International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-3, Issue-5)

Research Article **May 2014**

An Image Based Path Planning Using A – Star Algorithm

Heramb Nandkishor Joshi, Prof J. P. Shinde
Dept. Of Electronics and telecommunications
STES's Sinhgad Academy of Engg, Pune, India

Abstract—

The aim of this paper is to plan a path for autonomous robot, based on image processing techniques in the unknown environment. The proposed system finds and analyses an optimal path for a robots, while avoiding obstacles along the way. The environment is first captured as an image using a camera. Obstacles detecting methods are then performed to identify the existence of obstacles within the unknown environment. In this paper we focus on the application of space exploration where Rocks and craters are recognized as obstacles and then shortest path is obtain by A-Star algorithm.

Keywords — Obstacle detection, Path Planning, Shortest path, A-star algorithm.

I. INTRODUCTION

Sojourner was the Mars Pathfinder robot landed on July 4, 1997, the goal was to send a robot to explore the terrain and gather data. Sojourner executed detailed command sequences determined by human controllers and uploaded each command cycle, two or three time per sol (The term sol is used by planetary astronomers to refer to the duration of a solar day on Mars). Sojourner's maximum speed was 0.01m/s and it traveled several hundred meters in the course of its 83 sol mission [1]. The cost of space mission is decided by energy and time is consumed by the robot. And this was in higher side in case of Sojourner robot.

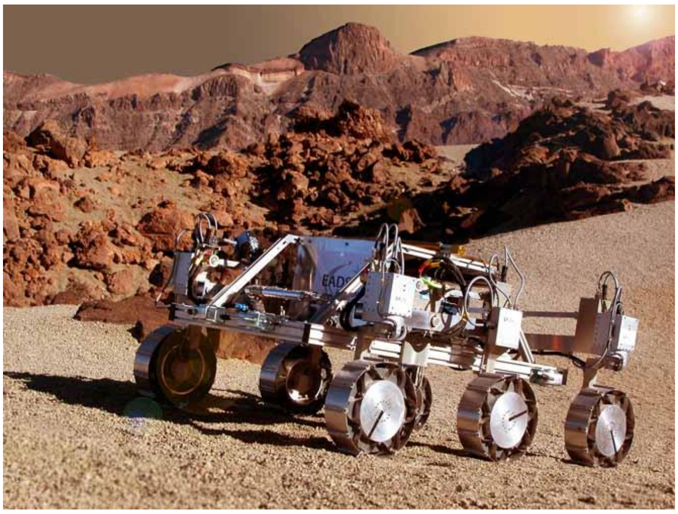

Fig. 1. Robot in unknown environment.

The path planning method describe in this paper aims to calculate shortest path for a robots, while avoiding obstacles along the way. In field of space exploration, autonomous robots are generally based on navigating from a starting point to target point in unknown environment. In these kinds of applications the points that should be visited are unknown. But, for an autonomous robot, consuming less energy and time is very important. To meet these constraints, a shorter path is preferred rather than a longer path. Therefore an intelligent path planning algorithm is always required [2].

Path planning for robots is a complex task. In order to find the shortest path between starting point to goal point in the environment, it is usually needed a map to find shortest the path and follow it. It would be more useful solution for which maps are not available from source to destinations particularly in unknown environment like mars surface [3]. Even more, it would be better, if system able to find the better solution by itself and guide robot to the goal.

A number of methods have been used before to attempt to solve the problems of autonomous. One method a robot system could use to learn about its environment is take image and finds obstacles around and then finds the shortest path between starting point to goal point by avoiding these obstacles. By this method a system would then learn where obstacles are as it encounters them and can avoid them while navigating robot from a starting point to target point. Using this information the system can decide how to move robot from point A to point B in an efficient manner without running into the previously mapped obstacles.

The path planning method described in this paper aims to calculate the safe path more effectively. Computational efficiency would contribute largely to future space exploration missions because the performance of CPU is limited in space (e.g. clock frequency is one tenth of our PCs) [4]. The proposed method is based on stereo images processing which can be captured by the camera set equipped on system or satellite.

II. OBSTACLE RECOGNITION

Digital image processing techniques have been widely used in various types of application recently. Recognition of obstacle using vision (camera) system is a challenging task in the field of image processing. In the system like autonomous robot, the ability and efficiency of system to capture and process the images is very much essential.

The vision system recognizes an obstacle in a Region of Interest of a targeted image. The vision system includes an image change unit and a Region of Interest detection unit. The image change unit converts the obstacle into an edge image which represents as an edge of obstacle. The region of interest detection unit divides the edge image into a plurality of regions, Get a sum of edge component values of an edge line in each of the regions and compare this total sum with a predetermined (experimentally) threshold value by respected regions, then get a region, in which the total sum of edge component values is greater than the threshold value, as the region of interest from among the plurality of regions.

