Generating Random Paths between Two Points in Space: Proposed Algorithm

Ahmad Tayyar
Department of Computer Science, Isra University, Amman, Jordan
ahmad_tayyar@iu.edu.jo

ABSTRACT

In this paper, a computerized method is proposed to generate a random path that consists of finite number of points in space, those points are randomly generated in number. They satisfy the following condition:

$$L(P_iP_n) < L(P_{i-1}P_n)$$

Where $L(P_iP_n)$ is the length of the segment (P_iP_n). This is a significant issue and should be taken into account for the importance of generating paths which cannot be predicted in advance, and every repeated generation gives new path in different way from previous ones. This contribution will be applicable to various fields, such as economic, engineering, militaries, and other fields of applied sciences.

Furthermore, the suggested method depends on random displacements. Therefore, we generate random displacement at the starting point to decrease the remaining distance between the actual point (P_i) and the target point (P_n). This procedure is repeated at the end of the previous displacement to reach a point with a certain distance to the target point (P_n).

The evaluation of the developed algorithm shows that the generated points are converged to the target point.

KEYWORDS

Random paths; graph theory; configuration space; military computing; mobile robots

1 INTRODUCTION

There is a possibility to generate many paths that can link two specific points in the space. The objective is to generate points that are converged to the target point. Therefore, the following condition should be considered:

$$L(P_{i-1}P_n) > L(P_iP_n)$$

For i = 1, 2, …, n-1

There is unlimited number of paths that satisfy the previous condition between P_0 and P_n.

Let P_0 and P_n be two points in space. Initially we generate a random displacement at the point P_0, we consider the end point of that displacement is P_1 if it was convenience. This means that this displacement converges to P_n. Otherwise we regenerate another random displacement.

We repeat this procedure at the point P_1 to obtain the next point P_2, and so on until we reach a point with a specific distance from P_n. We obtain a set of points P_i for i= 1, 2, …,n which forms the points of the random path that link between P_0 and P_n [1].

2 THE IMPORTANCE OF THIS PAPER

It can be clearly illustrated that the generated paths using the proposed method can be useful in many applications such as military and economy that presents confidential pathways with various costs and different segments. For example, this method can be applied for wireless sensor networks and Ad-hoc mobile networks at schools or universities. In this manner, it is possible to generate mobile centers as points between the initial mobile station and the final mobile station. Other examples include robot Path planning, building and developing the roads in cities [2, 3, 4].

3 BUILDING THE ALGORITHM

Let's assume that $P_0(x_0, y_0, z_0)$ and $P_n(x_n, y_n, z_n)$ are two specific points in the space. We want to connect them with discrete points that have a specific criteria.

The point P_0 can approach to the point P_n if it takes any position inside the sphere $S_0(P_n, P_0P_n)$ which is centered in P_n and its radius is P_0P_n.

For this reason we generate a random displacement at P_0 within the sphere S_0 and then consider P_1 as the end point of this displacement as shown in Figure 1. It also repeats the generation process at P_1 within a new sphere $S_1(P_n, P_1P_n)$, as a result, we obtain a new point P_2.We continue generating points until reaching a point with a specific distance from P_n. Consequently, a set of points are obtained: $P_1, P_2, \ldots, P_{n-1}$ which are located on the surfaces of the spheres centered in P_n: $S_1, S_2, S_3, \ldots S_{n-1}$ as shown in figure 1. Linking the points $P_0, P_1, \ldots, P_{n-1}, P_n$ gives us a path (P_0, P_n).

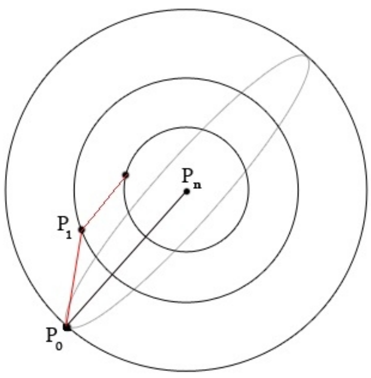

Figure 1. Generating points in Space

3.1 Square of Displacement

In order to simplify the process, we search the projection of the previous path on the plan [5]. Then our issue will become generating random paths within circles centered in P_n: $C_1, C_2, \ldots C_{n-1}$ which pass the points $P_1, P_2 \ldots, P_{n-1}$ respectively.

