Path planning for Unmanned Ground Vehicle

Fethi DEMIM[1], Kahina LOUADJ[2,3], Abdelkrim NEMRA [1]

***Abstract*— The aim of this paper is to find the path planning of an Unmanned Ground Vehicle (UGV) using shortest path algorithm that is a path of a string sequence of n vertices. Before that, the estimated point which has chosen is thought near real points with two experiences. These estimated points are chosen as the points which its minimum distance equal between 0.4m and 0.5m respectively. The goal is to calculate a path between the vertices of a graph that minimizes the robot path of estimated points. It can explain by finding the shortest path between two points. The main contribution of this work is seeking the shortest path of these points and minimizes the computational time and the energy that the robot will be provided along this path in minimal distance.**

I. Introduction

Path planning is an important primitive for autonomous mobile robots. Furthermore, lets to the robots find the shortest or otherwise optimal path between two points. Otherwise optimal paths could be paths that minimize the amount of turning, the amount of braking or whatever a specific application requires.

The shortest path is one of the most known problems and brings a lot of interest in Early history of shortest paths algorithms which are presented by Shimbel in 1955. Information networks. Ford (1956), RAND, economics of transportation, Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, Seitz (1957), Combat Development Dept. of the Army Electronic Proving Ground. Dantzig (1958)[2], Simplex method for linear programming, Bellman (1958)[1]. Dynamic programming. Moore (1959), Routing long-distance telephone calls for Bell Labs, Dijkstra (1959)[2], Simpler and faster version of Ford's algorithm.

The Shortest path problem is to find a distance between two vertices Shortest path problem is about finding a path between two vertices in a graph such that the total sum of the edge weights is minimized.

This problem could be solved easily when using all edge weights, but here the weights can take any value. The use of algorithms which find the shortest paths are important and not only in robotics, but also in network routing, video games and gene sequencing. The shortest path problems are by far the most fundamental and also most commonly encountered problems in the study of transportation and communication networks.

Finding the shortest path in a road network is a well known problem. Various proven static algorithms such as Dijkstra are extensively evaluated and implemented. When confronted with dynamic costs, such as link travel time predictions, alternative route planning algorithms have to be applied.

Shortest path algorithms are applied to automatically find directions between physical locations, such as driving directions on web mapping websites like Map Quest or Google Maps. For this application fast specialized algorithms are available.

If one represents a nondeterministic abstract machine as a graph where vertices describe states and edges describe possible transitions, shortest path algorithms can be used to find an optimal sequence of choices to reach a certain goal state, or to establish lower bounds on the time needed to reach a given state. For example, if vertices represent the states of a puzzle like a Rubik's Cube and each directed edge corresponds to a single move or turn, shortest path algorithms can be used to find a solution that uses the minimum possible number of moves.

In a networking or telecommunications mind set, this shortest path problem is sometimes called the min-delay path problem and usually tied with a wider path problem. For example, the algorithm may seek the shortest (min-delay) widest path, or widest shortest (min-delay) path, often studied in operations research, include plant and facility layout, robotics, transportation, and VLSI design. In our paper; before founding Shortest path, we use the Data science [10], to extract of knowledge from data. This domain, it incorporates varying elements and builds on techniques and theories from many fields, including signal processing, mathematics, probability models, machine learning, statistical learning, computer programming, data engineering, pattern recognition and learning, visualization, uncertainty modeling, data warehousing, and high performance computing. Although use of the term data science has exploded in business environments.

In our paper; before the founding shortest path, we use the data science [10], to extract of knowledge from data. Before that, the estimated point we have chosen is the points we thought near real points of two experiences. These estimates point, we choose the points whose distance minimum is 0.5m, and 0.4m. And the points we have sorted and judge are reliable points for our goal, we will seek the shortest path of these points.

And this result aims to minimize the travel time, and we will minimize the energy that the robot will provide along this path in minimal distance.

The paper is structured as follows : we begin with some definitions that use in our work. The Diksjtra algorithm is given in section II. In section III, we applied this algorithm for the robot. We finished by conclusion in section IV.

II. Essential Definitions

Definition 2.1: A directed graph $G = (V, E)$ consists of a finite set V of nodes and a finite set E of arcs. We will use n and e to denote the number of elements in V and E, respectively.

The node set is assumed to have elements $\{1, 2, ..., n\}$.

An arc a in E is an ordered pair (u, u) of nodes.

Definition 2.2: A path is a finite sequence of arcs $P = (a_l, a_2, ..., a_k)$, such that a_i starts where a_{i-l} ends, for $i = 2, ..., k$.

