Real-Time Obstacle and Collision Avoidance System for Fixed-Wing Unmanned Aerial Systems

PhD Oral Defense

June 6, 2012

Julien Esposito

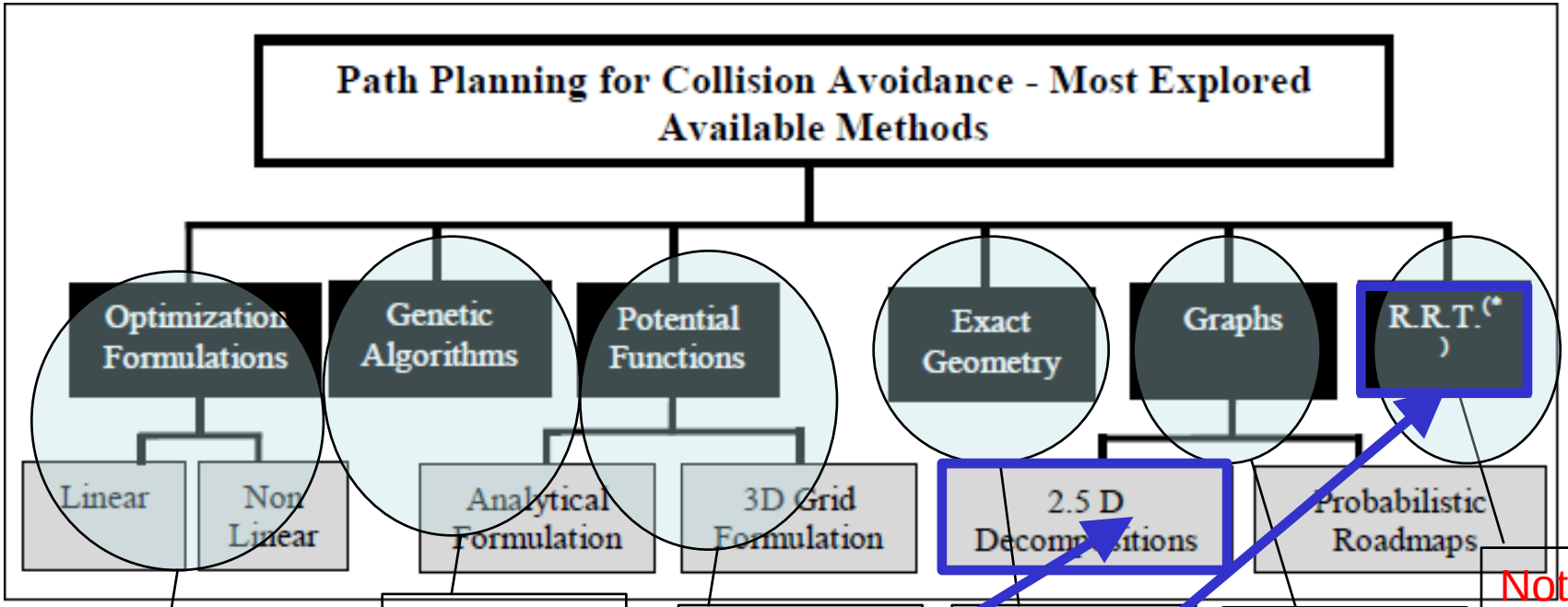
Acknowledgements for Funding Received

- KU TRI – Transport Research Institute
- KU GRF – General Research Fund
- CReSIS – Center for Remote Sensing of Ice Sheets
- NSF
- NASA

Goals and Ambitions

- Long term goal: reduce accidents, mid–air collisions, encourage FAA to include UAS in the National Airspace
- Mid-term goal: provide non-collaborative mid-air threat avoidance with the equivalent of "Google Maps for UAS"
- General goal: FAA not comfortable with actual threat avoidance scenario? → keep research up with simulations
- Research-specific to goal: Add Intelligence and realism to existing algorithm used in terrain avoidance.
- How:
    - Previous research: let unreachable zones to be explored → no sense in real-time avoidance → Change that!
    - Account for existing terrain, and remember where obstacles are detected in general → Anticipate and react, not just react

Existing Work



Solver= Black Box

Too heuristic

Grid dependent

Constant Speed

No Motion Accounted for

Not realist For real-time

Terrain Maps

+

+

=

Intelligent Threat-avoidance algorithm

Animation

Philosophical Aspects of Presented Work

Why prioritizing reaction over anticipation, i.e. why researching only the avoidance of a danger and not trying to anticipate that danger in the first place by using available data?

- Why not developing a piece of software such that a general elevation data map is transformed into the equivalent of what Google Maps provides (point location in urban environments)?
- Why not developing a variation of the RRT which improves the simulation realism by directly accounting for existing terrain data?

Original Contribution

Offline (before flight) Contribution

This reorganization of height maps into stratified tessellated two-dimensional obstacle-free trapezoidal flying zones. The end product is a Synthetic Terrain Avoidance (STA) Network.

Online (real-time, in-flight) Contribution

The second original contribution of this research to the field is the development of new type of Rapidly - exploring Random Tree using shape functions for point mapping onto the trapezoidal flying zones, and STA Network for Smart Rerouting

Outline

- Introduction to Non-Collaborative Mid-Air Threat Avoidance (TA) and Research Motivation
- Offline Trajectory Planning
- Online Reactive Mid-Threat Avoidance
- Summary and Conclusion

What is Non-Collaborative Mid-Air Threat Avoidance (TA)?

In General

- Relies on own sensors to detect & avoid threats (e.g. radar + vision algorithm)
- Experimental → Dangerous → Simulation Needed

In the presented work

- Before flight use elevation maps to produce Synthetic Terrain Avoidance Networks (STA Networks)
- During flight use a new derivation from the original Rapidly-exploring Random Tree algorithm

Motivation

Previous research focus on algorithms only, instead of making use of surveys for added intelligence

Offline TA – The Advanced Waypoint Mapping Generator (AWMG) – Why?