A point P may approach to P_n if it takes any position within the circle C centered in P_n with the radius PP_n as shown in figure 2.

To check the best way to generate random points with regular distribution in the plan starting from point P, we follow the following procedure:

x = x – L + random(2*L+1);
y = y – L + random(2*L+1);

As a result, this gives a random displacement within a square its side length (2L) and its center the point P(x, y).

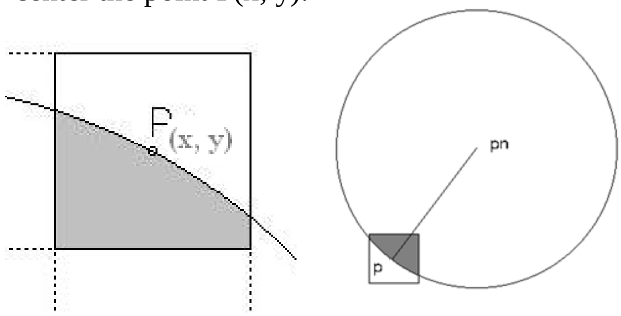

Figure 2

The probability of obtaining a displacement in the shaded area (Figure 2) will be acceptable.

We generate a random displacement at the point $P_0(x_0,y_0)$ within the square and point P_1 as the end point of that displacement if it was located within the circle C_0, which is indicated by the shaded area in the figure2. Otherwise we redo the procedure in the opposite case.

At P_1 we repeat the generation process within the circle C_1centered in P_n with the radius P_1P_n to obtain a new point P_2, and so on. We repeat points generation process until reaching a point with a specific distance from P_n. We will obtain the set of points $P_1, P_2, \ldots, P_{n-1}$.

The square that contains the movement is called the square of displacement. Its side length limits the distance value between the points that are generated to form the random path. Distance value between the points forming the random path increases by the increase of the square of displacement side length.

It should be notable that the generated points are becoming convergence toward P_n according to the proposed method. Thus, the generated points satisfy the following quality

$$L(P_{n-1}P_n) < L(P_{n-2}P_n) < \ldots . L(P_2P_n) < L(P_1P_n) < L(P_0P_n)$$

This sort of randomization in the displacements obtain various paths once we run this algorithm.

ISBN: 978-1-941968-04-8 ©2014 SDIWC

4 THE ALGORITHM

In the following, we present the algorithm that generates random path between two points as projection to plan.

The functionality of the proposed algorithm includes the following steps:

1. Enter the values x_0, y_0, x_n, y_n, L, d
2. Let P(x, y) be the current position, then $x=x_0$, $y=y_0$.
3. Calculate the distance L1(P, P_n) between the points P and P_n.
4. Generate a random displacement at the point P(x, y) within a square centered in P and its side length is 2L, so $x = x + \Delta x$ and $y = y + \Delta y$
5. Calculate the new distance L2(P, P_n)
6. If L2<L1 then link the points P and P_0 and then rename the new position by $P_0(x_0=x, y_0=y)$ and consider L1=L2.
7. If L2<d then stop.
8. Go to the step number 4.

The proposed algorithm has been implemented using C++ programming language since its efficiency and reliability as following:

```
void find(int d)
{float d1,d2;
  d1=pow(x1-x0,2)+pow(y1-y0,2);
  L=2+random(sqrt(d1)/2+1);
  do
  {x=x0-L+random(2*L+1);
    y=y0-L+random(2*L+1);
    d2=pow(x1-x,2)+pow(y1-y,2);
    if (d2<d1)
      {line(x0,getmaxy()-y0,
       x,getmaxy()-y);
       circle(x,480-y,4);
       x0=x;  y0=y; d1=d2;}
  }while (sqrt(d1)>d);
  line(x0,getmaxy()-y0,
   x1,getmaxy()-y1);}
```

5 EXAMPLES

In the following, we present six screenshots in Figure 4 showing two random paths in each of them for the same points P_0 and P_n. As a result, we have difference in the number of points and shape for each screen shot between the same points (P_0, P_n).

