K-Means derived strategized placement of starting points for parallel RRT's

by

K. Madhava Krishna

Report No: IIIT/TR/2016/-1

Centre for Robotics
International Institute of Information Technology
Hyderabad - 500 032, INDIA
February 2016

K-Means derived strategized placement of starting points for parallel RRT's

Krishna Gandhi[a], Gurshaant Singh Malik*[b] and K. Madhava Krishna[c]

[a]*Robotics Research Center, IIIT, Hyderabad, India*; [b]*Robotics Research Center, IIIT, Hyderabad, India*; [c]*Robotics Research Center, IIIT, Hyderabad, India*;

()

1. Introduction

Initial placement of RRT plays an important role when it comes down to improving performance in terms of uniform exploration of map. Algorithms like Distributed , K-Distributed for parallelizing of RRT have left the question of initial placement of RRT modules unanswered. Strategized placement of RRT modules can yield parametrically faster performance with a more homogeneous exploration as compared to designs where placement is done randomly. For $n > m$, m RRTs with strategized placement can perform better than n RRTs with non-strategized or random placement. This startegized placement of RRTs is achieved by clustering the map in m clusters using complex implementation of K-Means clustering algorithm which is sensitive to geometrical constraints of the map.

With our work, we propose strategized placement of starting points for RRT that can yield faster performance with a more homogeneous exploration.

2. Strategized placement

In a multi-module RRT design, an important issue is to decide the initial placement of RRT modules before the start of exploration. A naive way of placement of RRT modules before exploration is to randomly sample the free space and then placing these points as initial configuration start the exploration.

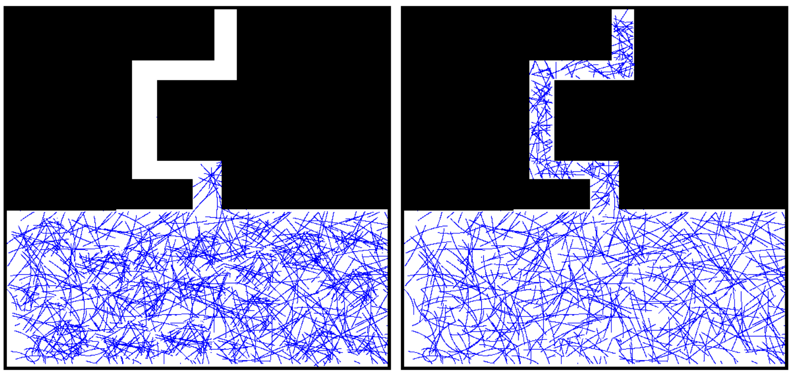

Figure 1. a.) Un-Strategized(Left) vs b.) Strategized exploration(Right)

*Corresponding author. Email: garymalik8080@gmail.com

During exploration, an RRT usually tends to grow towards less geometrically constrained spaces as compared to geometrically constrained or tight spaces. As a result less constrained areas will be explored more frequently and geometrically constrained areas will remain relatively unexplored. As shown in Figure. 1(a), randomly placed RRT modules will not take care of exploration of constrained spaces as efficiently as desired. Hence, it will not solve the problem of uniform exploration of map. Instead of randomly placed RRT modules in the free space, if the placement of RRT is done strategically, keeping complexity of the map in consideration, higher performance, in terms of uniformly exploring the entire map, including the constrained areas, in lesser time, can be achieved. It can be claimed with confidence that m un-strategized or randomly placed RRTs perform worse as compared n strategically placed RRTs where $m > n$. As shown in Fig. 1(b) the placement of RRTs is done strategically which leads to uniform exploration of map.

The method being proposed will take into consideration the complexity of map. It will parametrically calculate difficulty in exploring a particular area of map and then find the required best position to start exploration. The initial placement of staring points is done through the complex implementation of modified K-Means. This modified K-Means is implemented in MATLAB and the processed data is communicated to the FPGA for further exploration.