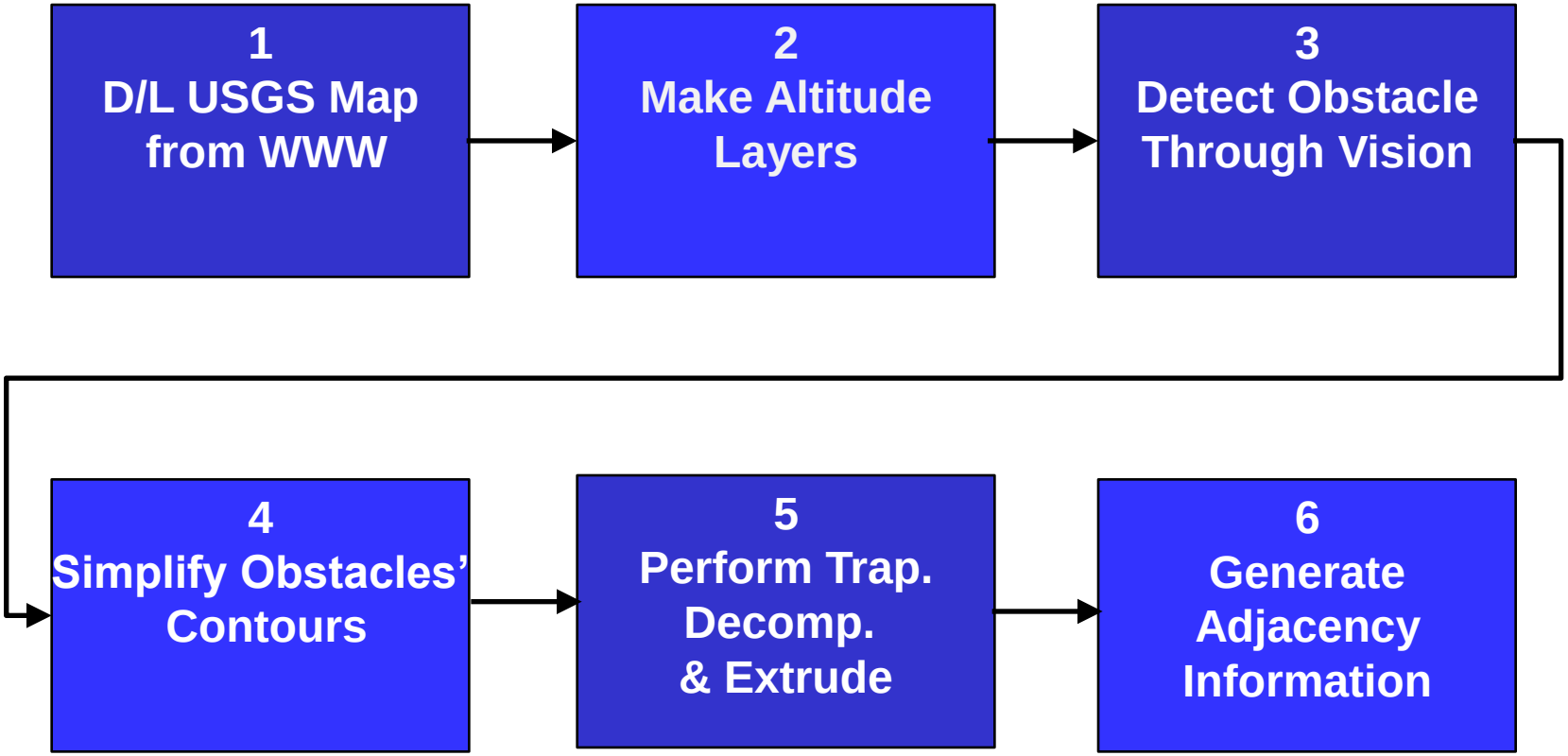
Motivations

- Sectional Charts exist for GA – What about UAS?
- Why Separating Algorithm from the Environment is it Applied to? Take Algorithm and Environment altogether to be smart
- Why not Unmanned Flyable Waypoint Generation?

Goal

To provide Added Intelligence to Real-Time (online) Reactive Threat Avoidance

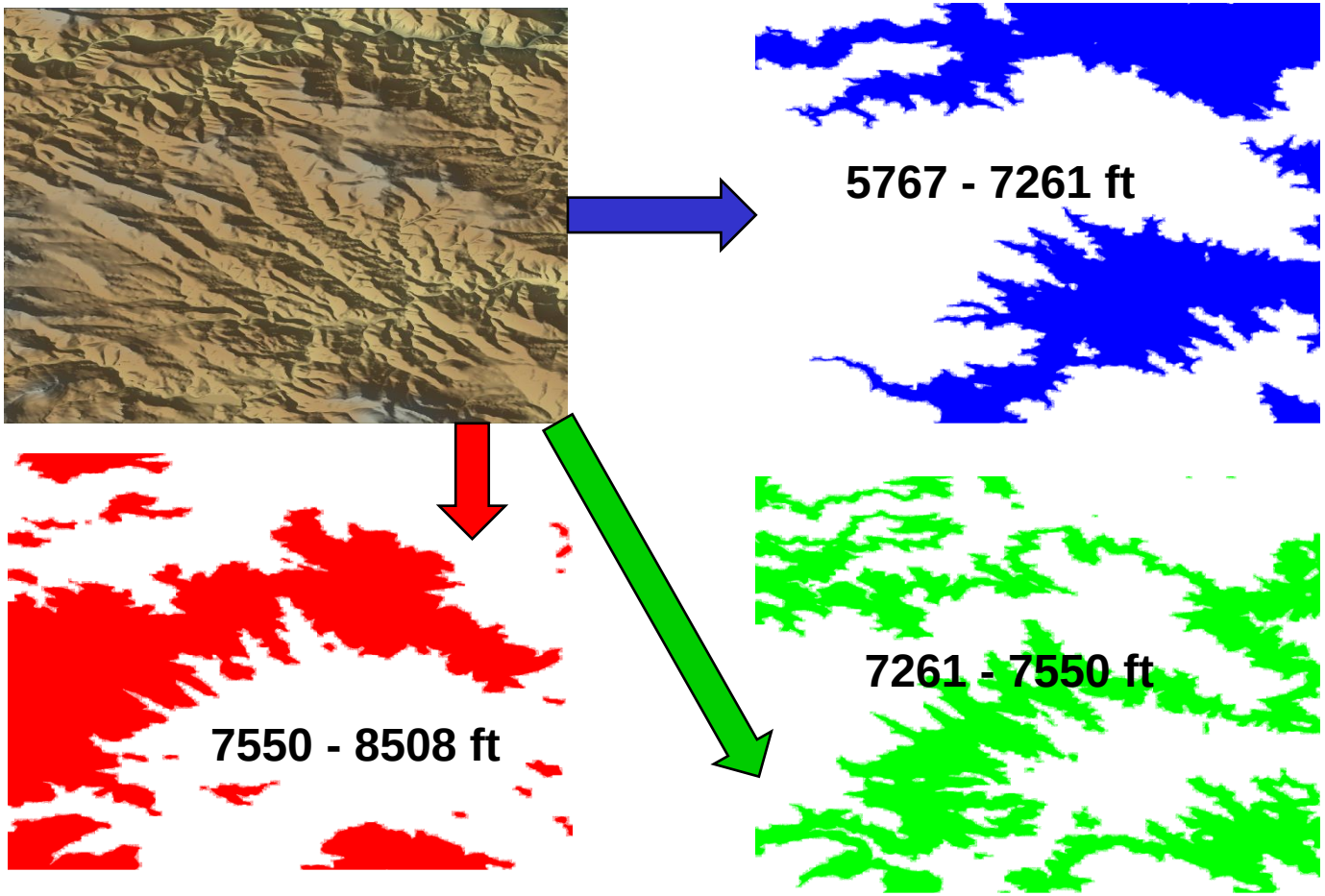
The Advanced Waypoint Mapping Generator – Overview



Motivation Simplify existing elevation data by treating it as superposition of 2D altitude layers to allow altitude scheduling on request

Goal Smart equivalent of sectional charts used in GA, rerouting-ready

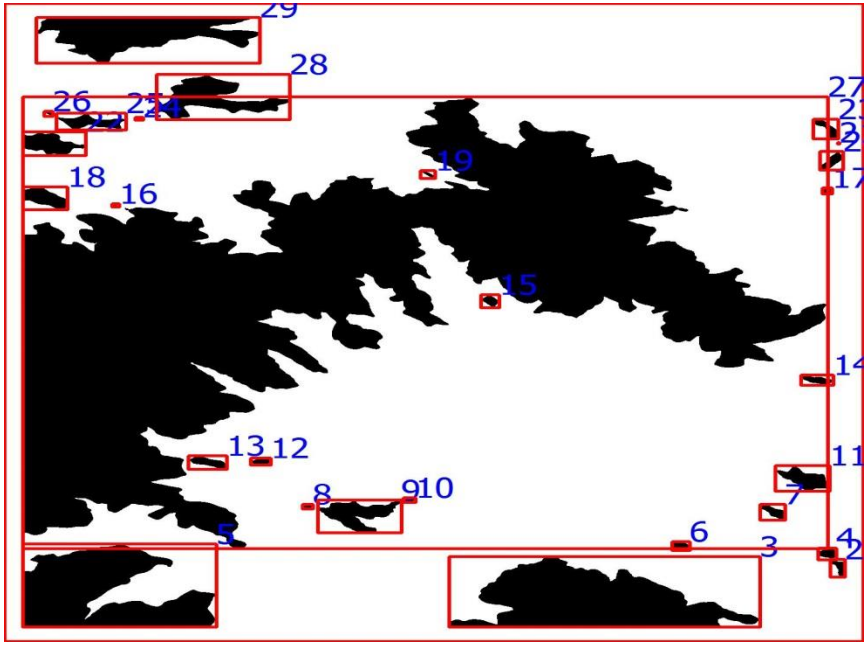
AWMG – Create 2D Altitude Layer (3)



<u>Motivation</u> Simplicity. Approximate 3D Path Finding by Stacks of 2D Layers

The AWMG – Detect Obstacle Through Vision (3)

Original Elevation Layer
Black = obstacles || White = Free space



Contour Detection
Obstacles surrounded by red bounding boxes

Question: **How to Extract Contours When Some Are Not Closed?**

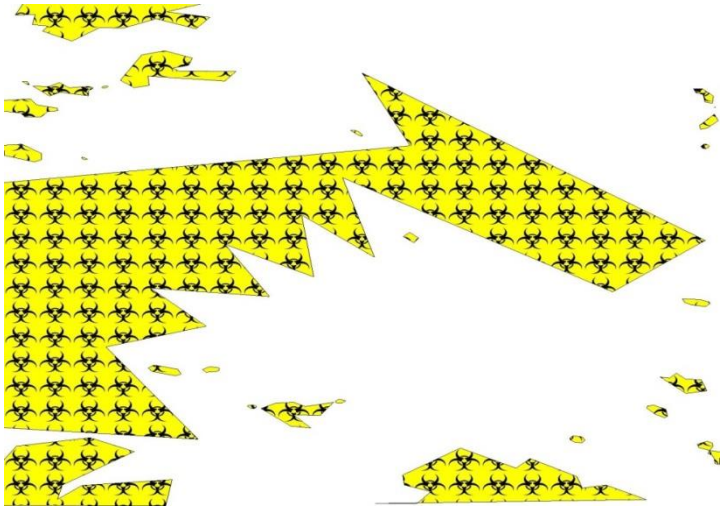
Answer: **Add Artificial White Border Surrounding the Layer's Boundaries**