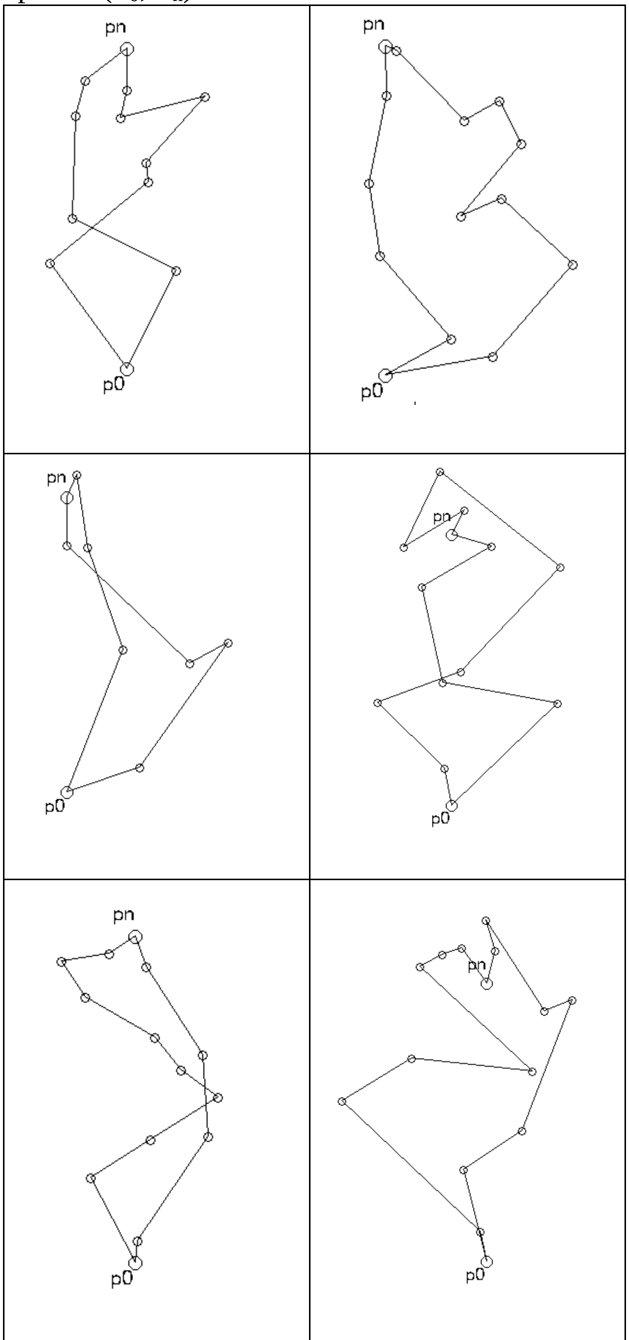

Figure 3. Example Screenshots

One of reasons in the different forms of paths is the random displacements. In addition, the value of square side is also an impressive factor in the different forms of paths.

6 CONCLUSIONS AND FUTURE WORK

We have shown the importance of generating a random path between two points (P_0, P_n). The proposed algorithm demonstrates the generation of random path in intelligent way. Such algorithm includes sequential related to various costs.

This study is applicable and valuable to several applications and networks such as WSN, Ad-hoc – mobile networks and telecommunication networks [6].

As a part of future work, this algorithm will be implemented in Matlab to calculate the weights of the best random path between two points. We are in progress applying this developed algorithm on an important issue which is the networks of water pipes against the leakage.

REFERENCES

[1] PEARSON, K. (1905). The Problem of the Random Walk. *Nature*, 72(1865), pp.294-294.

[2] Barraquand, J., Langlois, B. and Latombe, J. (1992). Numerical potential field techniques for robot path planning. *IEEE Trans. Syst., Man, Cybern.*, 22(2), pp.224-241.

[3] Amato, N. and Wu, Y. (1996). A randomized roadmap method for path and manipulation planning. *Proceedings of IEEE International Conference on Robotics and Automation*.

[4] Roth, A. and Vate, J. (1990). Random Paths to Stability in Two-Sided Matching. *Econometrica*, 58(6), p.1475.

[5] Carmo, M. (1976). *Differential geometry of curves and surfaces*. Upper Saddle River, N.J.: Prentice-Hall.

[6] Meguerdichian, S. and Koushanfar, F. (2001). Exposure in wireless Ad-Hoc sensor networks MobiCom. In: *7th annual international conference on Mobile computing and networking*. New York: ACM New York, pp.139-150.