Real-Time Path Planning and Terrain Obstacle Performance for General Aviation Aircraft

James Doebbler, Paul Gesting & John Valasek

AIAA Guidance, Navigation, and Control Conference
Monday, August 15, 2005
Outline of the Presentation

- Motivation
- Objective and Scope
- Proposed Solution:
  1. Search Space
  2. Basic Search Algorithm
  3. Performance Improvement Techniques
- Example Results
- Conclusions
- Future Research
Motivation

Of 5,400 public-use airports, only 715 (13%) have precision instrument approaches (ILS).

Near all-weather accessibility to 5,400 public-use airports?

- 22% within 30 minutes of major/hub airport
- 41% within 30 minutes of any commercial airport
- 93% of population within 30 minutes of SATS-type airport

Ref. SATS program briefing charts
Small Aircraft Transportation System (SATS)

The SATS concept seeks to achieve:

- **Higher Volume Operations at Non-Controlled Airports**
- **Lower Landing Minimums at Minimally Equipped Landing Facilities**
- **Increased Single-Pilot Crew Safety & Mission Reliability**
- **En Route Procedures & Systems for Integrated Fleet Operations**

Auto Aircraft Landing Scheduling

Aircraft “Highway-in-the-Sky”
General Aviation and Free Flight

- A new way of managing air traffic
- Departure from highly structured system to a more flexible, distributed system
- Individual aircraft allowed to change route, speed and altitude, within limits
- More responsibility for aircraft separation safety rests on pilots
- Ultimate decision making authority still lies with the air traffic controllers
- Increased pilot responsibility can be relieved with onboard software to help with decision-making
Previous Research

- Potential field path planning (Hwang, Ahuja)
- Evolutionary algorithm based path planning (Nikolos, Valavanis, Tsourveloudis, Kostaras)
- Motion-library based trajectory planning (Richards, Ward)
- Simplified Memory Bounded A* search (Bokadia, Valasek)
Objective and Scope

- **Objective:**
  - Develop a system that will enable real-time detection and avoidance of terrain obstacles through path planning and dynamic updating

- **Scope:**
  - Designed for General Aviation class aircraft equipped for SATS operation
    - Limited computing power
    - Limited sensors
Cannot search continuous space and still achieve our goals

Set up a network of discrete nodes and connections
- Search for best path made up of these node connections
- The best path through the nodes should be close to the best path in the continuous space

Creating the search space is important to getting good solutions since the final path must come from there
Node Positions

- Create search layers at a range of altitudes
  - Maximum altitude set at the maximum service altitude of the aircraft
  - Minimum altitude set at the lowest of the start and end points in the path
    - Based on the fact that climbing uses more fuel than cruising
    - Valid for terrain avoidance, not valid when an airplane must descend to avoid an obstacle
  - Fill in intermediate levels, related to node connections
- In each layer, create an evenly-spaced 2D grid of nodes
  - Closer grid spacing increases the resolution of the path, but also increases the computation time
Node Connections

- Path segments are defined by the node connections based on two parameters:
  - Angular range (\(\alpha\))
    - Determines the maximum turning angle per path segment
    - Helps create
  - Distance range
    - Determines the length of each path segment
    - Minimum distance based on the vertical spacing of layers so that the aircraft follows its standard climb rate
    - Span of the range is chosen to include a reasonable number of nodes (usually set equal to the node spacing)
- Nodes are connected in the same level and to one level above and below
Node Connection Diagrams

- Goal Node
- Current Node
- Connection to:
  - Layer above
  - Current layer
  - Layer below
Search Algorithm Overview

- **A* Heuristic Search Algorithm**
  - Aims to minimize a path cost function
    - \( g(n) \) = cost to reach node \( n \)
    - \( h(n) \) = estimated cost to reach goal from node \( n \)
  - Cost function \( f(n) = g(n) + h(n) \)
  - The total cost along any path must be monotonically increasing
  - Complete and optimal search
  - Expand paths with the smallest cost first and repeat until goal is found

\[ h = 10 \]
\[ g = 0 \]
\[ f = 10 \]
\[ g = 4 \]
\[ h = 7 \]
\[ f = 11 \]
\[ h = 0 \]
\[ g = 4 + 7 \]
\[ f = 11 \]
\[ g = 4 + 4 \]
\[ h = 5 \]
\[ f = 14 \]
Search Algorithm Details