The AWMG – Obstacle Contour Simplification (4)

Motivation vectorize altitude layers in order to restructure free space into addressable sub-locations

Black ≡obstacles

Obstacle contour simplification



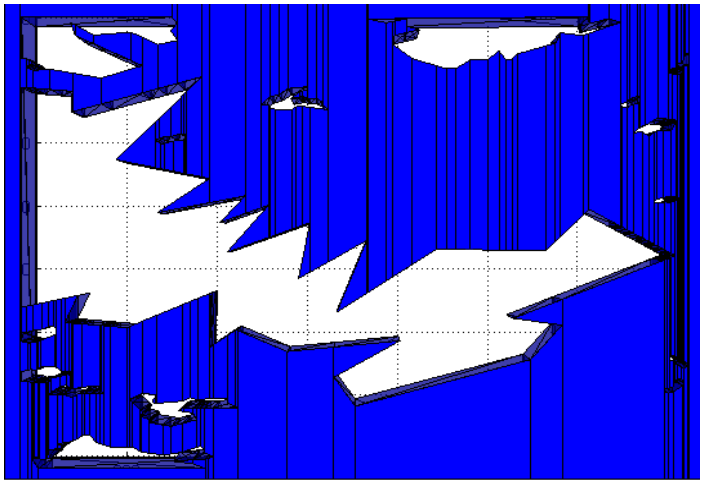
Biohazard texture ≡ obstacle

Algorithm employed for contour simplification

Douglas-Peucker

The AWMG – Trapezoidal Decomposition & Extrusion (5)

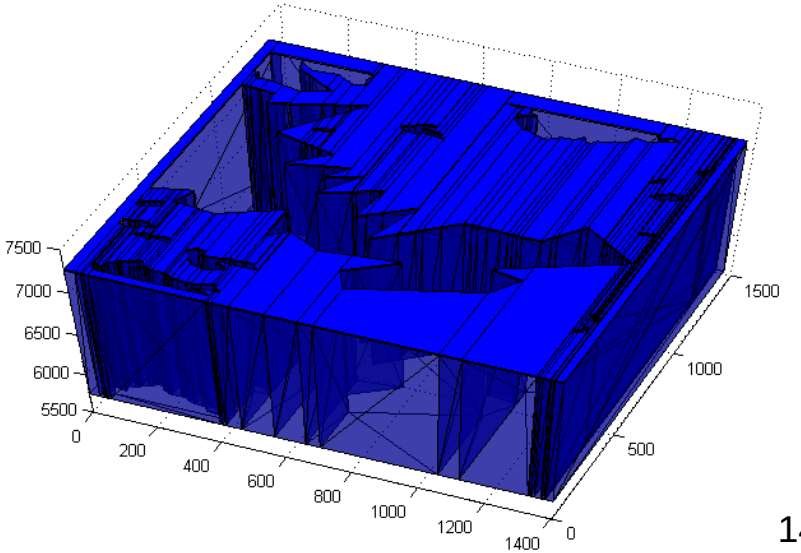
Motivation Vehicle location (decomp.) in 3D world using 2D Layers (extrusion)



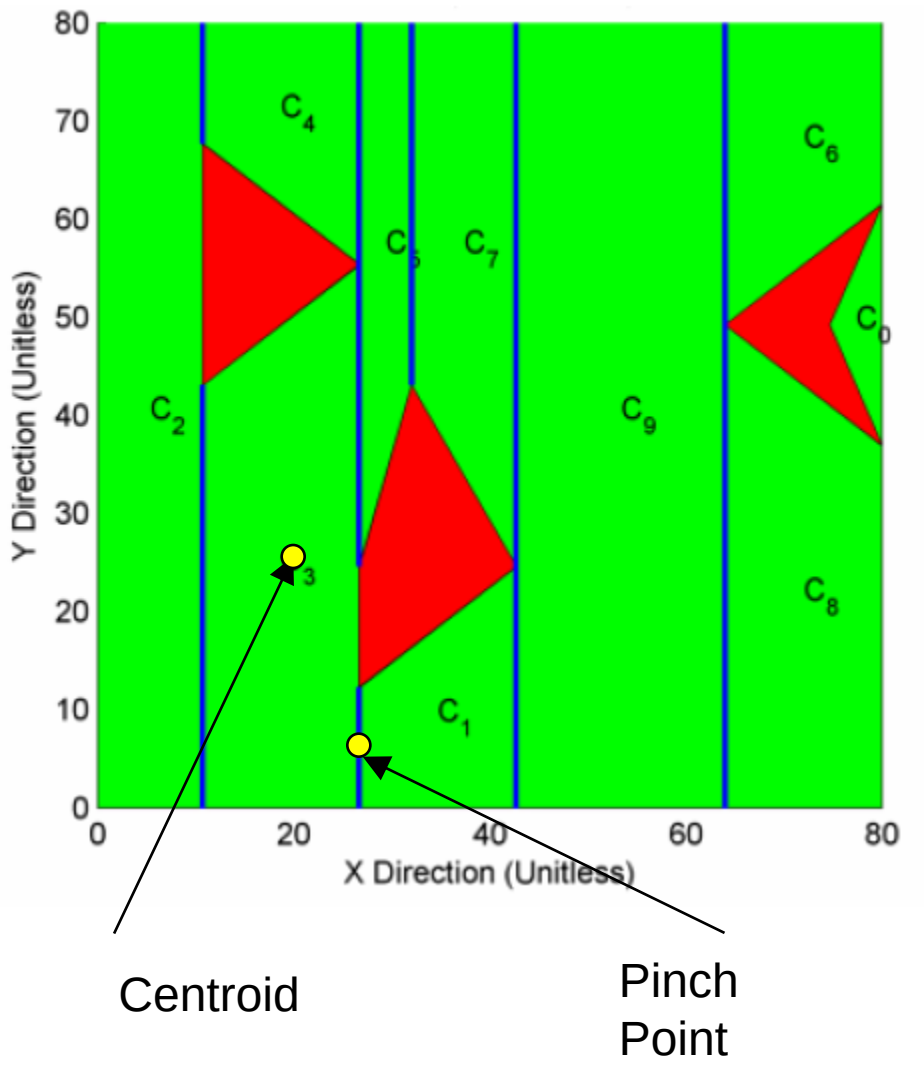
Blue = free space

- each vertex traversed by a line
- Vehicle location precision depends on cells' respective sizes
- Geometric paths are determined by centroids and pinch points

Extrusion enables a vehicle location in 3D world using 2D layers



AWMG – Generated Geometry (6)



Results

- Synthetic Terrain Avoidance network → free space is decomposed and ordered
- Cell connectivity is determined by edge sharing
- Travel between adjacent cells → use pinch points and centroids to transit from cell sequence to waypoints

Interest

Network can be interrogated to list all paths between two cells

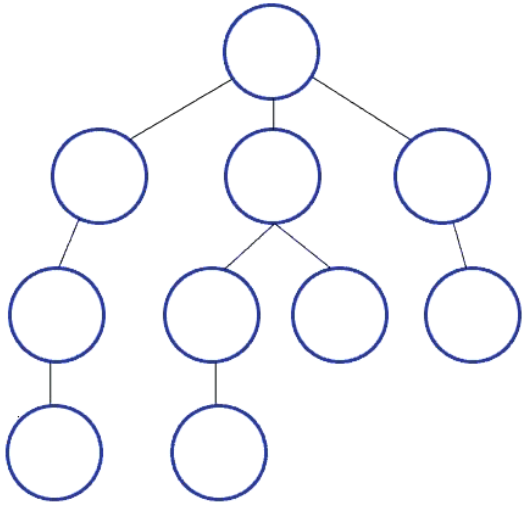
AWMG – Preliminary Result – Interrogatable Cell Sequence

Trapezoidal Cells → vertices (V)

Shared Edges → Edges (E)

