# An Optimal-Path-Finding Algorithm capable of Detecting all Obstacles within a Specified Region

**Amritam Sarcar**
Dept. of Computer Science and Engg.,
St. Thomas’ Collg. of Engg. & Technology Kolkata,
Kolkata 700023 WB INDIA
amritamsarcar@yahoo.co.in

February 8,2007

***Abstract-* This paper studies the problem of finding an optimal path within a region avoiding objects placed within the region. For detecting objects, we suggest the use of ultrasonic sensors. We then consider the problem of optimising the path to unambiguously identify each obstacle. We show that the problem can be tranformed from a geometrical to graphical domain, and then attempt to transform into numerical domain. We present some results on Lagranges’ Interpolation that help us progress towards the latter transformation, and hence find the optimal path. Also, we outline an approach to estimate its efficiency.**

## 1 Introduction

Optimal path finding is a challenging problem, not least because optimality is variously defined in this context. For instance, we may say a path is optimal if it covers every part of a region for ensuring the detection of obstacles. Alternatively, a path may be considered optimal if it is the shortest distance between its initial and final destination.

We will consider optimality in the light of *detection and avoidance of obstacles* and *least possible distance.* Briefly, we are interested in finding the obstacles. We assume we cannot examine the obstacles directly, so we must resort to observing the readings from a set of strategically placed sensors.

We will show that our model has strong links with studies of interpolation point sets. In particular, our work examines, among other things, the conditions under which a set of obstacles (which are represented as coordinates in 2-dimensional space) will always be detected and the path so outlined is the shortest one. Probabilistic estimates of a set of points in general position in 2-dimensional space have been derived. Later work addressed the case when the points are not in general position. Although we do not yet have the general conditions, we provide a subsidiary result for obstacle detection which we hope to be able to extend to the general case.

Our final goal is to compute a general equation for the trajectory of the path and use its structure to design an efficient algorithm that gives optimal or near-optimal paths. In this paper, we briefly describe the problem, suggest a method for the detection of the obstacles, examine the problem graphically, establish a link with Lagrange’s interpolation formula, obtain some simple bounds on the size of the input space, present results that help us progress towards the design of an algorithm and finally outline an approach to estimate its efficiency.

## 2 The Problem

We will consider a region with m obstacles. We are given a set of sensors that would unambiguously identify the obstacles. Each sensor has a specific range of vision. Let the range be e. The sensors are placed parallel to x and y-axes. For simplicity, we restrict the number of sensors to three. We are to formulate the trajectory of a path that would detect these obstacles and travel in least possible time from its initial to final destination.

## 3 The Geometrical Picture

Due to the sensors’ orientation it may be shown that for proper detection of obstacles, the sensors must be perpendicular/parallel to the objects.

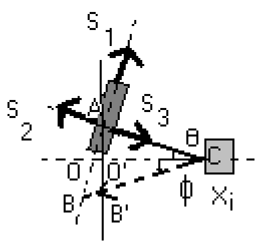



Fig.1. Shows the effect of sensor($S_3$) not positioned perpendicular to the object Xi where i≤m

From rectilinear propagation of sound, it may be argued that

/ACO = /OCB i.e. Θ=Φ (since angle of incidence = angle of reflection).

Due to simplicity, it is assumed that the receptor and the source of a sensor lies at the same point. Thus from Fig. 1 it is observed that pt. A and pt. B does not coincide. Point A being the source. For proper detection of obstacles OB=0. From the figure it can be established that for OB=0, O′B′=0. Now, in ΔO′CB′ /CO′B′ = 90 degrees.

tan Φ = O′B′ / B′C i.e. tan Φ = 0/B′C = 0.
i.e. Φ = tan-□ 0 = 90 degree.

This yields that the angle of reflection is 90 degree, which proves that the sensors must be aligned properly.

## 4 Towards a Pure Graph Problem

We would like to remove the geometrical component and transform the problem into the graph domain, so that we can try to use the large body of results in co-ordinate geometry.

