Global Path Planning on Uneven Elevation Maps

Sunglok Choi, Jaehyun Park, Eulgyoon Lim, and Wonpil Yu

Robot and Cognitive Convergence Research Dept., ETRI, Republic of Korea
(E-mail: sunglok@etri.re.kr, Web: http://sites.google.com/site/sunglok)

Abstract - This paper introduce about graph-search based global path planning on uneven elevation maps. An elevation map is an efficient and popular representation for 3-D terrains due to its easy manipulation by a computer. On the elevation map, we investigate three different optimal paths in the aspects of travel distance, travel time, and energy consumption. A distance/time-optimal path is derived by simple extension of A* on 2-D grid maps. A formulation on energy consumption leads an energy-optimal version and traversibility criteria. We demonstrate effectiveness of our proposed method by experiments on randomly generated Gaussian hills.

Keywords - Path Planning, Elevation Map, A*, Time-optimal, Distance-optimal, Energy-optimal

1. Introduction

Global path planning is a process to generate a collision-free path from the start to the goal on the given map. It is one of the most fundamental and inevitable tasks for autonomous navigation. Due to such importance, there have been a number of works on global path planning from beginning of robotics. These works mostly solved the path-finding problem by modeling traversable space (or states) as graphs. For example, a visibility graph, Voronoi diagram, state lattice, roadmap, and grid map are kinds of graphs describing connectivity of the given space. A* and their variants are popular and effective tools for searching a path on the given graphs. Sometimes it is difficult to represent the given configuration space (c-space) as a complete and dense graph because the c-space is high-dimensional or continuous with complex constraints. A solution to overcome such difficulty is building a partial and random graph by sampling sufficient number of feasible states on the c-space. Probabilistic roadmap method (PRM) and rapidly-exploring random tree (RRT) are representative examples.

The world model (a.k.a. an appearance map) is important for global path planning to understand traversability and connectivity of the given space. The model can be classified by its types and dimension of description format. Appearance of 2-D space is described by a grid or vector map similarly to a raster and vector image in graphics. In robotics, a grid map is favored because it is easy for a computer to access and manipulate (e.g. build and update). Each cell on a grid map can have its value of obstacle occupancy (a.k.a. an occupancy grid map) or traversal cost (a.k.a. a costmap). Some robotic appli-

This work was supported partly by the R&D program of MKE and KIAT. (Project: 3-D Perception and Robot Navigation Technology for Unstructured Environments, M002300090)
maps. We extend this basic A* to optimal path planning on elevation maps. Distance/time-optimal A* on elevation maps is simply derived from the 2-D version of A* by extending its dimension. Energy-optimal A* is derived based on energy consumption by the robot’s traversal. The energy consumption is calculated through gravity, friction, drag, and energy loss. We also propose a way to check traversability of cells on elevation maps, which is our second contribution. We performed experiments on random Gaussian hills to verify effectiveness of the proposed path planning, which is demonstrated in Section 4.

2. Problem Formulation

Our problem is to find an optimal path from the given start to the goal on an uneven elevation map. The optimality is quantified by physically meaningful measures such as travel distance, travel time, and energy consumption. As shown in Figure 1, energy criteria is anisotropic according to slope and heading even though travel distance is same, so an energy-optimal path might be also meaningful in some aspects. An elevation map is a kind of grid maps whose cells additionally have their values of ground level at their location. The value of elevation for a cell $n$ is written by $n.z$.

Assumptions We assume four conditions to simplify the problem. First, the given robot is holonomic so that it can change its heading on a cell without translation. An omni-directional mobile robot satisfies this condition. A differential drive, the most popular one in robotics, also roughly follows it because its minimal radius-of-rotation is zero. Second, we assume that the robot is a point on the given map. This assumption makes the problem free from complex situations such as collision detection and multiple ground contacts. These two conditions are not realistic but common and efficient approximation for many global path planning, especially graph-search based approaches. Third, the robot moves with constant linear velocity, $v_c$. Since travel distance and time are proportional under this assumption, a distance-optimal path becomes a time-optimal path. Finally, we assume that configuration and parameters of the robot and its operating space are known and constant. In details, the robot has constant mass $m$. Energy loss may be happened due to energy transfer and internal consumption, whose value per second is written in $P_l$. Originally it is a non-linear function on load, velocity, and others [6], but we can roughly regard it as constant because many parameters are fixed. The robot can exhibit the maximal power $P_{max}$ and its corresponding maximal driving force will be noted in $F_{max}$. The maximal driving force is also derived by $F_{max} = (P_{max} - P_l)/v_c$. Moreover, we need to consider three external forces affected on the line of motion, gravity $F_g$, friction force $F_f$, and drag force $F_d$, which are described by