Benefit

Alternative path finding treated as standard Graph Theory Problem G(V, E)

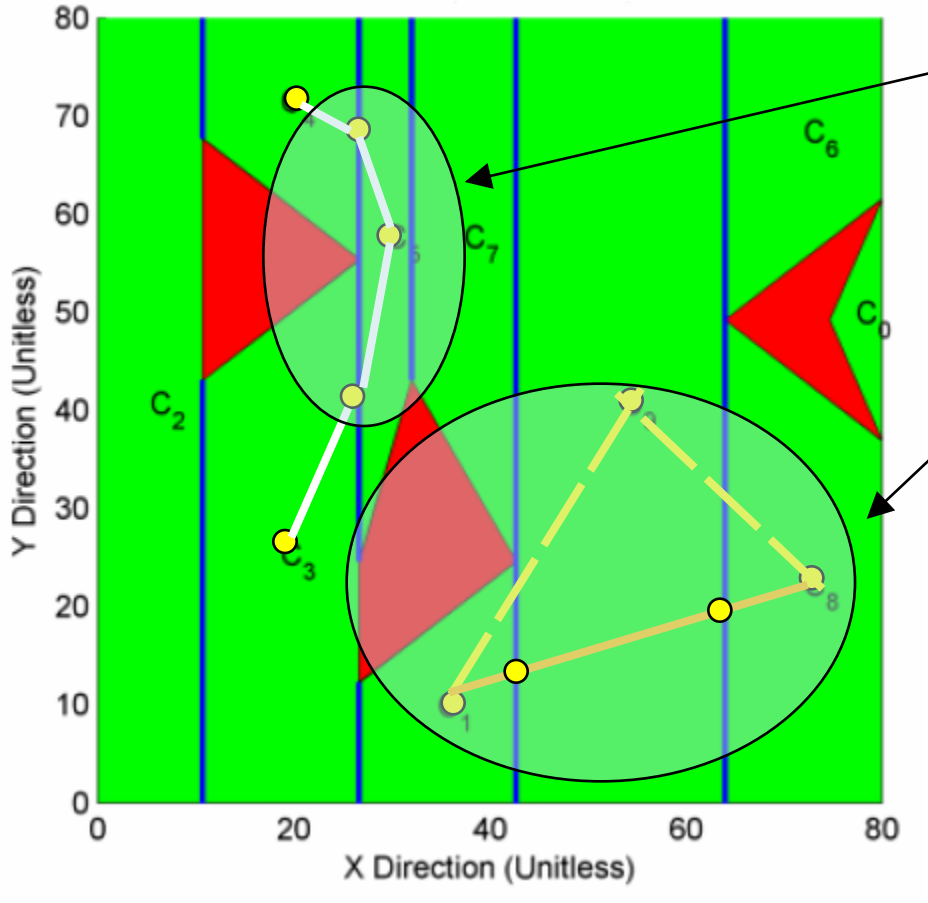


Depth First Search Animation

Result

- List all cells sequences between two cells is possible
- Rerouting A/C is possible if alternative cell sequences are available

AWMG – Inter – Cell Path Simplification



Vertical Trapezoidal Decomposition

Change in x direction → no simplification

No change in x direction → centroid not needed

Interest

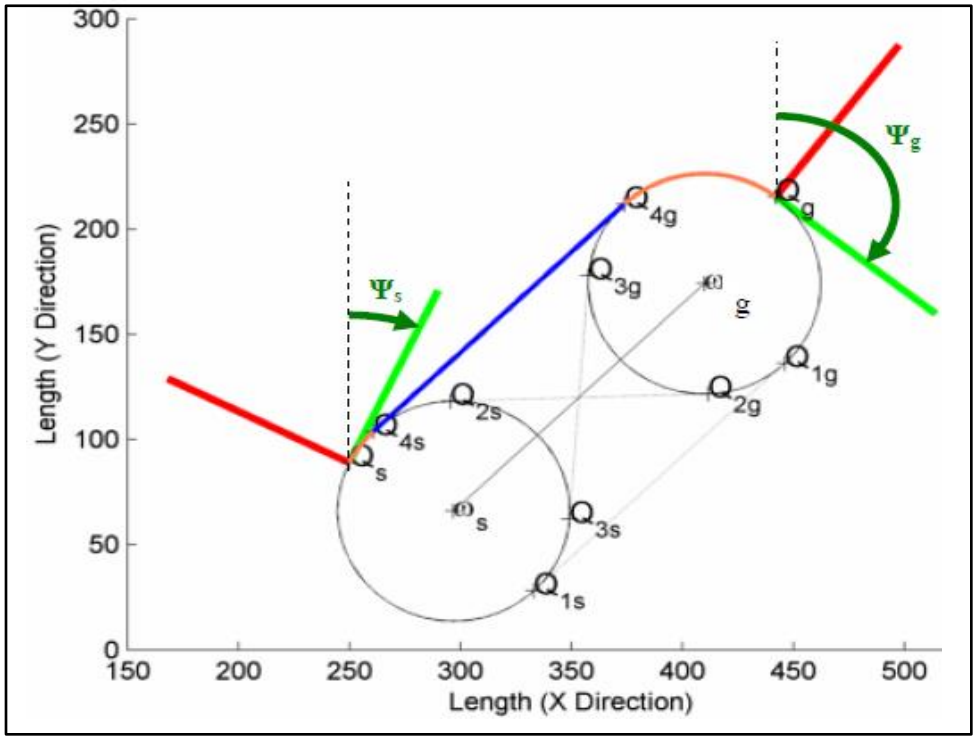
Paths can be ranked without suffering from penalty induced by original path generation process

AWMG – From Paths to Offline Flyable Waypoints Using Dubins Paths

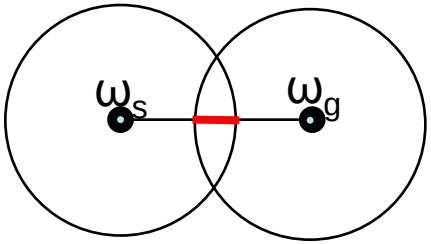
- Hovering impossible for a fixed-wing A/C (V_{stall} constrained)
- Minimum turn constraint $R \triangleq \dfrac{V_T^{\,2}}{g \times \tan\phi}$

Conclusion

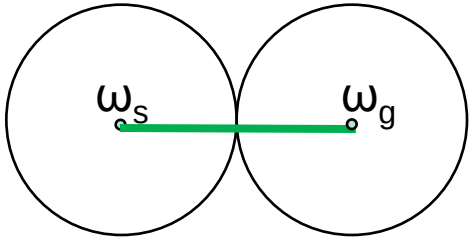
Not all paths can be flown



Dubins condition must be checked

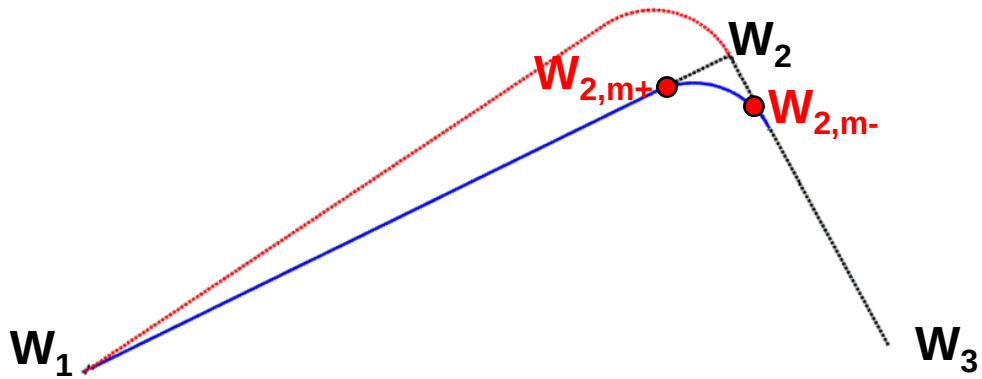


$d(\omega_s, \omega_g) < 2R$



$d(\omega_s, \omega_g) \geq 2R$

AWMG – Corrected Dubins Paths



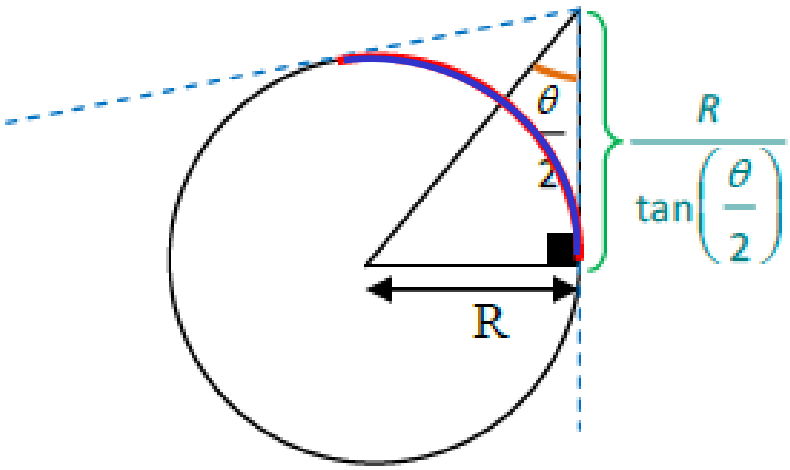
Dubins paths allow heading changes, but do not respect flying leg directions

