Two-level obstacles avoidance method for manned aircrafts

The 13th International Conference on ICT Convergence

Ha L.M, Tien D.M, Hung B.V, Tiem N.M, Tuan N.A

High Technology Industries Company

October 21, 2022

Overview

Overview

Path Planning as Graph Search Problem

Definitions:

- Path planning is a computational problem to find a sequence of valid configurations that moves the object from the source to destination. It is also defined as finding a geometrical path from the current location of the vehicle to a target location such that it avoids obstacles.
- Path planning is almost simulated as a graph search problem where the source and the destination are represented as two nodes (vertices).

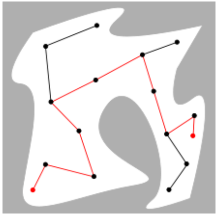

Figure 1: Graph representation of searching area (source: Wikipedia)

Path Planning as Graph Search Problem

Two phases of path planning:

1. Construct a graph representing the planning environment
   - Grid-based map, reduced visibility graph and navigation mesh are the utmost omnipresence representative ways
   - Note that the strategy simulating the search space has the huge impact on path shape and computational cost
2. Apply the graph search algorithm to find the best optimal path
   - Variants of A* algorithms are obviously considered as a base
   - Several papers try to optimise either the cost function or the way the searching propagation

Industrial domain

Flight guidance is the process of building a flight route for an aircraft from A to B that avoids all obstacles, both physical like mountains, clouds, and virtual such as **Air Defense Identification Zones** (ADIZ), conflicted territories, etc.

The most vital features of aviation navigation:

- Turn sensitivity: turn angle and number of turns are taken into consideration as same as path length
- Huge searching space: countless ways to travel from A to B and unknown obstacles
- Uniform cost environment: the cost when passing among areas is nearly constant

Literature overview

Numerous studies that (implicitly) take into account turning costs when using the A* algorithm on a geometric graph. Winter [1] suggest three approaches to modeling cost of turns:

- Costs per turn: in this case the shortest path is the path with the minimal number of turns
- Costs by constraints: legal restrictions for specific turns like turn restrictions can be modeled by maximum edge costs. Mobility constraints, like turn diameters of specific vehicles, can be introduced by classifying street curvature into cost values
- Costs proportional to angles: such cost functions can be used to optimize the total turning angle along a path

Literature overview

There is a trade-off between the path length versus the total turn angle. In some context, the agents demand the simplest path with minimal deviations because they feel more comfortable.

(a) Shortest and simplest path. (b) Path of minimal angle.

Figure 2: The simplest path (black) may be longer than other paths including more turns (a). The path of minimal angle (black) may be longer than other paths (b) (source: Winter [1])

Literature overview

Ballamajalu et al. [2] proposed the Turn and Orientation Sensitive A* (TOS-A*) algorithm which considers three heuristic functions that depend on: *path length, the availability of obstacles ahead*; and *number of turns at the next waypoint.*

Fransen and van Eekelen [3] used another method that calculates the heuristic cost of time and rotation based on *deviate angle*.

J. et al. [4]'s approach is a real-time algorithm, which generates a circular arc trajectory to avoid obstacles based on geometric relations between UAV and obstacles. Following this method, the UAV performs *three continuous turn arcs to avoid an obstacle* and then *returns to the original flight trajectory*.

Lin et al. [5] also studied a geometric approach for fixed-wing aircraft, which allows the aircraft to return to the original course after performing a series of turns.

Original A* algorithm

The A* algorithm, first introduced by Hart et al. [6], is the baseline for several graph searching algorithms. The main purpose of A* is find the lowest cost path. With an enhancement in cost function, A* propagate to destination faster than other ones. The formula of cost at node s:

$$f(s) = g(s) + h(s)$$

where: $g(s)$ is the cost of a shortest path from s_{start} to s **found so far**, $h(s)$ is an estimate of cost of a shortest path from s to s_{goal}.

Original A* algorithm

Two important characteristics of heuristic function:

- Consistency: satisfy triangle inequality
  $h(s_{goal}, s_{goal}) = 0$ and $h(s) \leq c(s, prev(s)) + h(prev(s))$, $\forall s \neq s_{goal}$
- Admissibility: the estimated cost is always less than or equal the real cost
  $h(s) \leq c_{real}(s, s_{goal})$

A* algorithm is said to be *admissible* if it is guaranteed to return an optimal solution.

Overview

Hierarchical approach to flight guidance task

When controlling an aircraft, it is easier for the pilot to fly on straight lines than to turn (change direction) frequently. Our method of obstacles avoidance works on 2 levels:

- Level 1 (Static obstacles avoidance): build a general flight route to avoid large obstacles like countries borders, conflicted areas, etc. using an A*-based algorithm in which we propose a new heuristic function that helps decrease route length and turn angles.
- Level 2 (Dynamic obstacles avoidance): add triangle-shaped segments to avoid small obstacles like clouds, and mountains that intersects with the flight route built at level 1. Additional segments are built by optimizing cost functions based on distances and turn angles.

Level 1 - Static obstacles handling

Static obstacles include borders of other countries, large mountains ranges. In this step, an A* based algorithm is used to build the visibility graph on vertices of obstacles and find a general route from A to B. We propose a new heuristic function that combines the distance cost and deviate cost:

$$h(s) = w_1 h_d(s) + w_2 h_t(s)$$

where:

- $h_d(s)$ is estimated distant cost from s to s_{goal}. It may be calculated by Euclidean distance formula.
- $h_t(s)$ is estimated turn cost at current node s. With $h_t(s) = \frac{\theta}{\pi} h_d(s)$, we turn the $h_t(s)$ into the same unit with $h_d(s)$. θ is the turn angle at current node, $\theta \in [0, \pi]$.