2.1 *Proposal*

The aim is to find centers or initial locations on map such that sum of distance from every free cell in the map to these centers is minimum. As shown in Fig. 2, a map is given which is free from obstacles. To initialise a single RRT, the best position would be from center of map as shown in Fig. 2(a), since the summation of weighted distance(explained in further sub-sections) from center to all other points is minimum. In a way we have taken the whole map as single cluster and treat the cluster center as RRT's starting point. Same we can extend for exploring the map with 2 and 3 RRTs in parallel as shown in Fig. 2(b),(c).

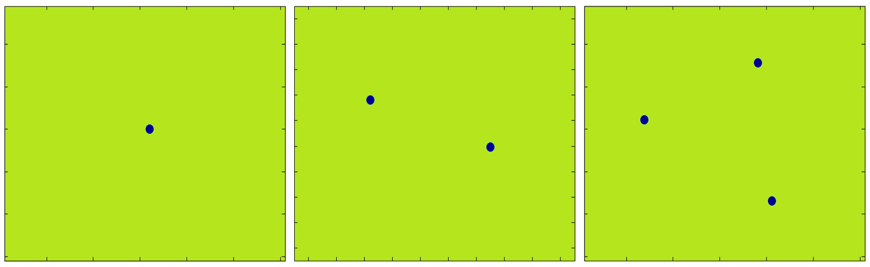

Figure 2. Strategized Location for 1, 2 and 3 RRTs respectively

To start a exploration with n RRTs working in parallel, n different strategized locations are needed. The method being proposed divides the free space of map into n clusters. All the points which belong to a particular cluster will be at minimum weighted distance from its cluster center as compared with other cluster centers. Since these n clusters will cover the map fully, every point in the map will be at minimum weighted distance from its cluster center. So the overall summation of weighted distances will be minimized, which fulfills our requirement. This overall summation of weighted distance is represented by a cost function U.

As shown in Fig. 3, p_1, p_2 and p_3 are the points of free space in map and a,b are the cluster centers.

Let $d_{x \to y}$ be the distance from cluster center to the free space points. $x \in \mathbb{C}lusterCenters$

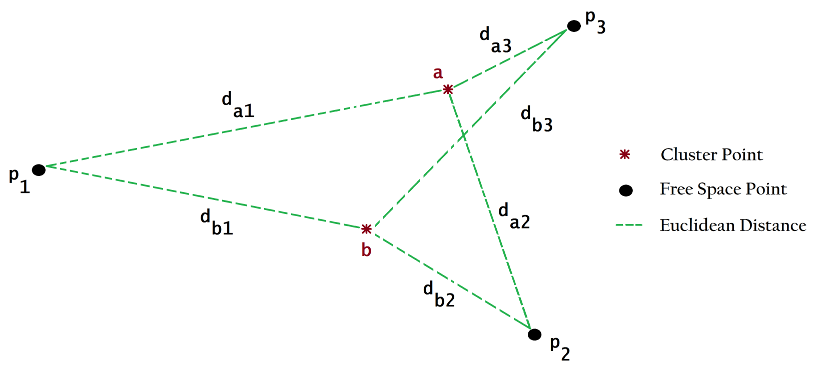

Figure 3. Cluster Points

and $y \in \mathbb{F}reespacepoints$. As we can refer to above Fig. 3

$$\begin{aligned} d_{b \to p_2} &< d_{a \to p_2} \quad \text{for point } p_2 \\ d_{b \to p_1} &< d_{a \to p_1} \quad \text{for point } p_1 \\ d_{a \to p_3} &< d_{b \to p_3} \quad \text{for point } p_3 \end{aligned}$$

So the cost function U for respective case will be

$$U = \arg\min((d_{b \to p_3}, d_{a \to p_3}) + (d_{b \to p_2}, d_{a \to p_2}) + (d_{b \to p_1}, d_{a \to p_1}))$$
$$U = \arg\min(d_{b \to p_3}, d_{a \to p_3}) + \arg\min(d_{b \to p_2}, d_{a \to p_2}) + \arg\min(d_{b \to p_1}, d_{a \to p_1})$$
$$U = d_{a \to p_3} + d_{b \to p_2} + d_{b \to p_1}$$