Result

$$d\left(w_{2,m-},w_2\right)=d\left(w_{2,m+},w_2\right)=\frac{R}{\tan(\theta/2)}$$

$$\text{with } \left(\overrightarrow{w_2w_{2,m-}},\overrightarrow{w_2w_{2,m+}}\right)=\theta$$

and R $\triangleq$ Radius of Curvature



AWMG – What if the Dubins Condition is not Satisfied?

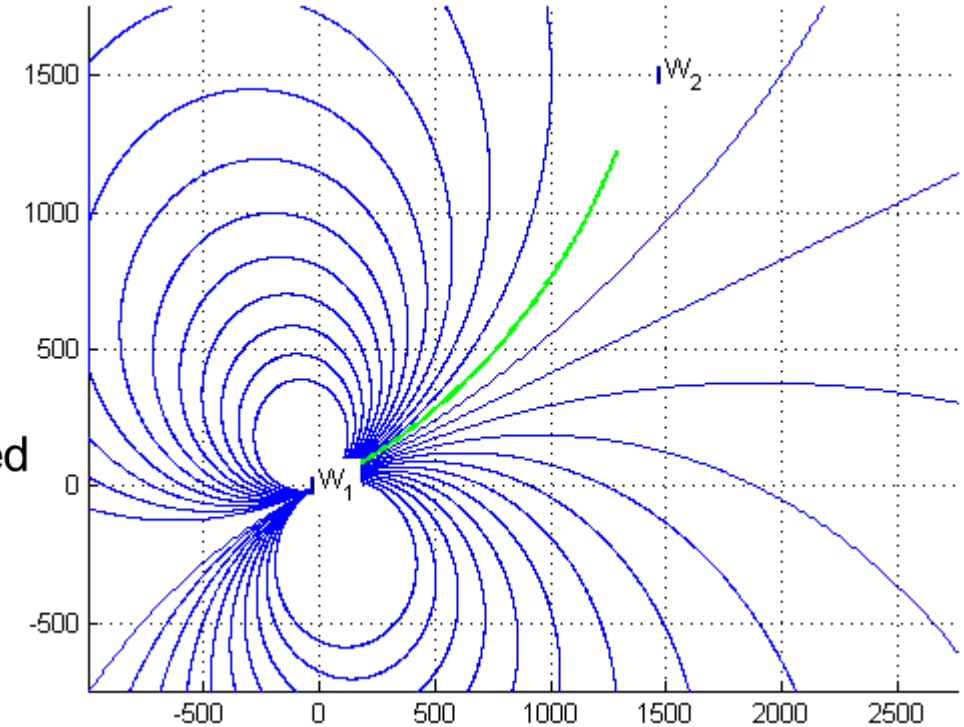
- Constant V_T assumed (u_1) and ϕ can vary between $[-\phi_{max}, +\phi_{max}]$ (u_2)
- Estimate time between $W_1(x_1,y_1,\psi_1)$ and $W_2(x_2,y_2,\psi_2)$ as

$$t = 10 * d(W_1, W_2)/V_T$$

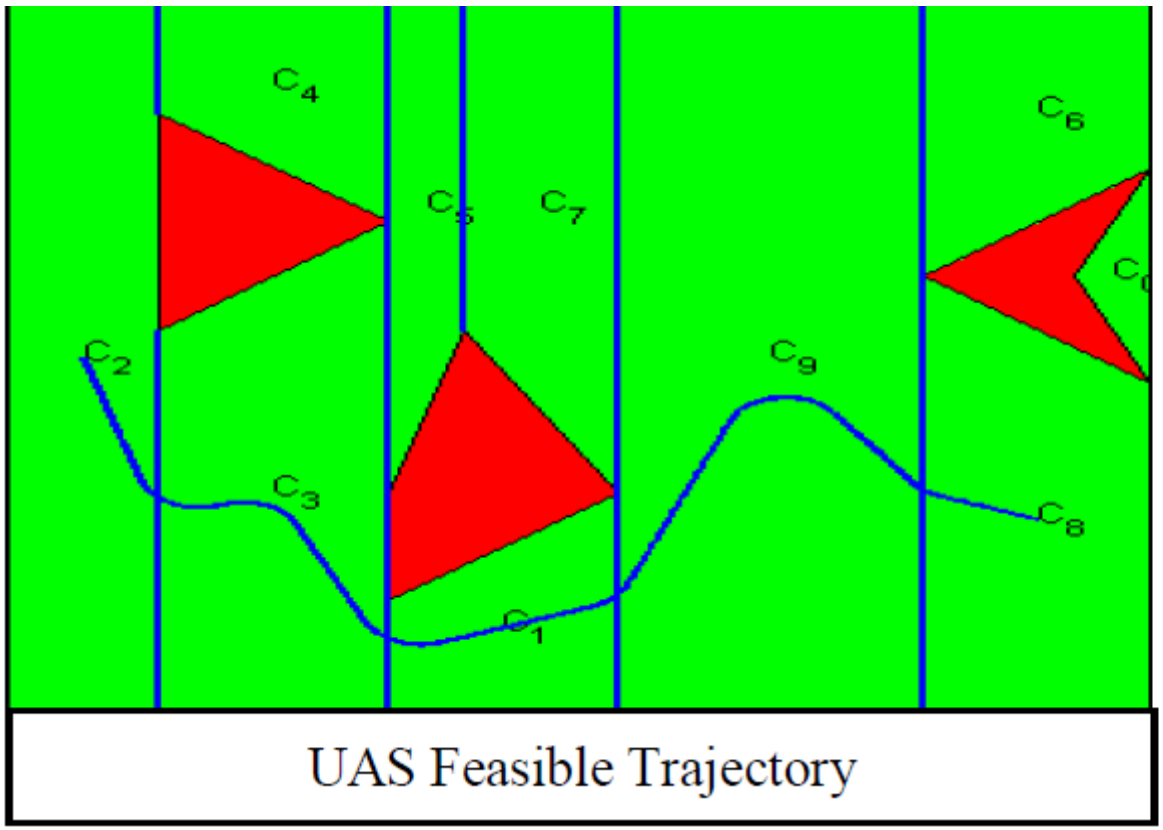
- Using Runge-Kutta and estimated t, solve

$$\begin{cases} \dot{x} = u_1 \cos\psi \\ \dot{y} = u_1 \sin\psi \\ \dot{\psi} = \dfrac{g \tan u_2}{u_1} \end{cases}$$