Calculating a total sum of edge component values of the edge line for each region, comparing the total sum of edge component values and a predetermined threshold value, and detecting an region of interest from the regions on the basis of the compared result for each region; and an image analysis unit scanning the detected region of interest by block units having a certain size to analyze whether an obstacle exists in the detected region of interest.

The edge image generation unit changes the source image into an intermediate image which is composed of an edge line of the object and binaries the changed intermediate image to generate the edge image having a gray scale.

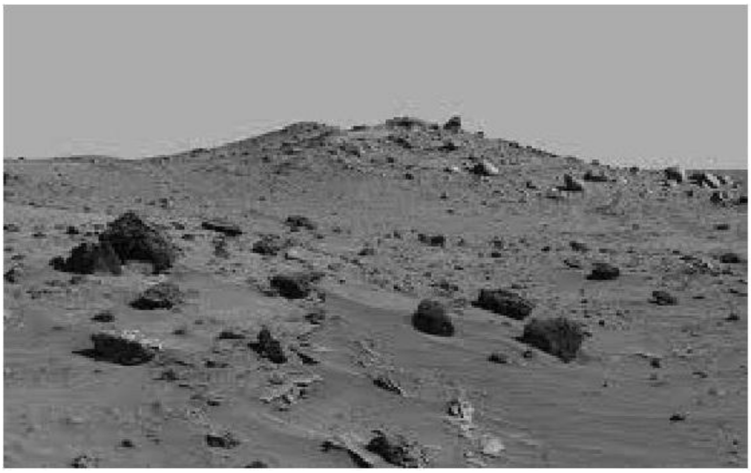

Fig.2. Gray scale image of ROI

A. Obstacle Detection

The first step of a sequence of algorithms is to detect obstacles from one captured image. All images are processed as grayscale images in this study, because the information quantity can be reduced to one third of color images, and the color information is unlikely to be crucial in planetary environment (Shown in fig2). Since natural environment is difficult to be recognized by only one criterion, the obstacle detection method described in this section combines two fundamental image processing: edge extraction and thresholding. The edge of the image represents the outline of obstacles, and the surface and the shadow of obstacles have brighter or darker pixels. After deriving the edge and the surface extraction results, they are integrated in order to compensate errors of one result with the other [4].

The edge of obstacles is extracted by calculating the variance in a small square window which has width of w[pixel]. The variance value V (p, q) at (p, q) on image is namely calculated as follows.

$$V(p,q) = \frac{1}{w^2} \sum_{i=p-\frac{w}{2}}^{p+\frac{w}{2}} \sum_{j=q-\frac{w}{2}}^{q+\frac{w}{2}} (A(p,q) - x(i,j))^2 \qquad (1)$$

Where A(p, q) is the average of pixel values in the window, x(i, j) is the pixel value at (i, j) [4].

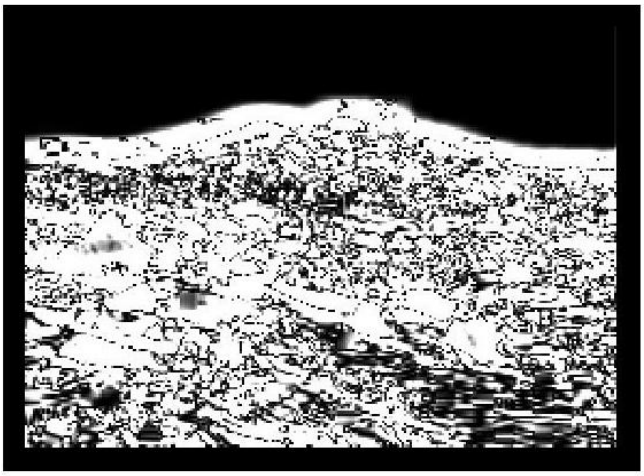

Fig.3.Image after Variance calculation

The variance is calculated on each pixel (Shown in fig 3). and then binaries by the threshold Th, which is one of the unique points in the proposed obstacle detection method, given as follows,

$$T_h = \overline{V(p,q)} + t \cdot \sigma \qquad (2)$$

Where σ is a standard deviation of V (p, q), t is a experimentally decided constant. By this expression, the threshold, which is the key parameter in binarizing process, is decided according to the captured image [4].

This thresholding process expects rover to properly detect obstacle areas even in natural and unknown environment such as planetary surface because the algorithm is robust to the change of light condition.

Fig.4. Result of obstacle detection

The surface of obstacles is detected by extracting bright and dark pixels. Thresholds are derived in the same way as Eq. (2) in order to secure the robustness. Finally, these results are mixed inclusively, and areas which do not have both edge and surface information simultaneously are rejected, since such areas tend to represent no obstacles but tiny roughness of the surface (Shown in fig 4).

