Autonomous Path Planner for Aerial Photography

William Betteridge

Contents

Project Outline

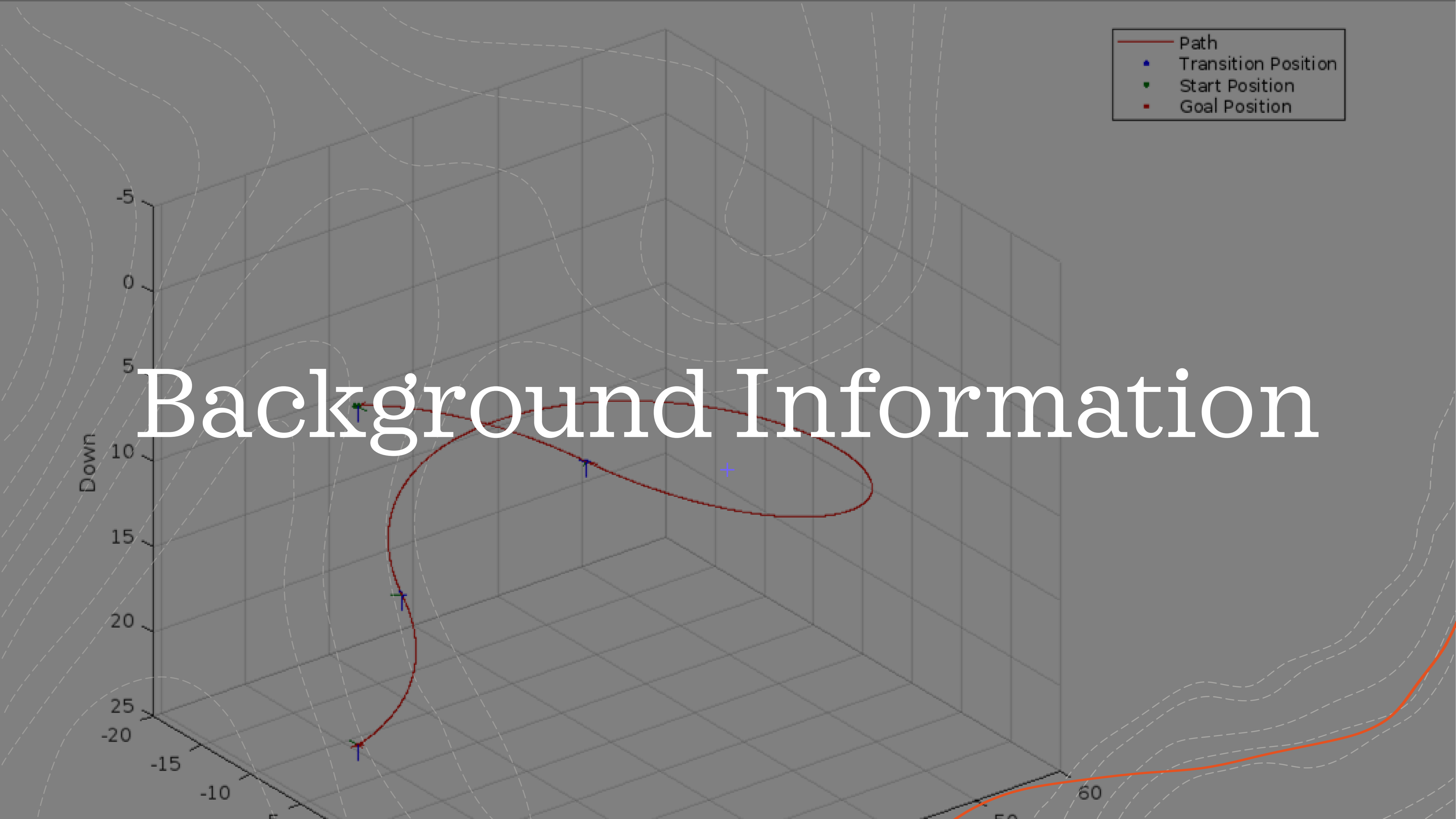
Calculate UAV flight path for aerial photography

Reduce user interaction required to set up a flight mission

Account for 3D (altitude changes)