So extending this method for a general case. Let

$$X = (x_1, x_2, ...x_n) \quad \text{Set of free space points}$$
$$V = (v_1, v_2, ...x_c) \quad \text{Set of cluster centers}$$

where x_i = Column vector of Co-ordinates of i^{th} free space point and n is the total number of free space points.
v_i = Column vector of Co-ordinates of i^{th} cluster center and c is the total number of cluster.
So the overall cost function comes out to be

$$U = \sum_{i=1}^{c} \sum_{j=1}^{c_i} | \, | \, x_j - v_i \, | \, |^2 \tag{1}$$

$$U = \arg\min \sum_{i=1}^{c} \sum_{j=1}^{c_i} | \, | \, x_j - v_i \, | \, |^2 \tag{2}$$

Here c_i is the total number of free space points in cluster i.

2.1.1 Cost function U minimization

The cost function U is minimized using K-means clustering algorithm. Here free space are given as data points. The algorithm uses a local search approach to partition the points into c clusters. A set of c initial cluster centers is chosen arbitrarily. Each point is then assigned to the center closest to it. The centers v_i are again recomputed as centers of mass of their assigned points.

$$v_i = \frac{\sum_{j=1}^{c_i} x_i}{c_i} \tag{3}$$

It's an iterative process and repeated until the process stabilizes and shifting of centers stops.

2.1.2 Path Correction

From the cost function equation 2, the term $|\,|\, x_j - v_i \,|\,|^2$ represents the Euclidean distance between free space point x_j and cluster point v_i. So by taking euclidean distance, it is assumed that there is a direct path between x_j and v_i. But there can be a possibility of not having a direct path between two points in map due to obstacles or configuration of map itself. As shown in Fig. 4 below, the two points x_j and v_i they don't have a direct path connecting both. So to

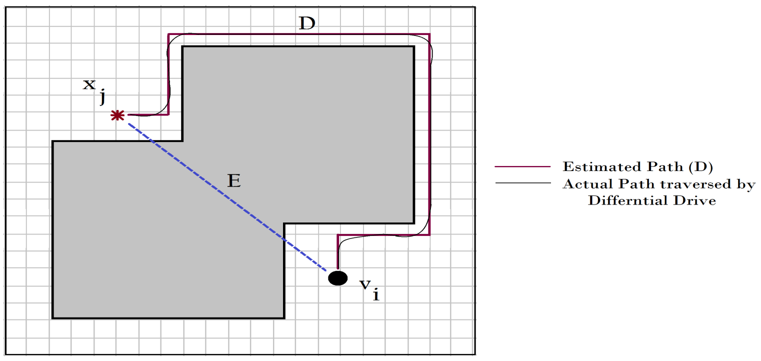

Figure 4. Path Correction

take care of such cases, instead of taking euclidean distance E, shortest path D possible between x_j and v_i is taken into account. It is assumed that this shortest path D is more or less equal to path traversed by differential drive robot. Using shortest path D in cost function equation 2 will give much better estimation of cluster centers and also includes the complexity of map. This distance D is calculated using Breadth First Search (BFS) and the obstacle avoided path is also stored in a custom stack for further operation.

2.1.3 Weighing cells

During the exploration RRT tends to move towards geometrically less constrained areas as compared to constrained areas. In geometrically less constrained spaces, the probability that RRT will get a kinematic path successfully to the sampled node with proper avoidance of obstacles is very high. Opposite is the case for geometrically constrained spaces, probability of getting a successful kinematic path is very low. Hence the density of nodes in less constrained areas as compared with tight spaces becomes high, which result in non uniform exploration of map. One of way to overcome this issue is to give more priority to geometrically constrained areas while clustering.