III. PATH PLANNING

A common problem in navigating robot is to produce a continuous motion that connects a start configuration 'S' and a goal configuration 'G', while avoiding obstacles along the way. The robot and obstacle geometry is described in a 2D or 3D workspace, while the motion is represented as a path in configuration space.

The set of configurations that avoids collision with obstacles is called the free space Cfree. The complement of Cfree in C is called the obstacle or forbidden region [6].

Simplifying the search area, is the first step in path planning and to simplifying the search area, the source image is divided into a square (node) grid which is called as grid-based approach (Shown in fig 5). This particular grid-based approach reduces search area to a simple two dimensional array. Each item in the array represents one of the nodes on the grid. The shortest path is found by figuring out which node should take to get from start configuration 'S' to goal configuration 'G'. Once the path is found, robot can move from the center of one node to the center of the next until the goal configuration 'G' is reached.

Low-dimensional problems can be solved with grid-based algorithms that overlay a grid on top of configuration space, or geometric algorithms that compute the shape and connectivity of Cfree. Grid-based approaches overlay a grid on configuration space, and assume each configuration is identified with a node. At each node point, the robot is allowed to move to adjacent node as long as the line between them is completely contained within Cfree [6].

This discretizes the set of actions, and search algorithms (like A- Star) are used to find a path from the start configuration 'S' to the goal configuration 'G'.

The performance of grid-based approach is based on the size of grid. Grid-based path planning must minimize response time and memory usage, these approaches require setting an appropriate grid resolution. Path planning is faster with large grids size, but the algorithm will fail to find paths through narrow portions of Cfree. Furthermore, the number of points on the grid grows exponentially in the configuration space dimension, which makes them inappropriate for high-dimensional problems [6].

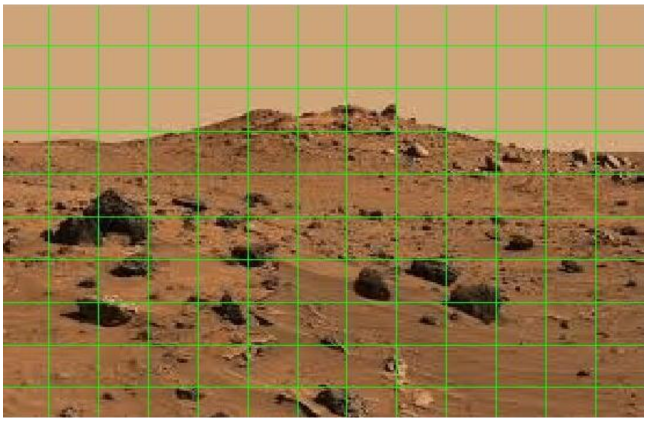

Fig.5. Grid-based approaches

Grid-based approaches often need to search repeatedly, for example, when the knowledge of the robot about the configuration space changes or the configuration space itself changes during path following. Incremental heuristic search algorithms replan fast by using experience with the previous similar path-planning problems to speed up their search for the current one

A. Path Planning

Navigating a terrain and planning the shortest path to a goal configuration, is one of the fundamental problems in path planning. While there are many approaches to path planning, one of the most common and widely used is the A-Star path algorithm.

A-Star requires a distance function from start configuration to goal configuration. It does not need an exact distance, just an estimate of how long it would take to get from start configuration to the goal configuration, in the best case.

The A-Star algorithm combines features of uniform-cost search and pure heuristic search to efficiently compute optimal solutions. A-star uses the distance between the current configuration and the target configuration and moves to the node that has the smallest distance. A-Star algorithm is based on best-first search algorithm in which the cost associated with a node is $f(n) = g(n) + h(n)$, where $g(n)$ is the cost of the path from the initial state to node n and $h(n)$ is the heuristic estimate or the cost or a path from node n to a goal node. Thus, $f(n)$ estimates the lowest total cost of any solution path going through node n. At each point a node with lowest f value is chosen for expansion. Ties among nodes of equal f value should be broken in favor of nodes with lower h values. The algorithm terminates when a goal is chosen for expansion.

The $h(n)$ part of the $f(n)$ function must be an admissible heuristic; that is, it should not overestimate the distance to the goal configuration. Thus, for an application like routing, $h(n)$ might represent the straight-line distance to the goal configuration, since that is physically the smallest possible distance between any two nodes

The algorithm begins by creating an open and a closed list. The open list contains the entire node that can be reached from the current node, and the closed list contains the parent of the current node. Now pick from the open list the node with the lowest distance. This becomes new current node, and the previous node is added to the closed list.

Now that we have a new current square, check the node adjacent to it. If the adjacent node is unpassable or on the closed list, then ignore it. If the adjacent node is not on the open list, add it. Finally, if the adjacent node is already on the open list, check if this new path is shorter, and, if it is, then change that node's parent to the current node. The algorithm ends when the goal node has been added to the closed list.

