No
Distance between the robot and the moving obstacle is less than the threshold distance ?
$d_{RM} < d_T$?
Yes
Local dynamic path planning using Potential field
Robot
No
Sub-goal?
Yes

Fig. 10. Flowchart of path planning in dynamic environment

3. SYSTEM SETUP

The idea of this work is shown in the experiment setup, which is specially designed to be similar to mobile robots operating in manufacturing industry as shown in Fig. 11. It is also a very useful setup to test the path planning algorithms of mobile robot, by just plugging in the algorithm to be tested into the software developed.

The block diagram of the system is described in Fig. 12. First, the overhead camera gets continuously information about mobile robot, obstacles and dynamic environment. Then, the data is transmitted to PC to build map, to locate mobile robot and obstacle positions. Next, the software will analyze the data, design the path planning algorithm, and make decisions to control robot via wireless LAN. After that, the mobile robot controller receives control signals via wireless LAN and then implements the task. The cycle is back to first step.

Due to the time limit, our work focuses only on image processing, path planning, and user interface modules. The purpose is to develop a software package to get information from image processing, to implement the path planning algorithm in a dynamic environment, to simulate the robot navigation in 3D virtual environment, and to interact friendly with users for robot control.

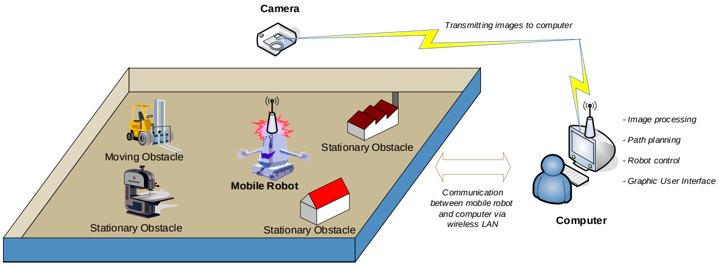


Fig. 11. The experiment model

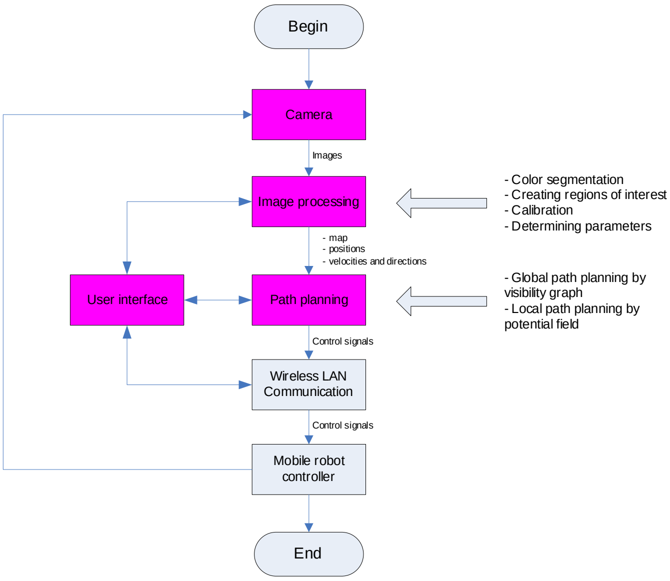


Fig. 12. Block diagram of the system

3.1. Image processing

3.1.1. Hardware

The camera using in this work is Sonys DFW-SX910 camera (Fig. 13). It allows for the clear capture of fast moving objects or still images in low light environments for applications such as machine vision, semiconductor inspection, image processing, print inspection, microscopy, and factory automation.

Fig. 13. Sonys DFW-SX910 CCD camera

These features of the camera are suitable for robot control in dynamic environments, where have moving objects and changeable light conditions. The camera is also useful for real time path planning by fast IEEE 1394 interface.

The camera is placed in the overhead off-board position, 3m directly above the center of the workspace. The optical axis of the camera is perpendicular to the workspace of the robot. The robot workspace is an area of 3200 mm×2400 mm. The camera is connected to a computer (laptop) via IEE 1394 interface. The robot can keep track of its position and those of the obstacle and the targets at all times, regardless of the relative position of these with respect to the robot [3].

3.1.2. Algorithm

The image processing is done using Intel Open Computer Vision (OpenCV) library [4]. OpenCV is a cross-platform middle-to-high level Application Programming Interface (API). It is free for commercial and noncommercial use.

