

Comparison of Various Obstacle Avoidance Algorithms

Vayedha Anshav Bhavesh
Instrumentation & Control Engineering,
Electrical Engineering Department
Institute of Technology, Nirma University
Ahmedabad, India

Abstract— In this paper, we address an issue related to autonomous navigation of mobile robots and a comparison of various existing solutions of the problem with merits and demerits of each. We put forward a comparison of various obstacle avoidance algorithms. Obstacle avoidance is back bone of autonomous navigation as it enables robot to reach desired location avoiding hurdles in the path. Many methods of realizing obstacle detection and collision avoidance have been proposed by researchers. Paper discusses some of widely accepted algorithms starting from the most rudimentary of all ‘bug algorithms’ to sophisticated algorithm ‘optical flow’.

Keywords—UAV-Unmanned Ariel Vehicle; UGV-Unmanned Ground Vehicle; APF- Artificial Potencial Field; VFH-Vector Field Histogram

I. INTRODUCTION

Obstacle avoidance is one of the prime issues related to autonomous navigation of mobile robots. It is not a viable option to model entire environment in which robot moves. In order to maneuver in dynamic environment robot has to be equipped with obstacle avoidance algorithm to deal with hurdles which are not known beforehand. It can be used for different types of machines such as modern cars, industrial robots or Unmanned Ariel Vehicles (Drones).

Obstacle avoidance can be classified in two sub stages called obstacle detection and collision avoidance. Different algorithms use different kind of sensors for obstacle detection. Data received from sensor is processed and controller sends signal to end effector in order to avoid obstacle.

Primitive algorithms used to stop robot to avoid collision but modern techniques enables the robot to detour obstacle using some kind of quantitative measurement of dimensions of obstacle. This paper discusses some of widely accepted algorithms such as bug algorithms, APF, VFH, bubble band technique and fixed sonar technique. Bug algorithms can be classified in four categories namely Bug 1, Bug 2, DistBug and Tangential Bug algorithm. We have implemented optical flow algorithm on simulation platform Opencv to drive a UAV. Results of simulation are shown in this work. Best technique for avoiding obstacle depends on specific environment and equipment availability. At the end we have summarized all the algorithms by listing merits and demerits of all which can be helpful in selecting proper technique according to our requirements and resources.

II. BUG ALGORITHMS

A. Bug 1 algorithm

Bug 1 algorithm is the simplest of all algorithms discussed in this work. It reaches to the goal all most all the times giving high reliability. But the matter of concern with this method is efficiency. Robot moves on the shortest path joining robot’s position X and goal location until it encounters a hurdle in the path. When it is confronted by an obstacle, it starts revolving around its surface and calculates distance from destination point. After one complete revolution it figure outs the point of departure which is closest to goal. It maintains or changes direction of motion depending on distance of leaving point from hit point. This method can be illustrated in three steps

- Head towards the goal
- if an obstacle is encountered, circumnavigate it and remember how close you get to the goal
- return to that closest point and continue

Robot revolves around each and every obstacle in the way towards goal increasing computational efforts. But ease of implementation makes it worth when only completion of task is required irrespective of time.

B. Bug 2 Algorithm

Bug 2 algorithm is modified form of Bug 1 algorithm. Instead of searching for minimum distance point, Bug 2 algorithm focuses on maintaining direction of motion towards goal. It calculates slope of the line joining initial point and desired point. When robot encounters an obstacle, it starts moving along the edge of the obstacle until it finds a point with the same slope. It starts moving on the line joining point of departure and goal. This algorithm is easy to program and provides better performance than Bug 1 algorithm in majority of cases. Robot does not need to encircle the entire object as in the Bug 1 algorithm.

C. Distance Bug Algorithm

Robot travels comparatively shorter distance than earlier versions of Bug algorithm allowing the robot to reach destination in less time. This algorithm exhibit different obstacle avoidance behavior. When confronted by an obstacle, it starts following the edge of the obstacle simultaneously calculating and storing the distance from its current and next position to destination. The leaving point, where it switches the behavior from obstacle avoidance to move to goal, is selected based on the condition that the distance of destination from its next position is greater than

the corresponding distance from its current position. The robot continues its obstacle avoidance behavior other-wise