for each ϕ and retain the solution closest to W_2



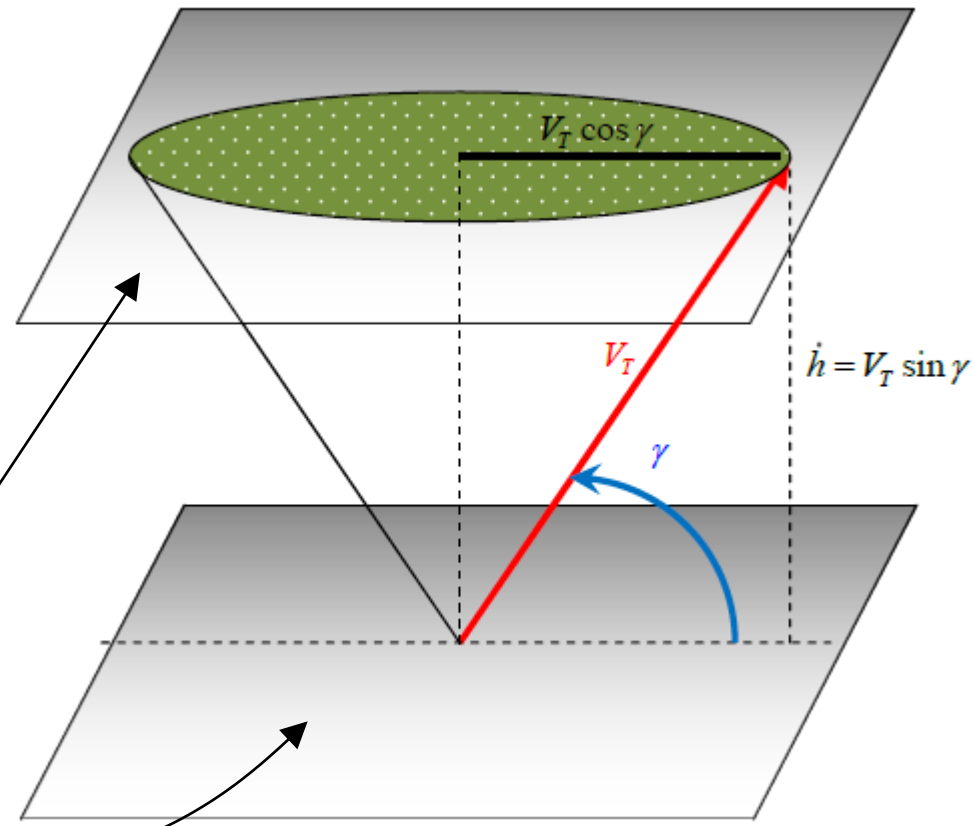
AWMG – End Result



F - UAS Flight Dynamics-Based Path Smoothing

AWMG – Inter-Layer Cell Connectivity

- Requires polygon clipping test (Sutherland–Hodgman) to determine connectivity
- if clipping test positive, additional Longitudinal Maneuver Cone test needed
- DFS result for inter-layer path finding grows dramatically for 2 or more layers



Higher altitude cell

Lower altitude cell

Longitudinal Maneuver Cone Test

AWMG – Videos

AWMG for Offline Trajectory Generation Intermediate Summary

- GA has sectional chart, now UAS mid-air threat avoidance has STA Network
- Elevation map reorganization and simplification enables vehicle location by subdividing the free-space into addressable regions
- Free-space subdivision allows to find multiple paths between two cells, but inter-layer path finding is more complicated
- Dubins paths and ODE solving are used to convert cell sequences into flyable waypoints

Part II – Reactive Threat Avoidance Using the DDRRT

Motivation

Need to face unpredicted mid – air threat situations by providing alternate rerouting when possible via a spatial exploration algorithm

Why Via Algorithm?

- Non-collaborative threat avoidance for drones is not authorized yet by FAA but research must go on → need for simulations
- Separate threat avoidance (logic) from threat detection (sensors). Threat detection is assumed

What is RRT?

A particularly efficient algorithm to explore space, using randomization to spread a tree which explores space rapidly. RRT = Rapidly – Exploring Random Tree

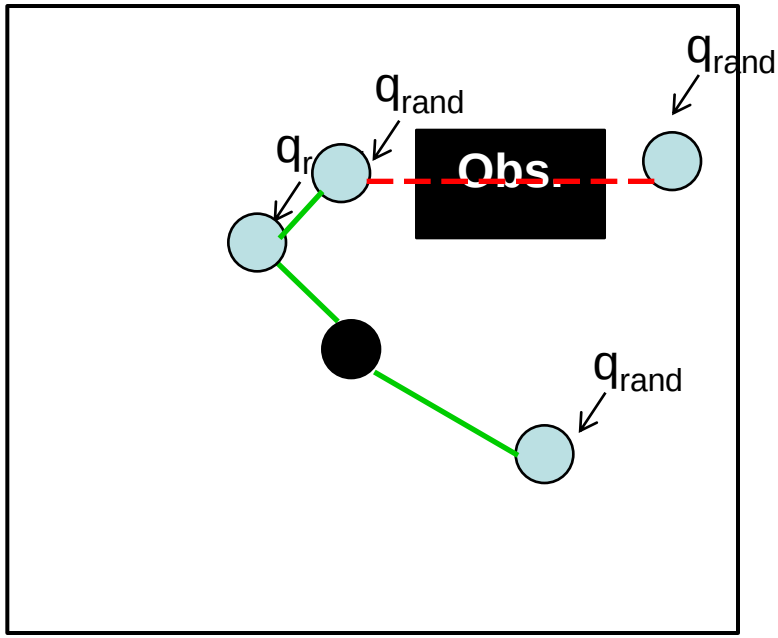
What is DDRRT?

RRT with added intelligence from STA Network taking explicit advantage of the terrain flown

Applications of RRT Derivatives

- **Reactive Holonomic Path Planning** find a collision-free path for a robotic manipulator
- **Reactive Non-Holonomic Trajectory Planning** find a collision free trajectory for a vehicle whose dynamics are constrained through equations of motion (fixed wing A/C dynamics), and Vstall (no hovering)

What is Rapidly-exploring Random Tree?



About the Animation

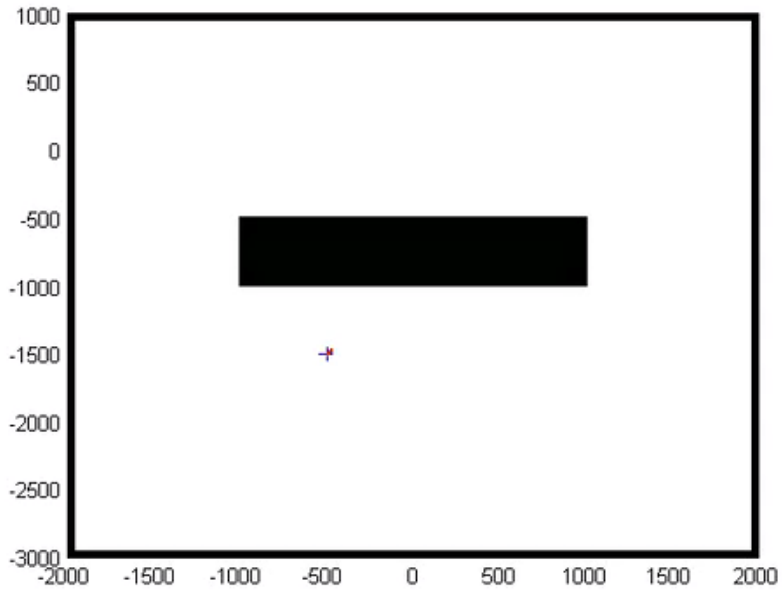
1. Q_{rand} are randomly selected positions in the space
2. a pulsating position is the nearest neighbor of a given q_{rand}
3. The algorithm itself has no knowledge that it cannot extend its branches through obstacles.
4. Non-obstacle traversal results from added logic using a software collision package (Opcode)

Inherent strength of the algorithm: rapid – exploration

Inherent weakness of the algorithm: potential for too much exploration

What is Rapidly-exploring Random Tree (continued)?

Definition: Algorithm iteratively growing branches from the closest node to a randomly selected sample, its nearest neighbor



From Video

Red – beginning of path finding

Green – end of path finding

Problem – "green grows on red"

Lack of Focus → Potential to not find the goal

Pros

- **Designed for exploration**
- **If there is a path between two points, will find it asymptotically**

Cons

- **Explores zones of little interest too: dispersion**
- **Works on a finite number of samples**
- **Non-holonomic in essence**

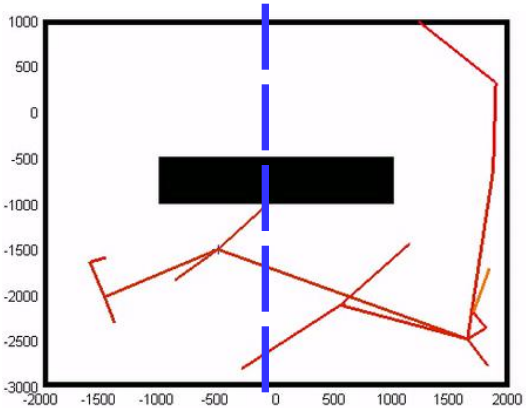
What are the existing types of existing RRTs?