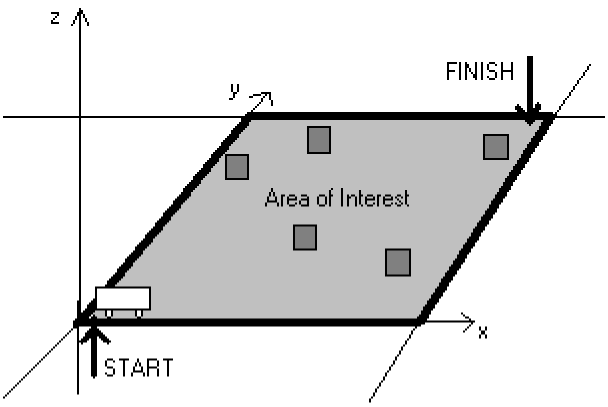



Fig.2. Shows the position of the vehicle where sensors are positioned, the obstacles and the area of interest/ region.

The 3-dimensional coordinate system can easily be simplified to a 2-dimensional one without any loss of information as everything happens in the x-y plane. In x-y coordinate system the position of the vehicle/sensors are denoted by (pi,pj) and every obstacle (xi,yi).

## 5 A Mathematical Formulae

The coordinate points of the sensors and the obstacles are of the form (x,y). Further, it may be argued that the trajectory of the vehicle is a function either of x or y-only and not both (since the position of the obstacles are fixed in time-space domain).

Let y=f(x) be a real valued function and is known for m equally/not equally spaced points. The function x gives the paths′ trajectory. From Lagrange′s intepolation method [1] we can now construct a polynomial function Φ(x) such that

$$\Phi(x_i)=f(x_i)=y_i,(i=0,1,2,...m-1) \qquad ....(1)$$

Now Φ(x) being a polynomial, we may take as

$$\Phi(x)=a_0(x-x_1)(x-x_2)...(x-x_{m-1}) + a_1(x-x_0)(x-x_2)...(x-x_{m-1})+...+ a_{m-1}(x-x_0)(x-x_1)...(x_{m-2}) \qquad ....(2)$$

Now the constant ai′s are to be determined as follows:

Putting $x=x_0$ in (2) and using (1), we get

$$y_0=a_0(x_0-x_1)(x_0-x_2)...(x_0-x_{m-1})$$

Hence, $a_0=\dfrac{Y_0}{(x_0-x_1)(x_0-x_2)...(x_0-x_{m-1})}$

Proceeding the same way, we have

$$a_1=\frac{Y_1}{(x_1-x_0)(x_1-x_2)...(x_1-x_{m-1})}$$

Substituting the values of ai′s we get

$$\Phi(x) = \Sigma l_i(x)y_i \qquad ...(3), \quad \text{where}$$

$$l_i(x) = \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_{m-1})}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_{m-1})}$$

Here Φ(x) is the path equation of the robot where ($x_i$, $y_i$ ) are the positions of the obstacles, (i=0,1,2,...,m-1).

## 6 A Review of the Problem

We observe that the trajectory of the path so found fails to avoid collision with the obstacles, but detects them. Thus we are required to modify the formula.

## 7 Towards a Solution

We initially consider a segment of the path say, XY. Here X is the present location of the vehicle and Y is an obstacle. The path so formed using equation (3) is XYPQRS. However, points Y,P,Q,R are obstacles. Thus we seek to find an alternate/better path.

(For proper detection of obstacles it follows from 3 that the vehicle must be on the line passing pt. Y and __|__ x-axis.)

Thus point Y′ must be on the line $x=x_j$ where $x_j$=constant for a particular obstacle. Pt. Y′ becomes $(x_j,y')$. The aim is to minimise the lateral distance from the obstacle.

From figure 3, it is evident that:
In $\Delta XYY'$ $\angle XYY' = \Phi$ degrees.
Now, $\cos\Phi = YY' / XY$.

$$\text{i.e. } \cos\Phi = \left| \frac{\sqrt{(y'-y_i)^2}}{\sqrt{(x_j-x_i)^2+(y_j-y_i)^2}} \right|$$

The denominator is constant = K(say).

$$\text{i.e. } \cos\Phi = \left| \frac{(y'-y_i)}{K} \right|$$

We know $-1 \le \cos\Phi \le +1$, however $0 \le |\cos\Phi| \le 1$.

$$\text{i.e. } \left| \frac{(y'-y_i)}{K} \right| = 0 \text{ , for minimum value.}$$

Thus, $y'-y_i=0$, i.e. $y'=y_i$. Similarly, proceeding in the same manner pts. P′,Q′ and R′ are found.