B. The A- Star Algorithm

1) Put the start node on the list OPEN and calculate the cost function f (n). {h (n) = 0; g(n) = distance between the goal and the start position, f(n) = g(n).}

2) Remove from the List OPEN the node with the smallest cost function and put it on CLOSED. This is the node n. (Incase two or more nodes have the cost function, arbitrarily resolve ties. If one of the nodes is the goal node, then select the goal node)

3) If n is the goal node then terminate the algorithm and use the pointers to obtain the solution path. Otherwise, continue

4) Determine all the successor nodes of n and compute the cost function for each successor not on list CLOSED.

5) Associate with each successor not on list OPEN or CLOSED the cost calculated and put these on the list OPEN, placing pointers to n (n is the parent node).

6) Associate with any successors already on OPEN the smaller of the cost values just calculated and the previous cost value. (min(new f(n), old f(n)))

7) Goto step 2.

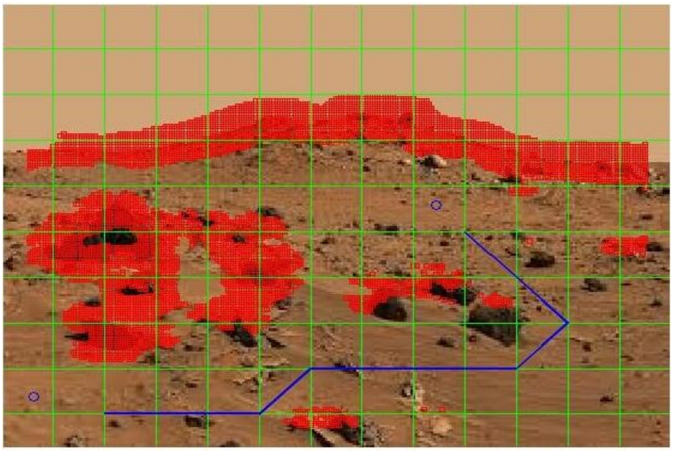

Fig.6. Path Planning

IV. CONCLUSION

'Image based path planning using A-star algorithm' system appears to be an excellent solution for system where the expensive sensors cannot be use for their autonomous robot. The obstacle detection and the path planning method attempt to address the need for the efficient autonomous robot with fewer resources by extracting required information from captured images The location of obstacles is identified by the images captured from camera. Then, using the A-Star algorithm technique, the shortest path to the goal configuration is determined.

The obstacle detection system detects obstacles by combining two fundamentals of image processing: variance (equation 1) and thresholding (equation 2). By searching dark and bright pixels, surface and the edge of obstacle is extracted from high variance in window.

The combining process compensates for errors which occur in each calculation to get robust detection of obstacles, the result of obstacle detection is as shown in Fig 4.Grids are easy to implement and offer fast memory access. The performance of grid-based approach is based on the size of grid. The selection of appropriate grid size result into the better results. A grid-based map obtained from the vision-based obstacle detection system given as an input to the A-star algorithm. If we use an admissible heuristic, A-Star algorithm is capable of finding out the most optimal path for the robot shown in fig 6.

REFERENCES

[1] L. Pedersen, D. Kortenkamp, D. Wetterngreen, I. Nourbakhsh "A Survey of Space Robotics"

[2] Claudiu Popirlan, Mihai Dupac "An Optimal Path Algorithm for Autonomous Searching Robots" Annals of University of Craiova, Math. Comp. Sci. Ser. Volume 36(1), 2009, Pages 37–48 ISSN: 1223-6934

[3] Xavier P_erez, Cecilio Angulo, Sergio Escalera "Vision-based Navigation and Reinforcement Learning Path Finding for Social Robots" September 3, 2010.

[4] Mariko Sakuta, Shogo Takanashi and Takashi Kubota "An image based path planning scheme for rovers" in 2011 IEEE International Conference on Robotics and Biomimetics December 7-11, 2011, Phuket, Thailand.

[5] Stephen J. Tobias, A. Antonio Arroyo "Autonomous Pathfinding", 2000 Florida Conference on Recent Advances in Robotics May 4-5, 2000, Florida Atlantic University.

[6] "Pathfinding" website.[online] Available: http://en.wikipedia.org/wiki/Pathfinding

[7] Mark W. Maimone, P. Chris Leger and Jeffery J. Biesiadecki: "Overview of the Mars Exploration Rovers' Autonomous Mobility and Vision Capabilities", International Conference on Robotics and Automation, 2007.

[8] Basem M. ElHalawany, Hala M.Abdel-Kader, AdlyTagEldeen, Alaa Eldeen Elsayed "Modified A* Algorithm for Safer Mobile Robot Navigation" U2013 Proceedings of International Conference on Modelling, Identification & Control (ICMIC)Cairo, Egypt, 31stAug.- 2nd Sept. 2013