In this work, the moving obstacles and the static obstacles are of different single colors. The image processing module must identify the following parameters in real time from each image captured:

+ The location of any obstacle (s) present.

+ The direction of moving obstacle(s).

+ The velocity of moving obstacle(s).

+ The size of an area fully encompassing each obstacle.

These parameters have to be converted from pixel coordinates to the physical coordinates.

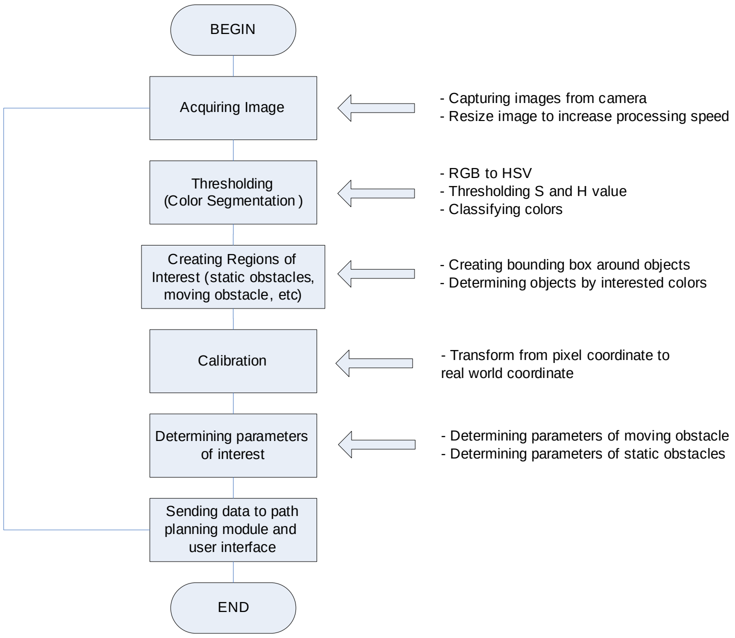

BEGIN
Acquiring Image
- Capturing images from camera
- Resize image to increase processing speed
Thresholding (Color Segmentation)
- RGB to HSV
- Thresholding S and H value
- Classifying colors
Creating Regions of Interest (static obstacles, moving obstacle, etc)
- Creating bounding box around objects
- Determining objects by interested colors
Calibration
- Transform from pixel coordinate to real world coordinate
Determining parameters of interest
- Determining parameters of moving obstacle
- Determining parameters of static obstacles
Sending data to path planning module and user interface
END

Fig. 14. The flowchart of the image processing module

The algorithm of the image processing module is described in Fig. 14. The first step of the module is to acquire images by an overhead CCD camera sensor. The images are sent continuously from the camera to computer via IEEE 1394 interface. The second step is to perform a color threshold to identify static obstacles, moving obstacles and background. The images from camera are converted from RGB to HSV. The result is an image consisting of binary pixels. The third step is to create region of interest containing obstacles by removing noises and fill true regions. The fourth step is to correct data from pixel coordinates to real world coordinates. The fifth step is to determine parameters including locations, directions, and sizes. The last step is to send all resulting data to path planning module and user interface. The results of image processing are shown in Fig. 15.

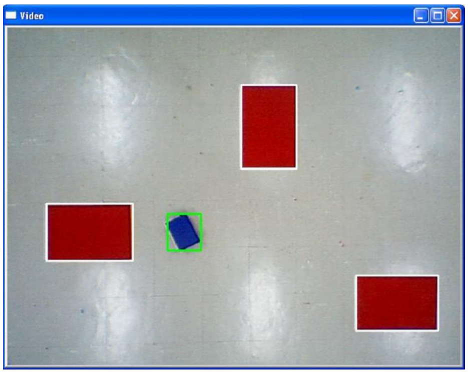
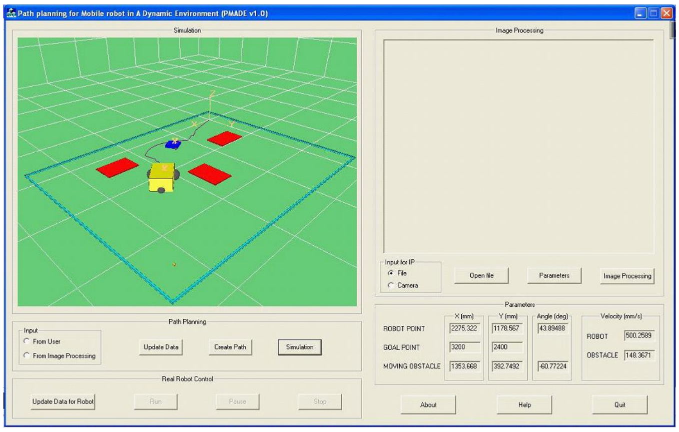