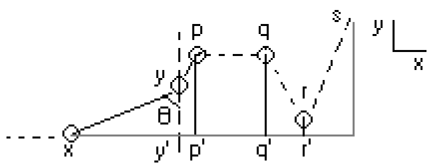


Fig.3. The motion path of the vehicle is shown. Here the pts. of X and Y are $(x_i,y_i)$ and $(x_j,y_j)$.

### 7.1 Optimal Path

Thus we arrive at two paths XYPQRS and XY′P′Q′R′S We show that the latter path is the shorter one than the former; and then go on to prove this is the *least-distance-obstacle-avoidance* path.

Let the coordinates of X,Y,P,Q,R,S be of the form (xi,yi) where i= 0,1,2,3,4,5.

Using Co-ordinate geometry we have the distance D as

$$D = \sqrt{(x_1-x_0)^2+(y_1-y_0)^2} + \sqrt{(x_2-x_1)^2+(y_2-y_1)^2} + \ldots + \sqrt{(x_5-x_4)^2+(y_5-y_4)^2}$$

Now the coordinates for X,Y′,P′,Q′,R′,S in XY′P′Q′R′S are $(x_0,y_0)$, $(x_1,y_0)$, $(x_2,y_0)$, $(x_3,y_0)$, $(x_4,y_0)$, $(x_5,y_5)$ (from fig. 3). Similarly we compute distance D′ as

$$D' = x_1-x_0+x_2-x_1+x_3-x_2+x_4-x_3+x_5-x_4+y_5-y_0$$

This reduces to

$$D' = x_5-x_0+y_5-y_0.$$

It can be easily shown that D′<D. Details are available in [2]. Furthermore, generalizing the above result we can show that any curve-linear path is more than straight-line path distance [2]. Hence, the trajectory of the path so formed is the least in any given region.(Since it travels through the longest straight-line distance).

This reduces the problem to only finding such points in the region that are at a minimal distance from an obstacle (and also detects it). Again using Lagrange's interpolation methods we replace $y_i$ in equation (1) with $y_i+\Delta_i$. Thus (1) becomes

$$\Phi(x_i)=f(x_i)= y_i+\Delta_i,\ (i=0,1,2,\ldots m-1)$$

where the objective is to minimise $\Delta_i$ and $\Delta_i > 0$

Proceeding in similar fashion we get

$$\Phi(x) = \sum l_i(x)(y_i+\Delta_i) \qquad \ldots(4), \quad \text{where}$$

$$l_i(x) = \frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_{m-1})}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_{m-1})}$$

Here $\Phi(x)$ is the path equation of the robot where $(x_i, y_i)$ are the positions of the obstacles, (i=0,1,2,...,m-1).

## 8 Simulation

It is difficult to directly test new methods for motion control, since the robot batteries get drained quickly and long experimental runs are not possible. Hence we designed a software simulator for the micro-robot environment (a screen shot is provided in Fig.3). The simulator implements random placement of obstacles across different environment. We are working to adapt the interface of the simulator library to be exactly the same as that of the actual robot control module. This would enable the physical equipment to be efficiently replaced by their virtual counterparts for testing purposes. Of course, since such simulations can never be entirely accurate, the final testing will have to be on the hardware.

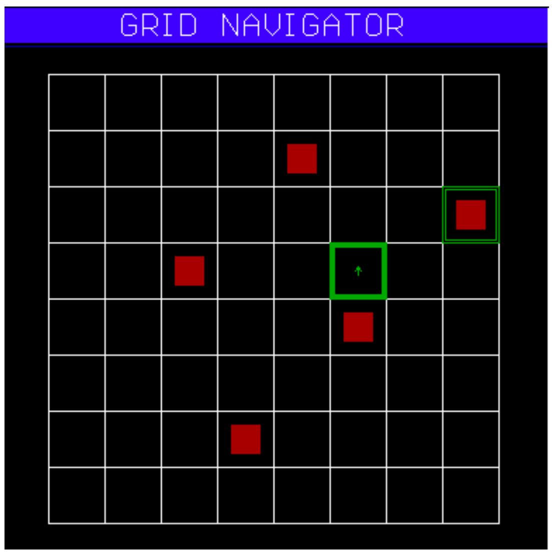



Fig.3. Grid navigator simulator in 8x8 grid with 5 obstacles

## 9 Results

On simulation it was found that the robot performed as it was expected. However it was unable to detect obstacles, which were placed in certain arrangements.