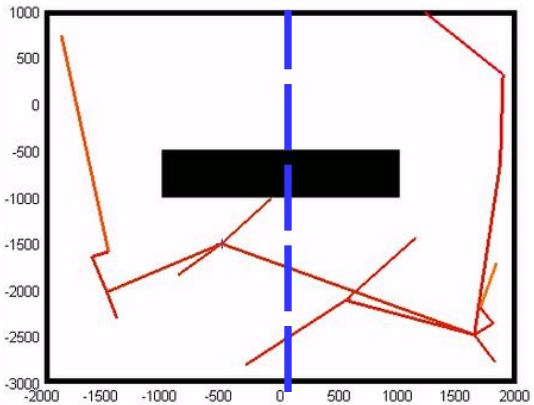
- **the Goal-Biased RRT** Shaping the probability to grow tree toward the goal. Danger → obstacle trapping, heuristic probability shaping
- **The Multi-RRT** faster than single RRT, but does not make sense for a single vehicle real-time exploration A UAS cannot be at two different point at the same time
- **RRT Star** Finds close to optimal solution but require k nearest neighbor → prone to dispersion

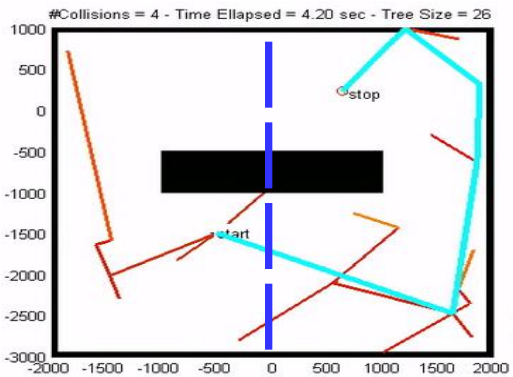
What is the DDRRT and why is new?

- 1st Rapidly-exploring algorithm to explicitly take terrain (or moving threats) into consideration
- space either known before the flying mission or dynamically discovered and reorganized (STA Networks) to be used by the DDRRT
- flown space → divided into obstacle-free trapezoids(used for trajectory re-planning, if needed and when possible). Result = Dispersion Reduction
- When exploration fails, STA Network can be polled to check for alternative prescribed routes against A/C dynamic constraints → Added intelligence

What is Dispersion?







<u>1 – Beginning of Path Finding</u>
Majority of exploration going to the RHS

<u>2 – Middle of Path Finding</u>
Exploration switches back to LHS

<u>3 – End of Path Finding</u>
The exploration on LHS was useless

<u>Dispersion – Pros:</u> Favors exploration
<u>Dispersion – Cons:</u> Loss of Focus – For real-time threat avoidance dispersion must be local, not global

Solution – Exploration is necessary but must be reduced to areas of interest. This can be achieved with the Double-Dispersion reduction Rapidly-exploring Random Tree (DDRRT)

DDRRT – Why Two Types of Dispersion and What Connection With STA Network?

Punctual Dispersion Reduction (PDR) – Adaptive Goal Biasing

Shapes the random variable distribution so growing tree toward waypoints is reinforced, but **goal biasing decreases** after too many failures

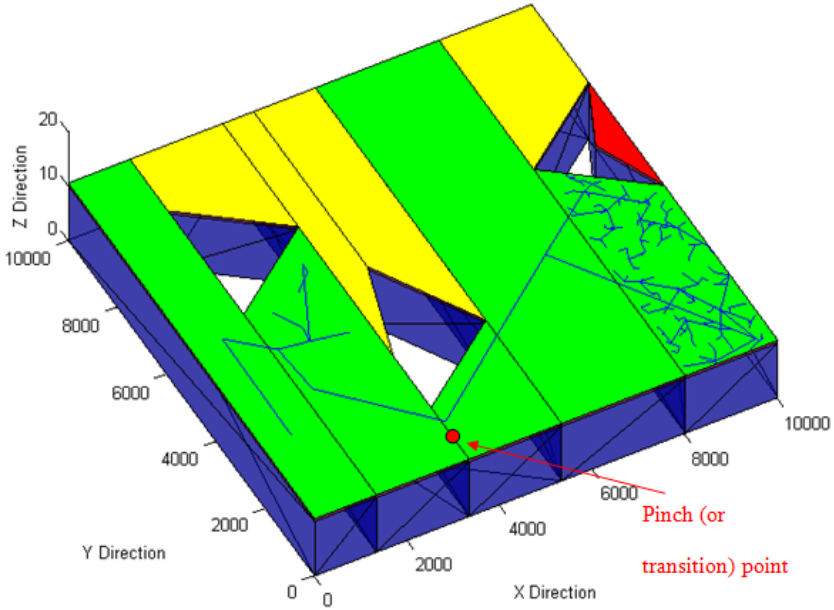
Spatial Dispersion Reduction (SDR) – Exploration Stays Within Local Area

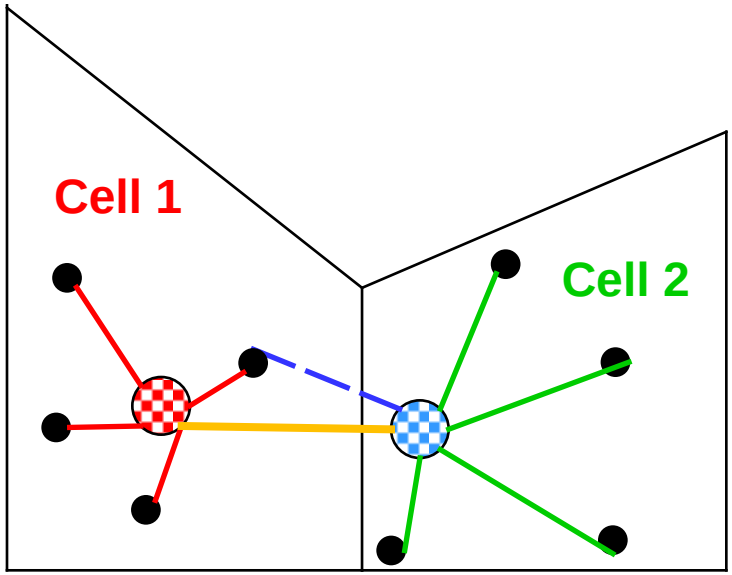
- At cell level, forces the PDR to stay within the current cells of interest and after the biasing of the PDR falls below a certain percentage the DDRRT remains local.
- # branches proportional to the area of the cells. Example: if 3 cells, 2000 seeds to draw and current cell 50% of total altitude layer, 1000 seeds spent on that current cell

Use of STA Network

Once PDR and SDR are exhausted, STA Network is polled for alternatives routes. Why? DDRRT must identify a failure after attempt before considering alternatives

What is the DDRRT SPR and How Does it Work?





<u>Cells of interest in green</u>, alternative cells in yellow. Even without following waypoints, the exploration remains in the green cells via use of shape functions.