$$F_g(\theta) = mg \sin \theta,$$

$$F_f(\theta) = \mu mg \cos \theta,$$

$$F_d = 0.5 \rho v_c^2 c_d A,$$

where $\theta$ is the inclined angle from the horizontal, $g$ is the gravitational field strength, $\mu$ is the friction coefficient, $\rho$ is the density of air, $c_d$ is the drag coefficient, and $A$ is the cross-sectional area of the robot. Figure 2 presents these forces according to configurations of the robot. Shortly we will write the overall power consumption by gravity, friction, drag, and energy loss as

$$P_{con}(\theta) = (F_g(\theta) + F_f(\theta) + F_d) v_c + P_l.$$  (4)

Since the inclined angle $\theta$ is positive when the robot comes up and negative when the robot comes down, so the gravitational force parallel to the incline, $F_g$, always has correct signs regardless of the robot’s moving direction.

3. A* Path Planning on Elevation Maps

3.1 A* Path Planning

A* is a graph-search algorithm to find the least-cost path from the start node $n_s$ to the goal node $n_g$. From the start node $n_s$, A* sequentially explores a visitable node¹ which can probably entail the least-cost path along the node. Such sequential exploration is terminated when A* reaches the goal node $n_g$. A fitness function quantifies the probable degree of being the least-cost path along a node $n$. It is defined by sum of a cost function $g$ and heuristic function $h$ as follows:

$$f(n) = g(n_s, n) + h(n, n_g).$$  (5)

A cost function $g(n_s, n)$ is (sequentially driven) traveled cost from $n_s$ to the query node $n$, and a heuristic function $h(n, n_g)$ is estimated cost from $n$ to $n_g$. The traveled cost is calculated by accumulating each edge-cost on a path from $n_s$ like

$$g(n_s, n) = g(n_s, p(n)) + c(p(n), n),$$  (6)

¹A node in OPEN priority queue
where $c$ is an edge-cost from its parent node $p(n)$. The parent node is selected as one of already visited neighborhood nodes\textsuperscript{2} which can make the child node $n$ have the least traveled cost from $n_r$. To generate the least-cost path, the heuristic should be less than equal to the minimum of all feasible traveled cost from $n$ to $n_r$, which is known as the admissible condition.

### 3.2 A* Path Planning on 2-D Grid Maps

A 2-D grid map is one of the simplest world representations for indoor and mild-terrain outdoor environments. Each cell is located at $(n_x, n_y)$ and contains information on obstacle occupancy or traversal cost.

A* and its variants are popularly applied to a 2-D grid map for goal-directed navigation. In the view of graph-search, non-occupied cells are nodes of a graph, and they are connected to their non-occupied adjacent cells with their respective edge-costs. When all cells have a same value of traversal cost (a.k.a. a uniform costmap), their edge-cost is usually defined by distance like

$$c_U(n_p, n_c) = d(n_p, n_c) \text{ where } (7)$$

$$d(n_p, n_c) = \sqrt{(n_c.x - n_p.x)^2 + (n_c.y - n_p.y)^2}. \quad (8)$$

Since a child cell and its parent are adjacent, the distance is usually $\delta$ or $\sqrt{2}\delta$ where $\delta$ is cell size. If each cell has a different value of traversal cost (a.k.a. a non-uniform costmap), its edge-cost should take into account its traversal cost like

$$c_N(n_p, n_c) = \frac{n_p.x + n_c.x}{2} d(n_p, n_c), \quad (9)$$

where $n.I$ is traversal cost of $n$, which means the degree of difficulty to move across the cell. One of popular heuristic functions is Euclidean distance,

$$h_U(n, n_e) = d(n, n_e), \quad (10)$$

which satisfy the admissible condition on the flat world. Even on a non-uniform costmap, we assume the expected traversal cost as 1 because it is difficult to estimate the expectation ahead, that is $h_U(n, n_e) = h_U(n, n_e)$. Moreover, it needs be careful not to violate the admissible condition because it can be broken when traversal cost of cells is less than 1.

### 3.3 Traversability on 2.5-D Elevation Maps

A 2.5-D elevation map is a simple extension of grid map whose cells contain additional information on their ground levels. Due to such augmentation, it is possible to describe uneven terrains such as bumps, ramps, hills, and valleys.