Fig. 15. The image is created regions of interest

3.2. PMADE v1.0

PMADE v1.0 (Path planning for Mobile robot in A Dynamic Environment) is a software packet which is developed to create the path for mobile robot by image processing, to simulate the robot navigation in 3D virtual environment, and to export data files for robot control. Further, PMADE will be developed to control real mobile robots from path planning in a dynamic environment.



	X (mm)	Y (mm)	Angle (deg)
ROBOT POINT	2275.322	1178.567	43.89488
GOAL POINT	3200	2400	
MOVING OBSTACLE	1353.668	392.7492	-60.77224

	Velocity (mm/s)
ROBOT	500.2589
OBSTACLE	148.3671

Fig. 16. The interface of PMADE v1.0

In the version 1.0 as shown in Fig. 16, PMADE is designed for Pioneer 2-DX mobile robot. The environment is an indoor environment with some static obstacles and a moving obstacle. The static obstacles are made by red rectangular papers and fixed on the white floor. The moving obstacle is a toy car covered by blue paper. The moving obstacle moves linearly with a constant velocity. From the overview camera, PMADE v1.0 can detect all obstacles easily and continuously. The operation of PMADE v1.0 is detailed in Fig. 17.

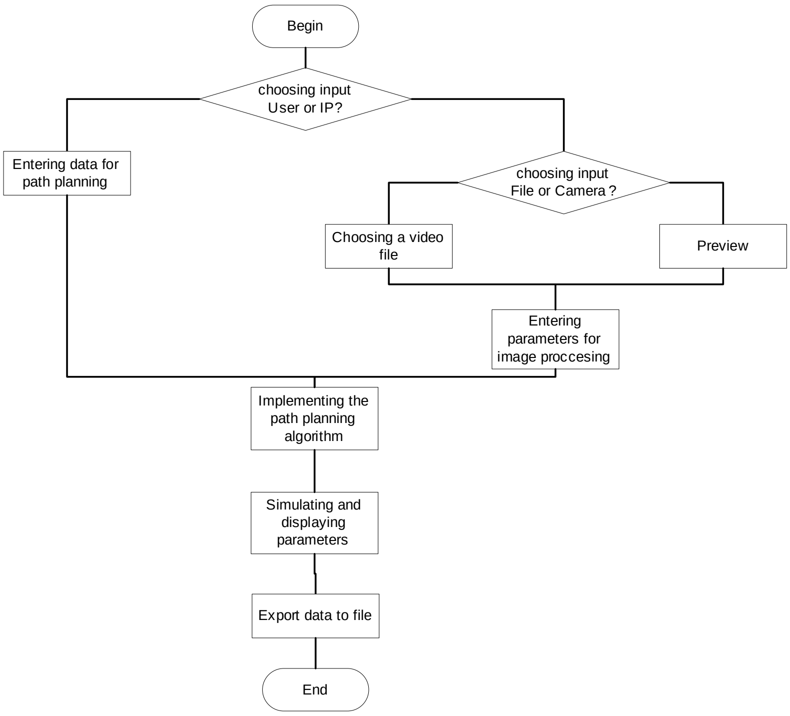
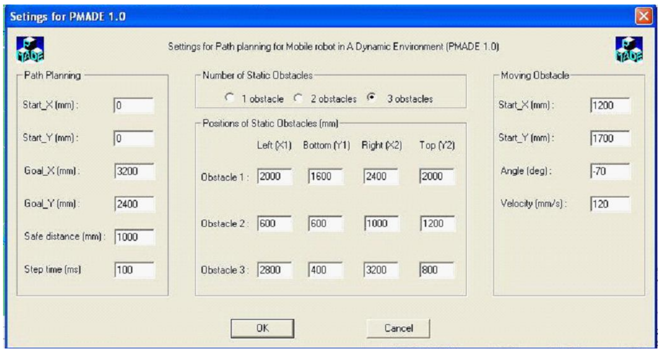

Begin
choosing input User or IP?
Entering data for path planning
choosing input File or Camera ?
Choosing a video file
Preview
Entering parameters for image proccesing
Implementing the path planning algorithm
Simulating and displaying parameters
Export data to file
End

Fig. 17. The operation flowchart of PMADE v1.0

3.2.1. Input from user

When the input from user is chosen, PMADE v1.0 allows users to design different environments for path planning. It is implemented by settings dialog as shown in Fig. 18.