So to resolve this issue we propose a weighing scheme, in which a weight is assign to every free cell. This weight is going to be proportional to the geometrical constraint of that respective cell. So higher the weight, more constrained the cell is and vice versa. The approach of calculating cell's weight is very similar to finding potential field of a point surrounded by electrostatic charges. Higher the electrostatic potential of given point more is the difficulty for charge to pass through that point. In this case, as shown in Fig. 5, obstacles in map will behave like charges and only those charges will contribute to field which are in line of sight.

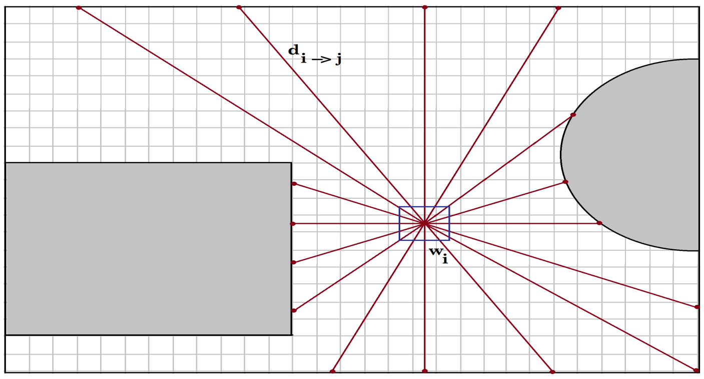

Figure 5. Weighing Cell

As shown in above Fig. 5, the weight of cell i is w_i is contributed to by all the obstacles, boundaries and tight spaces around it. To compute this weight, from every cell, n rays are projected, spaced equally at an angle of $2\pi/n$. As a particular ray hits an obstacle or boundary first in its line of sight, euclidean distance is computed from that cell i to the point of ray hitting first. So the weight due ray j on cell i will be w_{ij}.

$$w_{ij} \propto \frac{1}{d_{i \to j}} \tag{4}$$

Here $d_{i \to j}$ = Euclidean distance between cell i and point of first hit of ray j
So the total weight w_i of cell i will be sum total of w_{ij} for all the n rays.

$$w_i = \sum_{j=1}^{n} w_{ij} \propto \sum_{j=1}^{n} \frac{1}{d_{i \to j}} \tag{5}$$

$$w_i = \sum_{j=1}^{n} \frac{k}{d_{i \to j}} \tag{6}$$

Here k is a constant taken as 1, it is a constant multiplicative factor.

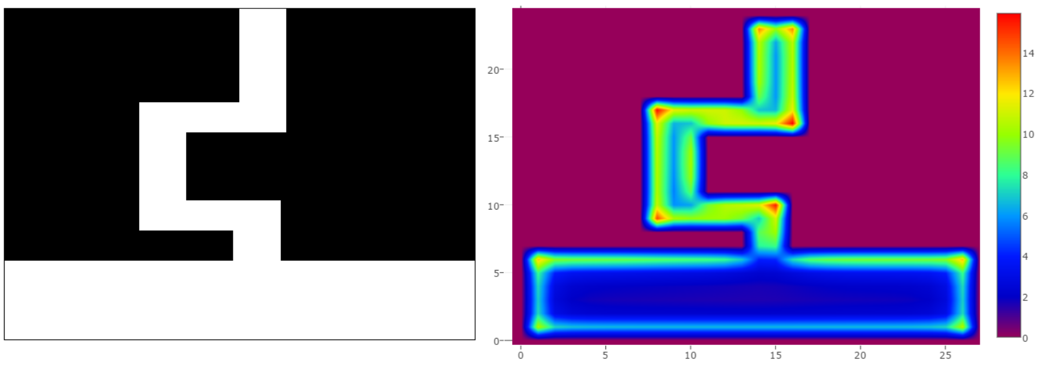

Figure 6. Weighing Cell's Heat Map

Above Fig. 15 depicts the heat map of the weight calculated for a particular map. The yellow and red colour represents the geometrically constraint area in the map while blue represents cold or less constraint area.