A* need to know that it is traversable or not from the parent cell $n_p$ to its child $n_c$ in order to expand its search. On a 2-D grid map, we can know such traversability directly from a value of occupancy or traversal cost\textsuperscript{3}. However, we need to derive the traversability based on the physical constraints on a 2.5-D elevation map. First, the robot can move if the required power becomes less than equal to its maximal power, $P_{con} \leq P_{max}$. In the view of forces, this condition becomes

$$F_g + F_f + F_d \leq P_{max} \quad (11)$$

Second, the robot unintentionally slides down ‘steep’ downhill because its gravity strongly pushes it down. Moreover, the robot may be accelerated during its sliding, which means this situation breaks our third assumption in Section 2. It might be possible for the robot to navigate on the steep downhill by reverse driving forces such as backward spinning wheels and fans. We will not consider this solution because it is sometimes uncontrollable and difficult to consider its complex and nonlinear phenomenon in global path planning. In the conservative aspect, we define this situation as intraversable and lead additional condition like

$$F_g + F_f^* + F_d > 0 \quad (12)$$

\textsuperscript{2}A node in OPEN or CLOSE list

\textsuperscript{3}A cell is not traversable when it has occupancy value more than the threshold or has an infinite cost value.
where \( F^*_f \) is the maximal friction force, probably static friction.

For example, these two conditions become much simpler inequality like

\[
-\mu_s < \tan \theta(n_p, n_c) = \frac{\Delta(n_p, n_c)}{d(n_p, n_c)} \leq \mu_s - \mu_r
\]

in the internal torque and rolling configuration shown in Figure 2 (b) with no drag force. This inequality is well matched with our intuition that the robot cannot climb on steep uphills and becomes uncontrollable on severe downhills. This configuration also has a very interesting threshold that the robot needs no driving force, \( F_g + F_f + F_d = 0 \). The threshold is at \( \tan \theta(n_p, n_c) = -\mu_s \) when there is no drag force. The robot will move with ideal rolling when the slope is more than the threshold. However, it suffers from sliding when the slope is less than the threshold, so the robot needs to hold and release its wheels frequently to keep the constant velocity. Therefore, the friction coefficient is varied like

\[
\mu(\theta) = \begin{cases} 
\mu_s & \text{if } -\mu_s < \tan \theta < -\mu_r \\
\mu_r & \text{if } -\mu_r \leq \tan \theta \leq \mu_s - \mu_r 
\end{cases}
\]

which is briefly shown in Figure 3.

3.4 A* Path Planning on 2.5-D Elevation Maps

We apply A* to an elevation map with additional consideration on elevation changes. We follow the basic of A* described in the previous subsections. For a distance-optimal path, the edge-cost should take into account not only horizontal distance, but also vertical distance. The distance-optimal edge-cost is derived as

\[
c_D(n_p, n_c) = s(n_p, n_c) \quad \text{where}
\]

\[
s(n_p, n_c) = \sqrt{d(n_p, n_c)^2 + \Delta(n_p, n_c)^2} \quad \text{and}
\]

\[
\Delta(n_p, n_c) = n_c - z - n_p - z,
\]

which is three-dimensional distance between two cells considering elevation difference \( \Delta \). As mentioned in Section 2, the distance-optimal edge-cost is same with the time-optimal edge-cost,

\[
c_T(n_p, n_c) = c_D(n_p, n_c),
\]

because we assume that the robot has constant linear velocity. Similarly, for an energy-optimal path, the edge-cost should be defined by required energy to move between two cells as follows:

\[
c_E(n_p, n_c) = \max \left( P_{con} (\theta(n_p, n_c)) \frac{s(n_p, n_c)}{v_c}, 0 \right). \tag{19}
\]

We ignore kinetic energy of the robot in the edge-cost because it is constant all the time due to constant velocity assumption. The edge cost \( c_E \) is limited to more than equal to 0 since \( P_{con} \) can be negative when the node \( n_c \) is at a deep valley, \( \Delta \ll 0 \). Specifically, in the internal torque and rolling configuration shown in Figure 2 (b) with \( F_d = 0 \), it is possible to lead much simpler form like

\[
c_E(n_p, n_c) = \max \left( mg(\Delta(n_p, n_c) + \mu(\theta(n_p, n_c)) d(n_p, n_c)) + P_{con} (\theta(n_p, n_c)) s(n_p, n_c/v_c), 0 \right). \tag{20}
\]

Finally we can define heuristic functions for distance/time-optimal and energy-optimal paths as follows:

\[
h_{D}(n, n_g) = h_T(n, n_g) = s(n, n_g) \tag{21}
\]

\[
h_{E}(n, n_g) = \max \left( P_{con} (\theta(n, n_g)) \frac{s(n, n_g)}{v_c}, 0 \right). \tag{22}
\]

4. Experiments

4.1 Configuration

We perform experiments on simulated environments to verify effectiveness of the proposed path planning.