Setings for PMADE 1.0

Settings for Path planning for Mobile robot in A Dynamic Environment (PMADE 1.0)

Path Planning
Start_X (mm): 0
Start_Y (mm): 0
Goal_X (mm): 3200
Goal_Y (mm): 2400
Safe distance (mm): 1000
Step time (ms): 100

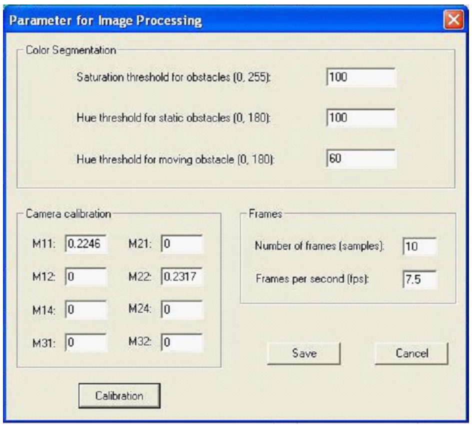
Number of Static Obstacles
○ 1 obstacle ○ 2 obstacles ● 3 obstacles

Positions of Static Obstacles (mm)

	Left (X1)	Bottom (Y1)	Right (X2)	Top (Y2)
Obstacle 1:	2000	1600	2400	2000
Obstacle 2:	600	600	1000	1200
Obstacle 3:	2800	400	3200	800

Moving Obstacle
Start_X (mm): 1200
Start_Y (mm): 1700
Angle (deg): -70
Velocity (mm/s): 120

Fig. 19. Parameter dialog for image processing


Parameter for Image Processing

Color Segmentation
Saturation threshold for obstacles (0, 255): 100
Hue threshold for static obstacles (0, 180): 100
Hue threshold for moving obstacle (0, 180): 60

Camera calibration
M11: 0.2246 M21: 0
M12: 0 M22: 0.2317
M14: 0 M24: 0
M31: 0 M32: 0

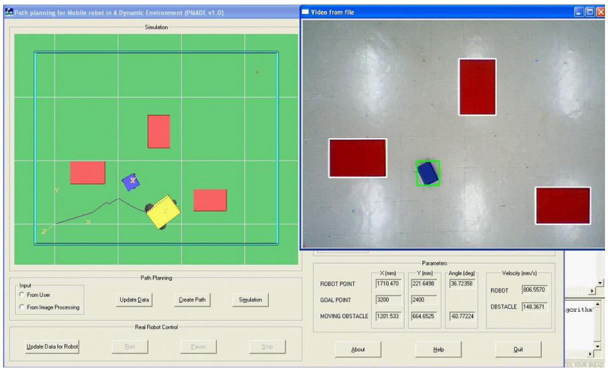
Frames
Number of frames (samples): 10
Frames per second (fps): 7.5

Fig. 18. User Settings of PMADE 1.0 for path planning

3.2.2. Input from image processing

When the input from image processing is chosen, users can select one of two types of image processing input: video file or camera. The images are processed to get the information of environment for path planning. The image processing algorithm was detailed in the previous part.

The parameters for image processing can be modified by the parameter dialog as shown in Fig. 19. After having the parameters, the image processing is done to get the information of environment. Then, the path planning algorithm is implemented. Finally, the simulation is run at the same time with the video from file or camera (see Fig. 20).