Definition 2.3: The path length (or simply length) of P is defined to be $d(P) = l(a_l)+....+l(a_k)$. If $a_i = u_i)$ for $i = 1, ..., k$, the path P is also denoted by the node sequence $(u_0, u_l, ..., u_k)$; u_0 is called the source (or origin) of P, u_k is called the destination of P, and P is called a path from u_0 to u_k.

P is simple if $u_f \neq u_l$ for all $i \neq j$.

*F.Demim, K.Louadj, A.Nemra

[1] Laboratoire Robotique et Productique, Ecole Militaire Polytechnique, Bordj El Bahri,BP 17, 16111, Alger. Algérie. demifethi@gmail.com, karim_nemra@yahoo.fr

[2] Université de Bouira. Dpartement de Mathmatiques. Bouira. Algérie. louadj_kahina@yahoo.fr

978-1-5386-5065-3/18/$31.00 ©2018 IEEE

Definition 2.4: P is a shortest path from u to u if $d(P)$ is the minimum length of any path from u to u; in this case, $d(P)$ is the shortest distance from u to u. The shortest distance matrix is an $n \times n$ matrix whose $(u, u)th$ entry is the shortest distance from u to u.

III. Dijkstra Algorithm

Dijkstra's algorithm one is faster than other Belleman-Ford algorithm but is restricted to nonnegative length functions. The general framework of the method is the following scheme, described in this general form by Ford [1956].

◇ Initially, set $d(s) := 0$ and $d(v) := \infty$ for each $v \neq s$.
♭ Next, iteratively,
choose an arc (u, v) with $d(v) > d(u) + l(u, v)$
and reset $d(v) := d(u) + l(u, v)$.
♮ If no such arc exists, d is the distance function.
♮ The difference in the methods is the rule by which the arc (u, v) with $d(v) > d(u) + l(u, v)$ is chosen.

Dijkstra's Algorithm prescribes to choose an arc (u, v) with $d(u)$ smallest (then each arc is chosen at most once, if the lengths are nonnegative). This was described by Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz [1957] and Dijkstra [1959]. A related method, but slightly slower than Dijkstra's method when implemented, was given by Dantzig [1958], and chooses an arc (u, v) with $d(u) + l(u, v)$ smallest.

IV. Application of an Unmanned Ground Vehicle

When one tries to get from one point to another in a network, looking for the shortest path, that is to say the one whose distance is the smallest. If the number of possibilities Paths between the starting point and the end point is small, it is sufficient to calculate the length of each of the paths by summing the length of the links which compose it and to directly compare the lengths obtained.
But such an exhaustive solution quickly becomes impractical if the number of possible paths is large. Fortunately, there are algorithms that exist to have to calculate all possible projects. For this purpose, we implemented a various strategies. The composed of nodes are represented by a graph (the vertices of the graph) and the Oriented links (the arcs of the graph).
Dijkstra's algorithm depends on the principle of "exploration from the best". For example, from the best predecessor visited, it can be used when all arcs have a non-negative value comparing to what is expended on a given path. In this case, there can be no absorbing circuits, but there may be other circuits, such as two-way links. In practice, this algorithm is very often used to solve routing problems in telecommunications networks. We consider 12 estimated points given in table 1, in two cases, firstly in 0.5m, secondly 0.4m.

Let calculate Euclidean distance between the true position and an estimated position of two experiments. After that, in the first experiment, we chose the positions which their distances are lower than 0.5m and 0.4 m in the second experiment.

Minimal distance in second experiment is 24.6214 m. And path corresponding to this distance is 1, 3, 4, 5.

Minimal distance in second experiment is 6.3856m. And path corresponding to this distance is 1, 5, 3, 6.
In figure 1 and figure 3, we present nodes of Graph for estimating points for first experiment and second experiment respectively. In figure 2 and 4, its corresponding shortest path of Graph 1 and 2 respectively, its presented in red color.

True Position	Estimated Position $(x(m),\ y(m)$	
(x, y)	First experiment	Second experiment
(0,0)	(0,0)	(0,0)
(4.90;0.70)	(4.87;0.85)	(4.84;0.75)
(7.20 ; 1.61)	(7.16; 1.81)	(7.04 ; 1.98)
(7.77; 2.80)	(7.77 ; 3.03)	(7.43 ; 3.22)
(6.13 ; 4.54)	(6.19; 4.77)	(5.43 ; 4.57)
(5.80; 6.16)	(5.93 ; 6.33)	(4.73 ; 6.01)
(8.52; 8.27)	(8.67 ; 8.41)	(7.15 ; 8.62)
(8.53 ; 10.33)	(8.75 ; 10.39)	(6.64 ; 10.73)
(-3.35; 8.82)	(-3.17; 9.28)	(-4.73; 6.52)
(-3.55 ; 2.36)	(-3.92; 2.73)	(-4.28 ; 0.37)
(-3.36 ; 0.47)	(-3.85 ; 0.74)	(-3.86 ; -1.53)
(0.08 ; 0.09)	(-0.38; 0.26)	(0.09 ; 0.02)