- Cost functions used:
  - \( g(n) = g(n-1) + \text{dist}(n,n-1) + \sum w_i(n) \)
  - \( h(n) = \text{dist}(n,\text{goal}) \)
  - \( w_i \) are additional cost factors
    - Climbing incurs an additional fuel cost
    - Terrain collisions are given very large costs

- Collision Detection
  - Points along the path segments are checked against terrain data
  - A path segment collides if any of its points pass within specified distance limits of the terrain

- Priority heap (based on lowest cost) used to store information about the nodes/paths
Performance Improvement Techniques

- **Simplified Memory-Bounded A* Search**
  - When a specified amount of memory is used, only keep the current best paths
  - **Not guaranteed optimal**

- **Generate nodes and connections dynamically**
  - Does not create the nodes and connections beforehand
  - When a node is expanded, its children are created and the node is deleted from memory

- **Profiling of the algorithm showed that the majority of processing time was spent checking points along path for terrain proximity**
  - A significant number of the path segments do not hit terrain at all, but still take as long or longer to check than paths that do hit terrain
  - Major improvement potential from optimizing the collision checking processing
Multi-Resolution Terrain Data

- Represent terrain data as a 2D grid of boxes with each box having the value of the highest terrain in the region encompassed by the box.
- Generate several versions of this data with different size boxes - smaller boxes are higher resolution data.
- Start checking paths against the largest boxes.
- If possible terrain collision is detected, check against next higher resolution terrain data until sure.
- Must generate all the levels of terrain, but this only needs to be done once beforehand for the terrain data.
Numerical Example

- Goal: Show that the algorithm can run quickly on present day computers (2.4GHz Pentium 4, 512MB RAM, implemented in C++)
- Four test cases are run to compare effects of different node spacing (5, 2, 1 and 0.5 mi) on the quality of the solution
- The solutions were also compared to a run without the multiple terrain resolutions which took approximately 3 minutes with a 0.5 mile spacing grid
Example Parameters

- **Terrain data:** approximately 70 mi x 100 mi USGS terrain elevation data
- **Distance from start to goal:** approximately 100 mi
- **Node connections:** approximately 8 mi long
- **500 ft vertical separation between layers**
- **Aircraft parameters (large twin-engine GA aircraft)**
  - Cruise Airspeed: 140 knots
  - Total Cruise Fuel Consumption: 158 lbs/hr
  - Climb Rate: 500 ft/min
  - Climbing Airspeed: 120 knots
  - Total Climb Fuel Consumption: 240 lbs/hr
  - Maximum Altitude: 20,000 ft
Example Results

- **5 mile spacing**
  - 5 seconds
  - Rel. Path Cost=1.35

- **2 mile spacing**
  - 16 seconds
  - Rel. Path Cost=1.18

- **1 mile spacing**
  - 26 seconds
  - Rel. Path Cost=1.06

- **0.5 mile spacing**
  - 38 seconds
  - Rel. Path Cost=1.00
Conclusions

- Developed and demonstrated a real-time discrete-node heuristic search method for path planning with terrain obstacle avoidance.
- The multiple resolutions in the terrain data decreased the solution times by approximately a factor of 4 in the test cases run.
- The grid spacing of the search space has a significant impact on the quality of the generated path.
- Resulting method utilizes computer processing and memory requirements that are realistic for a future SATS-type GA aircraft.
Future Research

- Evaluate other node distribution schemes
- Investigate better heuristic functions to account for more than just the straight-line distance
- Introduce a more complex cost function based upon additional flight parameters
- Reduce memory usage by reducing the terrain data set needed in memory at one time (e.g. by paging)
- Extend to generic dynamic obstacles
- Integrate terrain avoidance with existing weather and traffic avoidance modules to create a single path advisory system
Thank You

Questions
or
Comments