	X (mm)	Y (mm)	Angle (deg)
ROBOT POINT	1710.470	221.6498	36.72358
GOAL POINT	3200	2400	
MOVING OBSTACLE	1201.533	664.6525	-60.77224

Velocity (mm/s)	
ROBOT	806.5570
OBSTACLE	148.3671

Fig. 20. The Image processing result

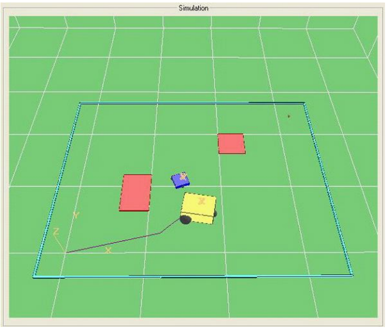


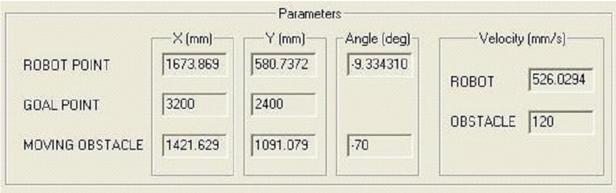
Fig. 21. The 3D simulation window

3.2.3. Outputs

The outputs of PMADE v1.0 consist of the animation from simulation, the parameters of interest displayed in numeric form, and position data file for robot control.

The simulation of path planning is displayed in 3D animation by OpenGL (Fig. 21). It allows users to see the virtual robot navigation created by the path planning algorithm. Users can change view and zoom to look at the simulation process more clearly.

At the same time with the animation, parameters of interest are displayed in numeric form (Fig. 22). The positions and velocities of robot and moving obstacle are continuously shown. It allows user to track the movement process of mobile robot and moving obstacle. These parameters can be used for real robot control.

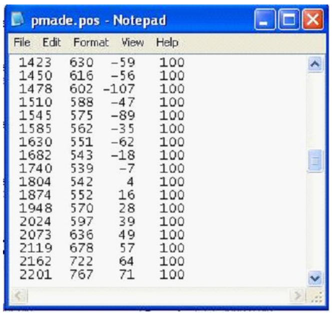


	X (mm)	Y (mm)	Angle (deg)
ROBOT POINT	1673.869	580.7372	-9.334310
GOAL POINT	3200	2400	
MOVING OBSTACLE	1421.629	1091.079	-70

Velocity (mm/s)	
ROBOT	526.0294
OBSTACLE	120

Fig. 22. Simulation parameters of interest

Finally, users can export the path planning data to a text file (Fig. 23). This file can be used for real control robot. The form of this file is designed according to the standard of position file for Pioneer 2-DX. A line of file includes 4 values: X coordinate, Y coordinate, angle, and step time. This file also can be the input for the software by Tang Quoc Nam [6], which is a control program for Pioneer 2-DX at AIT Mechatronics Lab.



```
1423  630   -59   100
1450  616   -56   100
1478  602  -107   100
1510  588   -47   100
1545  575   -89   100
1585  562   -35   100
1630  551   -62   100
1682  543   -18   100
1740  539    -7   100
1804  542     4   100
1874  552    16   100
1948  570    28   100
2024  597    39   100
2073  636    49   100
2119  678    57   100
2162  722    64   100
2201  767    71   100
```

Fig. 23. An output file of path planning

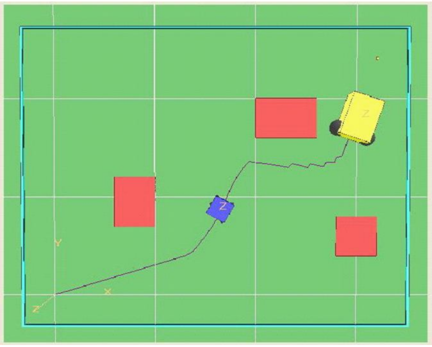


Fig. 24. Path planning by user input with three static obstacles

4. EXPERIMENTS

PMADE v1.0 is implemented some experiments using two input types: user data and image processing. These experiments are implemented with different environments (number of static obstacles, position and size of obstacle, velocity of moving obstacle, etc). To simplify, the start point of robot is $(0, 0)$ and the goal point is $(3200, 2400)$. The safe distance for path planning (used to start local path planning) is 1000 mm. The step time is 100 ms. These parameters can be modified by using the settings dialog.

An example by user input with three static obstacles is shown in Fig. 24. The moving obstacle in this case starts at (3000, 100) with 150 mm/s velocity and 150o angle.

Next, a path planning by image processing input with two static obstacles is shown in Fig. 25. The images are captured from a vide file. The moving obstacle has the velocity of 147 mm/s and the angle of -86.5^0. In the figure, all obstacles are detected exactly. The animation and the movie are run at the same time.