Map We used randomly generated Gaussian hill composed by multiple Gaussian functions with randomly selected mean, variance, and height. An example is shown in Figure 4. Since it can represent diverse appearance of terrains, we can achieve statistically meaningful results regardless of a specific type of terrains. Gaussian hills were generated within 15-by-20 m² rectangular area. Each hill was composed by 25 Gaussian functions whose
height was randomly distributed within $[0.4, 1.2]$ m. Finally, we made a elevation map from the Gaussian hill by quantization whose horizontal cell size was 0.1 m and horizontal level size was 0.05 m.

**Robot** We assumed that the robot moved $v_c = 0.5$ m/s linear velocity with $m = 12$ kg mass. The robot was driven by internal torque from wheels as shown in Figure 2 (b). The friction coefficients between the ground and wheels are $\mu_s = 1.0$, $\mu_t = 0.8$, and $\mu_r = 0.01$, respectively, which is similar to friction between concrete surfaces and hard rubbers. The robot additionally spent energy around $P_f = 30$ W due to its energy transfer and internal circuits.

**Algorithms** We executed three algorithms to evaluate their performance: A* on 2-D grid maps, distance-optimal A* on 2.5-D elevation maps, and energy-optimal A* on 2.5-D elevation maps. We measured their performance based on travel distance, energy consumption, and computing time. We implemented three algorithms in C++ and executed them in Intel Core i7 at 2.80 GHz (using only single core) with 8 GB RAM. For statistically meaningful results, we executed path planning 100 times from random start to random goals on 10 randomly generated maps (overall: 1,000 times).

4.2 Results

The experimental results are described by three measures with respect to path length which is the number of cells composing a path. This representation explains performance more exactly because three measures highly depend on path length. The performance is presented in Figure 5, and path examples are shown in Figure 6. In the sake of convenience, A* on 2-D grid maps, distance-optimal A*, and energy-optimal distance will be noted by A*-2D, A*-Dopt, and A*-Eopt, respectively. First, three algorithms had almost similar travel distance, and A*-Dopt had slightly better than the others. Since our distance measure is diagonal distance (a.k.a. octile distance), three algorithms had similar distance even though many examples shown in Figure 6 seem quite different in Euclidean sense. However, it is clear that two proposed algorithms generated much natural paths than A*-2D because they avoided unnecessary hill climbing. Second, A*-Eopt found a path which required almost 10 percents less energy than A*-2D. A*-Dopt was also better than A*-2D because it went around unnecessary hills. Third, two proposed algorithms spent more time than A*-2D due to more complex calculation of edge-cost. A*-Dopt spent around 30 percents more computing time, and A*-Eopt spent three times more computing time.

4.3 Limitations

We found several limitations of the proposed algorithms during experiments. It would be good information for other researchers and further works about path planning on elevation maps. First of all, energy-based path planning on an elevation map may suffer from serious quantization error. For example, in our experimental configuration, each cell has elevation error almost 0.025 m in average. The error causes much larger error in potential energy around 2.94 J per each cell, and the derived error is sequentially accumulated along a path. If its path length is 100, the accumulated error in potential energy is around 294 J, which is significant as shown in Figure 5 (b). Probably our energy-optimal path is not optimal in real environments. When we measured the required energy of path by A*-Eopt in non-quantized maps, it often worse than a path by A*-Dopt. A polygon map may have a similar problem because it is also an approximation on real environments. Second, it is necessary to relax our four assumptions, especially constant velocity assumption. To satisfy the constant velocity assumption, we assume the robot will rotate and hold its wheels repeatedly on downhill. Since we regard that such motion undergoes kinetic friction, its energy consumption is much more than that of going up. In other words, our energy consumption model shown in Figure 3 is unnatural because of keeping constant velocity. Finally, A* seems not enough for path planning on an elevation map. A* algorithm stretches its search toward only its eight neighborhoods, so it cannot check traversability of cells which are more than two cells apart. For example, there is a ramp which is so steep that the robot cannot climb strictly. A* cannot find an alternative feasible path which is a series of zigzags as shown in Figure 7. We believe that any-angle path planning [7] can solve this problem.

5. Conclusion

In this paper, we propose A* path planning on uneven elevation maps. Under four assumptions to simplify the problem, we extend A* to generate distance/time-optimal paths and energy-optimal paths on 2.5-D maps. Moreover, we explain about how to check traversability from elevation information with physical constraints. We also show effectiveness of the proposed path planning through experiments on randomly generated Gaussian hills. As further works, we need to deal with limitations of our current approach, which is described in Section 4.3.

References

Fig. 5: The performance of three algorithms is quantified by three measures: (a) travel distance, (b) energy consumption, and (c) computing time.

Fig. 6: Each figure shows an example path generated by three algorithms on randomly generated Gaussian hills.