D. Tangent Bug Algorithm

Tangent Bug algorithm finds tangents to the obstacle and calculates distances of robot from points where they touch the obstacle. These points are denoted by O_i , where i denotes index number. In this approach robot takes path of the tangent which maximally decreases heuristic distance i.e. $d(\text{robot position}, O_i) + d(O_i, \text{goal})$. At times it has to act as bug algorithm by following the edge of the obstacle if heuristic distance starts increasing while moving towards O_i . It is a memory type algorithm which exhibits motion-to-goal and boundary following behavior. A value d_{\min} which is the shortest distance observed so far between the sensed boundary of the obstacle and the goal and d_{leave} which is the shortest distance between any point in the currently sensed environment and the goal are continuously updated. It terminates boundary following behavior when $d_{\text{leave}} < d_{\min}$

III. ARTIFICIAL POTENCIAL FIELD MATHOD

APF assumes robot and obstacle to be having same polarity of electric potential whereas goal to be having opposite polarity. This implies that robot is repelled by obstacle and attracted by goal. Robot moves in the direction of resultant of all force vectors acting on it. Robot's velocity decreases as it moves closer to goal and it stops when distance becomes zero. This method falls into a trap when robot reaches a point of local minima. It is a point other than destination point where resultant of all the forces is zero. This situation is depicted in the diagram where robot is shown in green color and destination point in red.

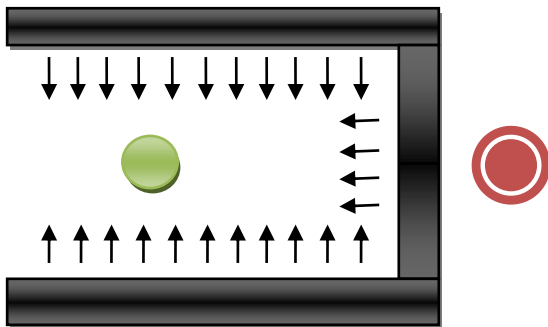


Fig. 1. Example of local minima

IV. VECTOR FIELD HISTOGRAM

VFH constructs a polar histogram based on knowledge acquired through range sensors. This approach overcomes issue of sensor noise. Histogram is a graph between probabilities of presence of obstacle to angle associated to sonar sensor. The probabilities are obtained by creating a local occupancy grid map of the environment of robot's surrounding. The histogram is used to discover all the passages large enough to allow the robot to pass through. Selection of path is based on cost function which is a function of alignment of the robot's path with the goal, and on the difference between the current wheel orientation and the new direction. Minimum cost function is desirable.11

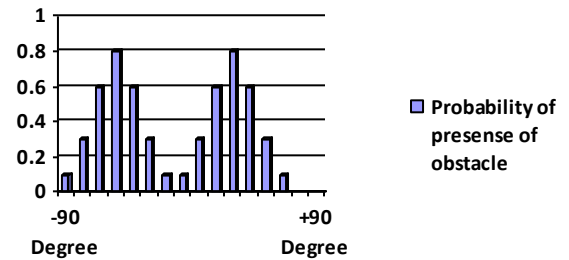


Fig. 2. Example of polar histogram

V. BUBBLE BAND TECHNIQUE

Bubble band method was proposed by khatib in [1]. This method defines a bubble around the robot containing maximum free space, which can be travelled in any direction. Shape and size of bubble are function of model of robot's geometry and range sensor information. A band of such bubbles can be formed for navigation of robot avoiding collision. It comprises distance sensors placed at fixed angular distance. Concept of bubble is used to judge whether the obstacle will cause a possible collision. Combining the profile of the boundary of the obstacle with the bubble concept, we can determine the maneuvers to avoid collisions

VI. FIXED SONAR METHOD

Fixed sonar sensors are used to measure distance of obstacle from robot's current position. Two sonar sensors are mounted on the front part of robot. Position of obstacle is classified in different zones based on combinations of sensor readings. Distance can be obtained by velocity formulae i.e. $v = d/t$. Where v = velocity of sound in air = 340 m/s. In this method care has to be taken in placing sonar sensors so that they do not detect ground as obstacle. Ease of implementation and less computation requirements are important merits of this method. But there are some major demerits of this method. Scattering of wave occurs only when the roughness of an object's surface is large compared to the wavelength of incident wave. Ultrasonic waves have wavelength much larger in magnitude. So, they do not get reflected from extremely smooth surface. Also speed of sound is dependent on air temperature. 10° variation in temperature causes 1% error in distance. Geometry of object also affects distance value obtained from sonar.