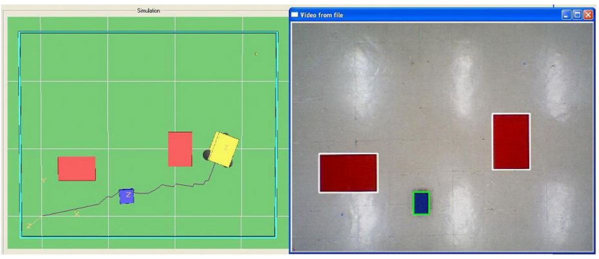


Fig. 25. Path planning by image processing input with two static obstacles

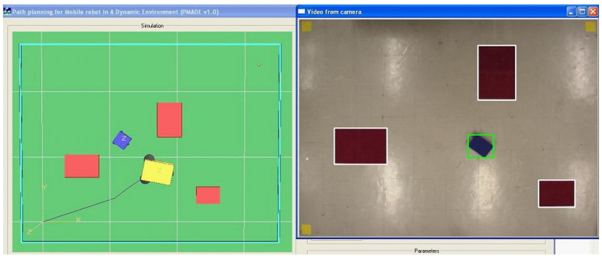

Path planning for Mobile robot in A Dynamic Environment (PMADE v1.0)
Simulation
Video from camera
Parameters

Fig. 26. Path planning by image processing input with three obstacles

Finally, another path planning by image processing input with three static obstacles is shown in Fig. 26. The images are acquired directly by camera. The moving obstacle in this case has the velocity of 121 mm/s and the angle of -34.3^0. All obstacles are detected exactly but the simulation is slower than the movie because of the image processing time.

After some experiments, results are analyzed and estimated. In most of experiment, PMADE v1.0 can generate successful paths for obstacle avoidance in a dynamic environment. The path is relatively smooth. In a few cases, the robot has a collision with the moving obstacle because the path is locked or the moving robot travels too fast. The problem can be solved by modifying the constant coefficients of potential field functions. It needs time to implement more experiments to choosing more suitable constant coefficients. The maximum moving obstacle velocity, which the software can be determined, depends on the limitation of camera and the ability of computer. In our experiment, with PMADE v1.0 on Pentium M 1.6 GHz laptop, the maximum velocity of moving obstacle is around 300 mm/s. About the cycle time for software, if the step time is 100 ms, it takes about 60 ms to process images and to calculate the path planning algorithm. Thus, to synchronize the animation and the movie, the real step time for simulation in PMADE v1.0 is 160 ms.

5. CONCLUSIONS

The proposed path planning algorithm can produce efficient paths, because the path can be searched for the local optimal path without forgetting the final goal. Even if the robot falls into the problem of local minima by applying potential method, the global path is planned again to escape from the stationary points. The method can also decrease the parallel run phenomena by applying a prediction potential.

A software packet called PMADE v1.0 with user-friendly interface for path planning of mobile robot in a dynamic environment is developed. PMADE v1.0 can be used for image processing of robot environment, path planning in a dynamic environment, simulation of robot navigation, and data export for real robot. In this work, PMADE v1.0 is designed for Pioneer 2-DX but it can be developed for different mobile robots. Some experiments with both user and image processing inputs for PMADE v1.0 are implemented with successful paths for mobile robot navigation.

REFERENCES

1. J. Hasegawa, K. Kurihara, N. Nishiuchi, Collision-free path planning method for mobile robot, *IEEE International Conference on Systems, Man and Cybernetics* **3** (2002).
2. K. Kurihara, N. Nishiuchi, J. Hasegawa, K. Masuda, Mobile robots path planning method with the existence of moving obstacles, *10th IEEE Conference on Emerging Technologies and Factory Automation* **1** (2005) 195–202.
3. E. Kruse and F. M. Wahl, Camera-based observation of obstacle motions to derive statistical data for mobile robot motion planning, *Proceedings of the IEEE International Conference on Robotics & Automation*, (1998) pp.662-667. NOI DAU?
4. Intel Corporation, Open Source Computer Vision Library Reference Manual, 2001.

5. CMU 1394 Digital Camera Driver by from Carnegie Mellon University, The Robotics Institute, http://www.cs.cmu.edu/ĩwan/1394/.

6. Tang Quoc Nam, "Development of a localization algorithm for mobile robot in material transportation system", AIT Thesis, ISE 2002.

Received on February 19 - 2009