$\forall w_i \in [0, 1]$ and $w_1 + w_2 = 1$, $h(s) \leq h_d(s) \leq c_{real}(s, s_{goal})$ guaranteed the heuristic function is admissible.

Level 2 - Dynamic obstacles handling

On this level, for each dynamic obstacle, we generate a triangle-shaped path that bounds the obstacle. The aircraft changes direction at point p1, heading to point p2, then returns to the planned flight route at point p3. We try to optimize 1) path length; 2) off-planned trajectory segment; 3) direction change.

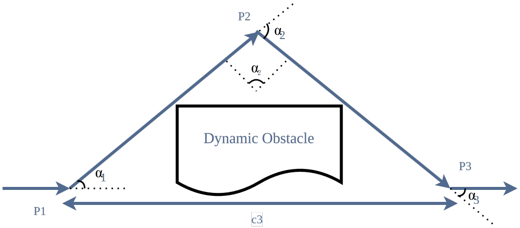

Figure 3: Triangle-shaped path of dynamic obstacle avoidance

Level 2 - Dynamic obstacles handling

We propose three cost functions for those subjects of optimization:

$$C_1(p_1, p_2, p_3) = d(p_1, p_2) + d(p_2, p_3)$$

$$C_2(p_1, p_2, p_3) = d(p_1, p_3)$$

$$C_3(p_1, p_2, p_3) = \begin{cases} \phi_0(constant) & \text{if } \alpha_1 = \alpha_3 \text{ and } \alpha_1 \in \{\frac{\pi}{3}, \frac{\pi}{4}\} \\ \phi(\alpha_1) + \phi(\alpha_2) + \phi(\alpha_3) & \text{otherwise} \end{cases}$$

Note that in C_3, we prefer the angle approximating $\{\frac{\pi}{3}, \frac{\pi}{4}\}$ which is most comfortable with the pilot. R is the turn radius of aircraft.

C_1 is the length of route from p_1 to p_3 via p_2. C_2 is the distance from p_1 to p_3 - the off-planned-route length. C_3 indicates the cost of turning, as we minimise the changes of flight heading. The final cost function for one dynamic obstacle is calculated by formula:

$$C_{total} = \beta_1 C_1 + \beta_2 C_2 + \beta_3 C_3$$

where: $\beta_1, \beta_2, \beta_3 \in (0, 1)$ are weighted parameters for combining cost functions and $\beta_1 + \beta_2 + \beta_3 = 1$

Overview

Result and Comparison

We generated thousands test cases which has several static and dynamic obstacles. The first experiment is in level 1 when comparing our upgraded A* algorithm with default A* and Dijkstra algorithm. We also developed a simulate environment for both flight testing and visualisation.

Table 1: Static obstacle avoiding methods comparison

	Dijkstra	Default A*	**Our method**
Average running time (ns)	14510784	10491297	**10827947**
Average distance (km)	586,29	586,29	**632,86**
Sum of turn angles (radian)	3,24	4,06	**2,63**
Max turn angle (radian)	1,26	1,26	**0,96**
Average number of turns	6,64	6,64	**6,43**

Result and Comparison

Figure 4: Shortest paths were generated by different methods in 1 test case

Result and Comparison

In level 2, we compare our triangle-based method with two constant-angle methods which are traditional method of pilots.

Table 2: Dynamic obstacle avoiding methods comparison

β_1	β_2	β_3	$\frac{\pi}{3}$	$\frac{\pi}{4}$	**Our method**
0,01	0,01	0,98	7706,47	7462,66	**7443,51**
0,01	0,02	0,97	8591,96	8474,15	**8386,85**
0,01	0,09	0,90	14790,38	15554,58	**14468,22**
0,30	0,30	0,40	86194,03	78879,85	**76452,43**
0,40	0,10	0,50	86694,03	73161,79	**72084,95**

Overview

Conclusion

The obstacles avoidance approach in this paper uses two phases, which handle static and dynamic obstacles separately.

- We propose a new heuristic function for the A* algorithm to optimize not only the travel distance of aircraft when avoid obstacles, but also reduce the turn angles and make flight routes simpler
- We also define cost functions that consider turn angles and off-planned-route segments for avoiding dynamic obstacles while not changing the original flight route calculated at the level of the static obstacles

We believe that such an approach helps improve automatic flight route calculation and make flights safer.

References I

[1] Stephan Winter. Modeling costs of turns in route planning. *GeoInformatica*, 6, 2002. doi: 10.1023/A:1020853410145.

[2] R. Ballamajalu, M. Li, C. Hochgraf F. Sahin, R. Ptucha, and M. E. Kuhl. Turn and orientation sensitive a for autonomous vehicles in intelligent material handling systems. *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 606–611, 2020. doi: 10.1109/CASE48305.2020.9216869.

[3] Karlijn Fransen and Joost van Eekelen. Efficient path planning for automated guided vehicles using a* (astar) algorithm incorporating turning costs in search heuristic. *International Journal of Production Research*, 0(0):1–19, 2021. doi: 10.1080/00207543.2021.2015806. URL `https://doi.org/10.1080/00207543.2021.2015806`.

References II

[4] Guo J., Liang C., Wang K., Sang B., and Wu Y. Three-dimensional autonomous obstacle avoidance algorithm for uav based on circular arc trajectory. *International Journal of Aerospace Engineering*, 2021(1-13):1–13, 2021.

[5] Z. Lin, L. Castano, E. Mortimer, and H. Xu. Fast 3d collision avoidance algorithm for fixed wing uas, journal of intelligent robotic systems. *Journal of Intelligent Robotic Systems*, 97(3-4):577–604, 2020.

[6] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:100–107, 1968.