VII. OPTICAL FLOW

As an observation point moves through the environment, the pattern of light reflected to that point changes continuously, creating optic flow. Optical flow or optic flow is the pattern of apparent motion of objects in a visual scene caused by the relative motion between an observer device (camera) and the scene. The optical flow methods try to calculate the relative motion between two image frames which are taken consequently at every position. Robot can use this information to determine flow in images. This flow indicates any moving object in the way of robot. If flow is detected, robot avoids obstacle by changing signals to end effectors.

In our experiment we have obtained optical flow in images captured through a video camera using python code in Opencv. Relative motion between camera and objects causes motion in image sequence. The 3-D motion of the objects and the camera induces 2-D motion on the image plane. It is this 2-D motion, called optical flow that needs to be recovered from intensity and color information of a video sequence. Most of the existing methods for motion estimation can be categorized in four ways: The correlation based methods, the energy based methods, the parametric model based methods and the differential based methods. We have opted for a differential technique based on the intensity conservation of a moving point to calculate the optical flow. For ease of implementation, we have calculated number of flow vectors in left part and right part of screen. If it exceeds a fixed number (in our experiment it is 20), we can consider it as presence of moving obstacle. This approach overcomes effect of noise. Moreover optical flow enables us to detect moving obstacle. In dynamic environment this method is helpful to drive unmanned vehicles (UAV, UGV etc.). We used optical flow for navigation of an Unmanned Ariel Vehicle negotiating the obstacles in the path. The frame was divided into two halves Left and Right. Optical flow vector was calculated for the two consecutive frames in both the halves. As per the vector magnitude and direction, the autopilot system was made to move the UAV in the other half, thus avoiding the obstacle. To deal with noise due to vibration of UAV frame we set a threshold value of flow vector which was determined by multiple experiments. Python code for optical flow is shown below.

```
#Python Code For optical flow
k=0.5
right=0
left=0

for y in range(0, flow.height, self.mv_step):
    for x in range(flow.width/2, flow.width,
self.mv_step):
        fx, fy= flow[y, x]
        if (abs(fx) > k) and (abs(fy) > k):
            color=(0,0,255):
            right=right+1
        else:
            color = (0,255,0):
            Line(self.cflow, (x,y), (int(x+fx),int(y+fy)), color)
```

We implemented this algorithm using Opencv and obtained result as shown below.

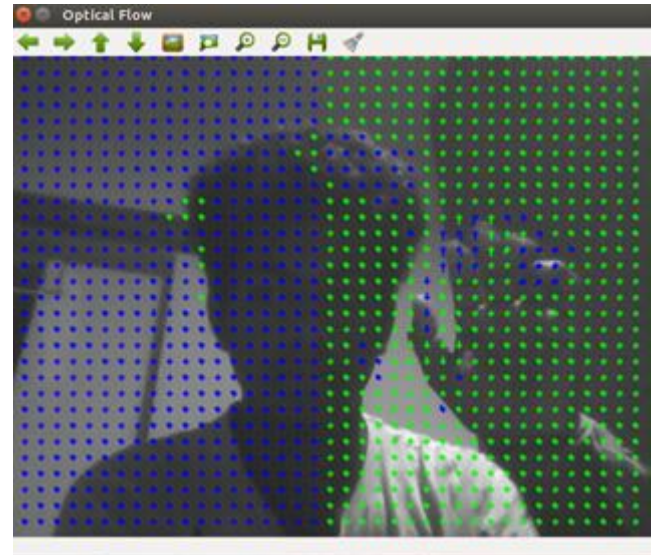


Fig. 3. Optical flow experimental result(Flow vectors)

VIII. COMPARISON AND CONCLUSION

After discussion of various methods of object avoidance we will list merits and demerits of each which can be useful in choosing one for specific application. Selection of best method depends on specific requirements and resources.

A. Bug 1 Algorithm

Merits:

- It is complete algorithm i.e. in finite time, it finds a path if such a path exists or terminates with failure if it does not.

Demerits:

- Large memory and computation is required in Bug 1 which results in lesser efficiency.
- It is exhaustive search algorithm

B. Bug 2 Algorithm

Merits:

- It is a greedy search algorithm i.e. it takes the first thing that look better.
- It requires less computation also it is easy to implement.

Demerits:

- In some cases it does not exhibit completeness.

C. Distance Bug Algorithm

Merits:

- It traverses less distance than Bug 1 and Bug 2 algorithm to reach destination
- Its path length is almost close to the length of Bird's eye-view because DistBug algorithm uses information coming from range sensors efficiently.