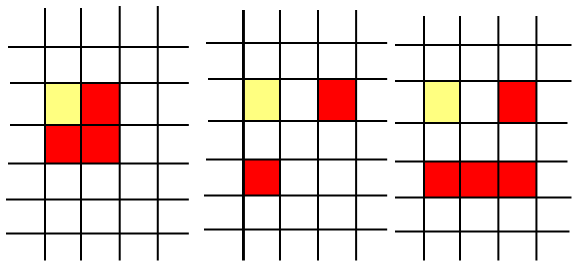


Fig.4. Obstacle arrangements with red signifying obstacles that are detected and yellow that is not, by the robot

The common thing amongst all these arrangements are that one of the obstacles remain hidden in orthogonal directions. It can be shown mathematically that such possible arrangements in a mxm matrix are of the form :

$$\sum_{1}^{m-1} i*(i+1) \text{ X } \sum_{1}^{m-1} I$$

And, total no. of arrangements is

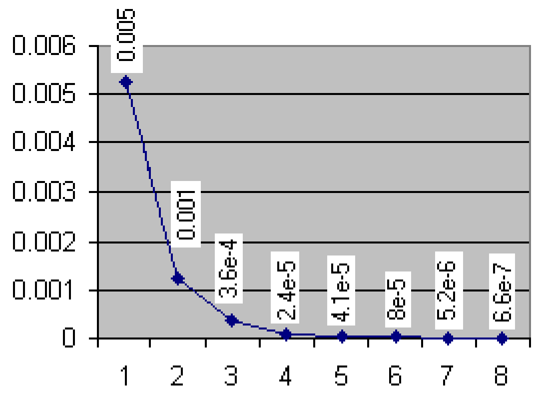
$$\prod_{x=0}^{p-1} (p-x) \qquad \text{,where p=K(const, no. of obstacles)}$$

For p=4, the following table is constructed

| Size | Undetected grids | Total grids | Ratio |
|---|---|---|---|
| 5x5 | 200 | 37950 | 0.00527 |
| 8x8 | 2352 | 1906128 | 0.00123 |
| 12x12 | 18876 | 51536628 | 3.66E-04 |
| 20x20 | 252700 | 3152219700 | 8.01E-05 |
| 25x25 | 780000 | 18890917500 | 4.13E-05 |
| 30x30 | 1955325 | 81466863075 | 2.40E-05 |
| 50x50 | 25510625 | 4.8711E+12 | 5.24E-06 |
| 100x100 | 824917500 | 1.24925E+15 | 6.60E-07 |

Table.1. Result of the simulator

From the above graph it can be shown that the ratio which denotes number of undetected robots, is very low. This can be neglected in real world scenario as the grid size would be much larger than 100x100. However this ratio can be made zero but making the no. of undetected obstacles=0 would make the algorithm complex and the time to traverse the entire region would increase exponentially.




Graph.1. Effect of undetection when size of the region increases

## 10 Conclusion

We have described a problem in obstacle detection and avoidance and outlined an approach towards its analysis and solution. The results obtained in this paper are a first step towards characterizing the problem. With a generalized version of Formula 4, we envisage an optimized covering algorithm for detection and avoidance that yields better results in less time.

## 11 References

**[1] Jeffreys, H. and Jeffreys, B. S.** "Lagrange's Interpolation Formula." §9.011 in Methods of Mathematical Physics, 3rd ed. Cambridge, England: Cambridge University Press, p. 260, 1988.

**[2] Whittaker, E. T. and Robinson, G.** "Lagrange's Formula of Interpolation." §17 in The Calculus of Observations: A Treatise on Numerical Mathematics, 4th ed. New York: Dover, pp. 28-30, 1967.