TABLE I
TRUE AND ESTIMATED POSITIONS FOR FIRST AND SECOND EXPERIMENTS

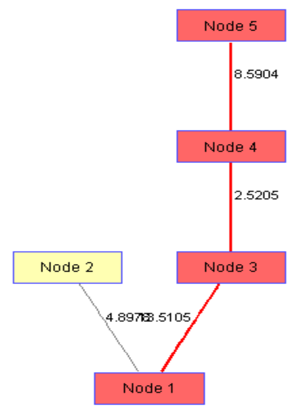

Fig. 1. Nodes of the first experiment

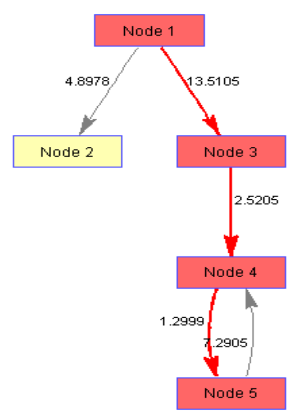

Fig. 2. Corresponding Shortest Path of the first experiment

V. Conclusion

In this article, we have found path planning of an Unmanned Ground Vehicle uses the shortest path algorithm in two experiences. Our results gave an objective satisfaction.
It consists in finding how the robot moves in an environment

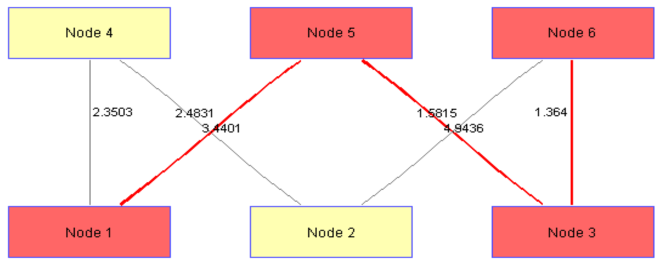

Fig. 3. Nodes of the second experiment

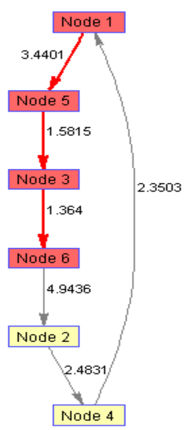

Fig. 4. Corresponding Shortest Path of the second Experiment

between a starting point and an arrival point taking into account different constraints. In the first experiment, the goal was allowed to go from point 1 to point 6. The second experiment is going from the point 1 to point 6. So, at the minimum distance we can find our robot point quickly if the robot is lost in the map, but it has a map with intermediate point distances between them in its environment. This can be interpreted by finding where it is the robot positioned at and to start from and to make more educated vehicle about which point is the best.

By comparing two experiences, we can say that the robot shortest path visits some points that are minimal distance during the search operation. Hence the robot shortest path algorithm is more efficient, faster and gains the energy source from the beginning to the final point.

REFERENCES

[1] E. Bellman. *On a routing problem,* Quarterly of Applied Mathematics, 16 (1), pp. 87-90, 1958
[2] E.W. Dijkstra, *A note on two problems in connection with graphs*. Numerical Mathematics, 1 (1), pp. 269-271, 1959
[3] S. Dreyfus *An appraisal of some shortest path algorithms*, Operations Research, 17 (3), pp. 395-412, 1969.
[4] R.W. Floyd. *Algorithm-97-shortest path*. Communications of the ACM, 5 (6), p. 345, 1962.
[5] H. Frank. *Shortest paths in probability graphs*. Operations Research, 17 (4), pp. 583-599, 1969.
[6] R. Hall. *The fastest path through a network with random time-dependent travel time*. Transportation Science, 20 (3), pp. 182-188, 1986.
[7] P. Loui. *Optimal paths in graphs with stochastic or multidimensional weights*. Communications of the ACM, 26 (9), pp. 670-676, 1983.
[8] P.B. Mirchandani. *Shortest distance and reliability of probabilistic networks*. Computers & Operations Research, 3 (4), pp. 347-676, 1976.
[9] D. Dubois, H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
[10] Lynda Tamine-Lechani, M.Boughanem, M.Daoud, *Evaluation of contextual information retrieval effectiveness*, overview of issues and research, Knowledge and Information Systems, 24(1), pp.1-34, 2010.