Demerits:

- It thinks about obstacle only when it encounters one and starts circumnavigating it whereas Tangent bug starts moving tangentially to obstacle showcasing pre-cognitive behavior

D. Tangent Bug Algorithm

Merits:

- It tries to minimize the heuristic distance so that robot has to move minimum to reach up to desired location.

Demerits:

- It is not a self-sufficient approach. While moving on a tangential path when distance starts increasing, it starts acting as a bug algorithm by following the edge of obstacle.

E. Artificial Potencial Field Method

Merits:

- It is a simplistic approach which is easy to implement

Demerits:

- Robot stops when it encounters a point of local minima
- It cannot detect devoid of passage between closely spaced obstacles
- Oscillations is an issue when it tries to negotiate with obstacle in vicinity or passes through narrow passage

F. Vector Field Histogram

Merits:

- It conquers problem of sensor noise by making polar histogram which represents probability of presence of obstacle in particular angular direction

Demerits:

- It does not guarantee completeness
- Local minima might not be negotiated
- It can be problematic to pass through narrow passage using this method
- It does not consider dynamics of robot

G. Bubble Band Technique

Merits:

- It out performs VFH while passing through narrow passages
- It demands less computation
- This method is cost effective as well flexible enough with respect to different sensors
- It can avoid static obstacle as well as it can negotiate with moving obstacle up to a certain level

Demerits:

- Does not provide smooth operation
- It requires higher level path planner
- It is vulnerable to sensor noise. All the limitations of sonar sensor affects operation of this algorithm
- Cannot guarantee reaching the desired point

H. Fixed Sonar Method

Merits:

- This method is cost effective and simple to implement
- Less data handling and computation is required

Demerits:

- Sonar cannot detect extremely smooth surface e.g. glass
- Output of sonar distance sensor is dependent on temperature of medium
- It depends greatly on geometry of obstacle making it a less reliable technique.
- Sonar sensor has to be mounted with care so that it does not detect ground as an obstacle.

I. Optical Flow

Merits:

- It is a sophisticated and reliable technique to detect moving obstacle

Demerits:

- It cannot detect stationary obstacles.
- It requires huge computation and data handling making it an expensive algorithm to implement
- Requires high speed microcontroller for camera interfacing and real time image processing

IX. ACKNOWLEDGMENT

I gratefully acknowledge the contributions of Prof. H K Patel (Associate Professor at Institute of Technology, Nirma University) for his valuable guidance in preparing this document.

X. REFERENCE

- [1] Khatib, O., Real-Time Obstacle Avoidance for Manipulators and MobileRobots., 5, 1986. The International Journal of Robotics Research: p. 90-98.
- [2] U. Dravidam and S. Tosunoglu, "A survey on automobile collision avoidance system", Florida conference on recent advances in robotics 1999
- [3] Choi, C., Crane, C., and Matthew, G., "Obstacle Avoidance Via Articulation, " Journal of Robotic Systems, Vol 8, pp. 465-484 , August 1991
- [4] Z. Yi, Y. Khing, C. Chin Seng, and Z. Xiao Wei, "Multi-ultrasonic sensor fusion for mobile robots", Proceedings of intelligent vehicle symposium, pp. 387-391, 2000.
- [5] M. ohaib, S.M Pasha, N. Javaid, J Iqbal, "Intelligent Bug Algorithm (IBA): A Novel Strategy to Navigate Mobile Robots Autonomously"
- [6] J.C. Baker, A Simon, "Sensor and navigation data fusion for an autonomous guided vehicle", Proceedings of Intelligent vehicle symposium, pp. 156-161, 2000.
- [7] Buniyamin N., Wan Ngah W.A.J., Sariff N., Mohamad Z., "A Simple Local Path Planning Algorithm for Autonomous Mobile Robots", INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, ENGINEERING & DEVELOPMENT, Issue 2, Volume 5, 2011
- [8] W.H. Munro, S. Pameroy, M. Rafiq, H.R. Williams, M.D. Wybrow and C Wykes, "Ultrasonic vehicle guidance transducers", Ultrasonics, Vol 28, pp. 349-354 1990.
- [9] Zhu, Y., Zhang, T., Song, J., & Li, X. (2012). A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge. Knowledge-Based Systems, 27, 302-313
- [10] G.D. Hager, Z. Dodds, "Robotic Motion Planning: Bug Algorithms", unpublished.
- [11] Sgorbissa, A., & Zaccaria, R. (2012). Planning and obstacle avoidance in mobile robotics. Robotics and Autonomous Systems, 60(4), 628-638