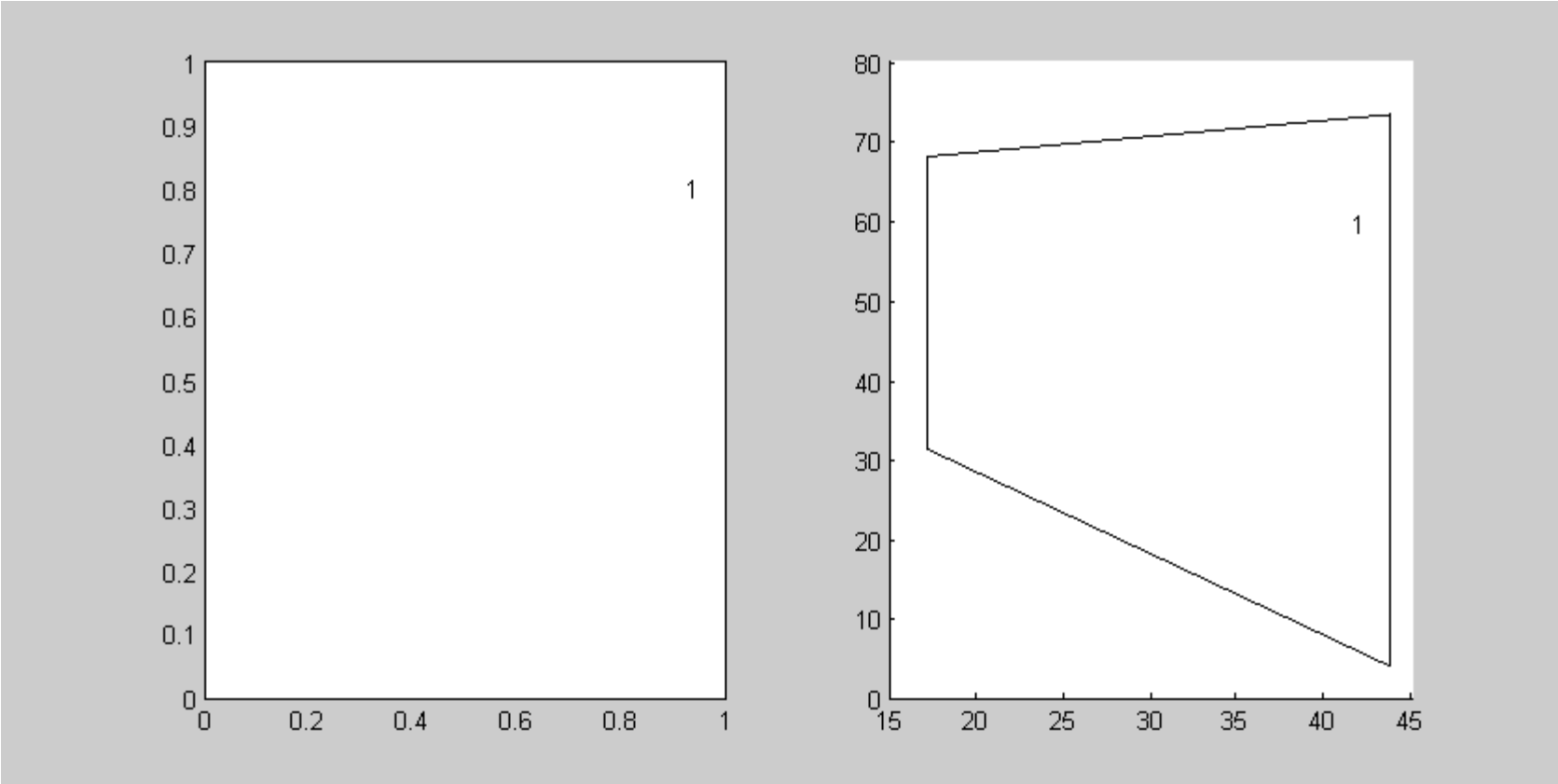
Challenge: Prevent seeds in cell 1 to be potential nearest neighbors of cell 2.

Why: need to move forward in the cell sequence

Solution: create new root in cell 2 so its nearest neighbor search is not influenced by cell 1

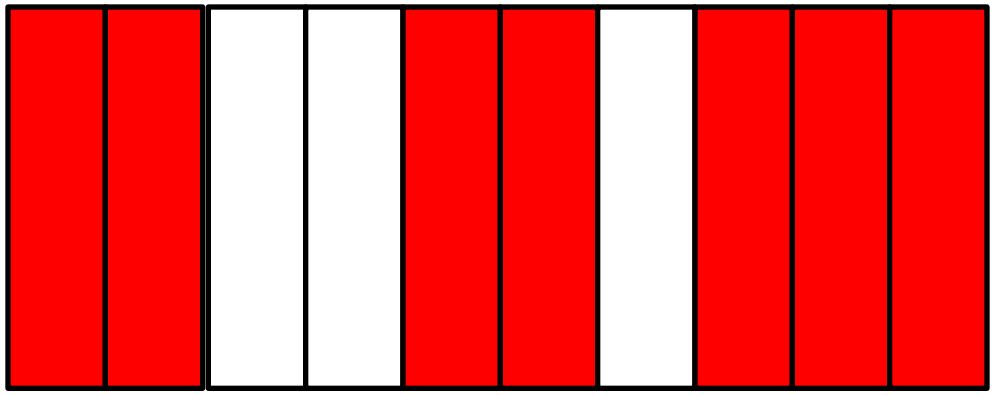
DDRRT SPR - How to Map a Position from a Square to a Trapezoid?

Question: How to map a point in space from a cube to a trapezoidal prism?



Answer: through the use of shape functions, transformations used in finite elements

What is the DDRRT PDR?

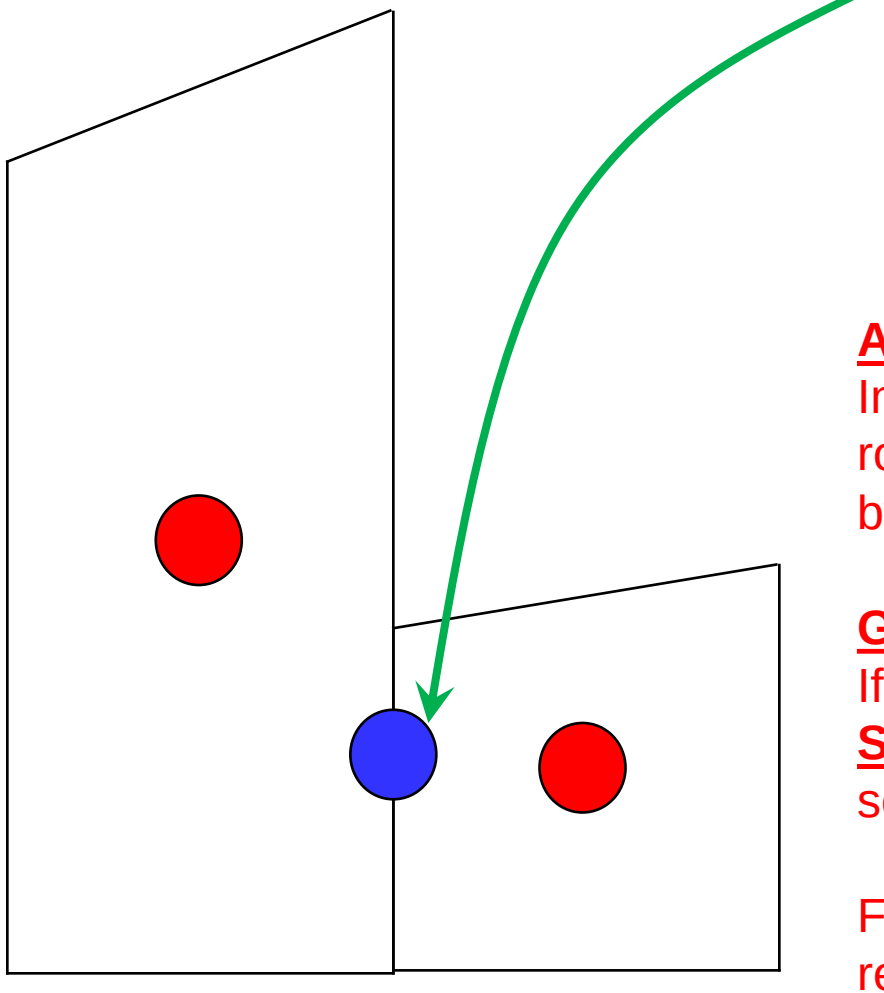


Last consecutive collisions = 3

→ goal biasing shaped to 70%

- Punctual Dispersion Reduction (PDR) = adaptive goal-biasing
  - Consecutive collision decrease the biasing
  - consecutive non-collision increase the biasing in order to target the goal
- Based on linear probability shaping considering successive collisions (3 consecutive collisions = reshaping of goal-biasing by -30%)
- Shaping the probability distribution for a waypoint to be chosen for extension does not guarantee the waypoint will be chosen (unless the biasing is 100%)

How SDR and PDR Work Together? Example



Assume a previously undetected obstacle (or enemy hot zone) is located at the blue point between 2 waypoints→ PDR WILL decrease

Algorithm

Instead of immediately querying alternative routes, try to go around first → PDR bounded by SDR

GOING AROUND THREAT

If PDR decreases below 50% THEN rely on **SDR only** in order to reach the cell where the second centroid (red) lays

From that point where SDR has helped to reach cell 2, check number of seeds spent before trying to reroute using STA

How SDR and PDR Work Together - Algorithm

1. Follow flyable waypoints until danger happens
2. Adjust PDR until it falls below 50% of biasing
3. IF below 50%, switch to SDR only, allowing to jump to the next cell (if possible)
4. Once working in the new cell, attempt at replaning trajectory (STA Network + Dubins & ODE if Dubins condition not met)
5. possible, toggle back PDR for next WP

Watch the MATLAB Video

Conclusion

Offline (before flight) Contribution

This reorganization of height maps into stratified tessellated two-dimensional obstacle-free trapezoidal flying zones. The end product is a Synthetic Terrain Avoidance (STA) Network.

Online (real-time, in-flight) Contribution

The second original contribution of this research to the field is the development of new type of Rapidly - exploring Random Tree using shape functions for point mapping onto the trapezoidal flying zones, and STA Network for Smart Rerouting