2.1.4 Final Weighted Path

In the last section we discussed about weight of a particular cell which tells us about the difficulty to traverse through it. As shown in the Fig. 7 ,in both the maps, traversal needs to happen from point c to point p. The shortest path D is same in both maps. But in Fig. 7(b), difficulty in traversing is much more as the weight of cell are much higher due to high geometrical constraints.

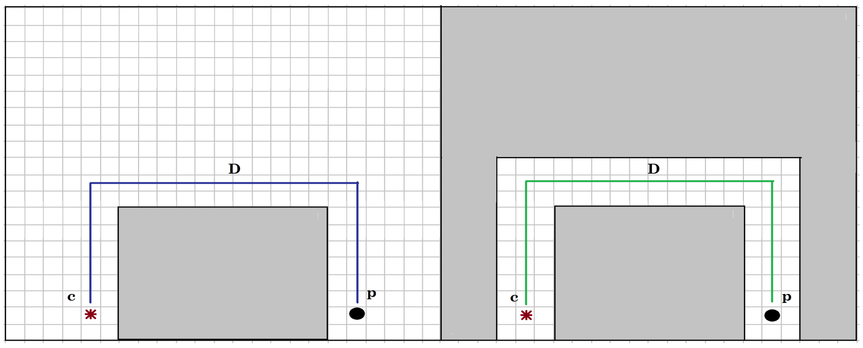

Figure 7. Weighted Path

So to account this difficulty or complexity of map into consideration, weighted shortest path is used, instead of just shortest path. So the effective weighted shortest distance will be summation of all the cell weight that came in the path.

$$W_{c \to p} = \sum_{j=1}^{q} w_i \tag{7}$$

Here q is number of cell came in between path from c to p and w_i is weight of respective cell.

2.1.5 Weighted Cost function

Initially the cost function U did not consider geometrical constraint and complexity of map. But as discussed in last section, the final weighted path will take care for complexity of map. Hence from 2 if we replace $|\,|\, x_j - v_i \,|\,|^2$ by weighted path $W_{x_j \to v_i}$, the new generated cluster will take care of complexity of map. So the final cost function U comes out to be

$$U = \arg\min \sum_{i=1}^{c} \sum_{j=1}^{c_i} W_{x_j \to v_i} \tag{8}$$

Fig. 8 given below will show the difference in all three strategies for a particular map.

As we can see from Fig. 8(a) RRTs are placed randomly as compared with Fig. 8(b) where RRTs are placed using 2 as cost function. So the complexity of map is not considered in RRTs placement. While in Fig. 8(c) placement is done using 8 cost function in which complexity of map is taken care by using weighted path.

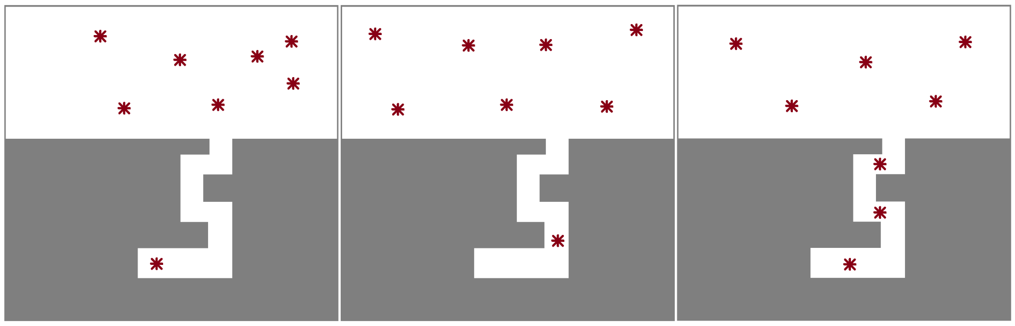

Figure 8. Initial Placement of RRTs across 3 different Strategies