Consider several environmental constraints

Background Information

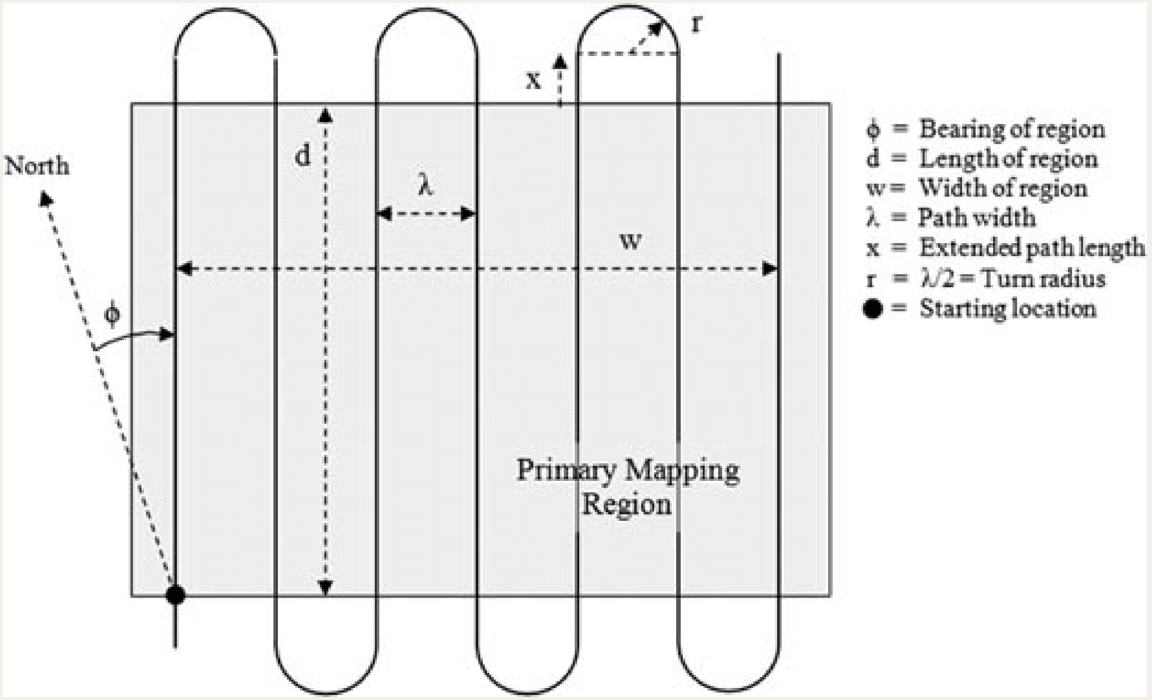


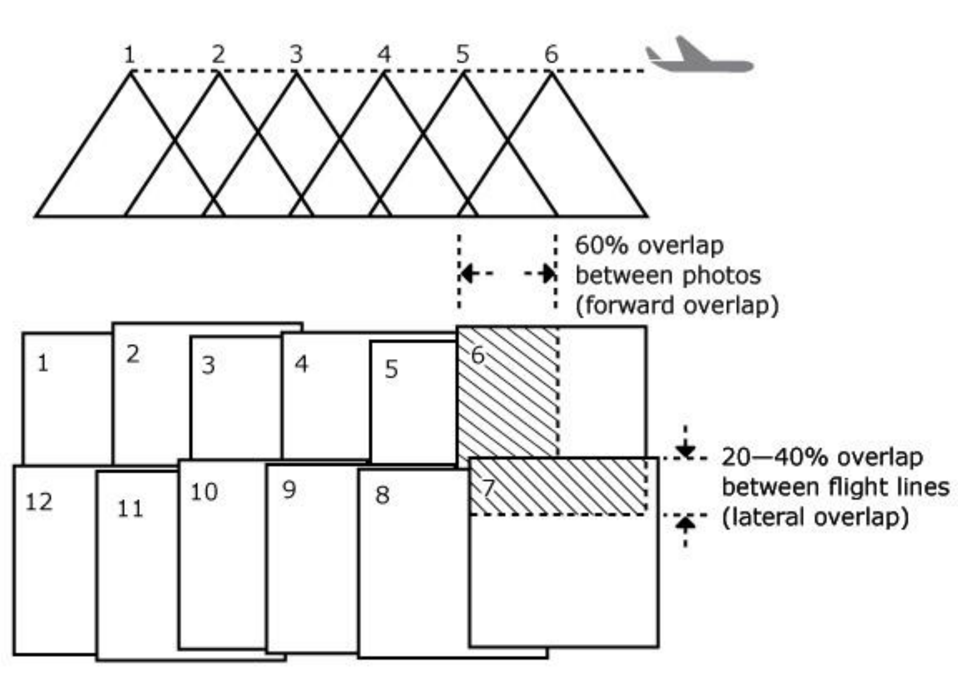
Unmanned Aerial Vehicles (UAVs)

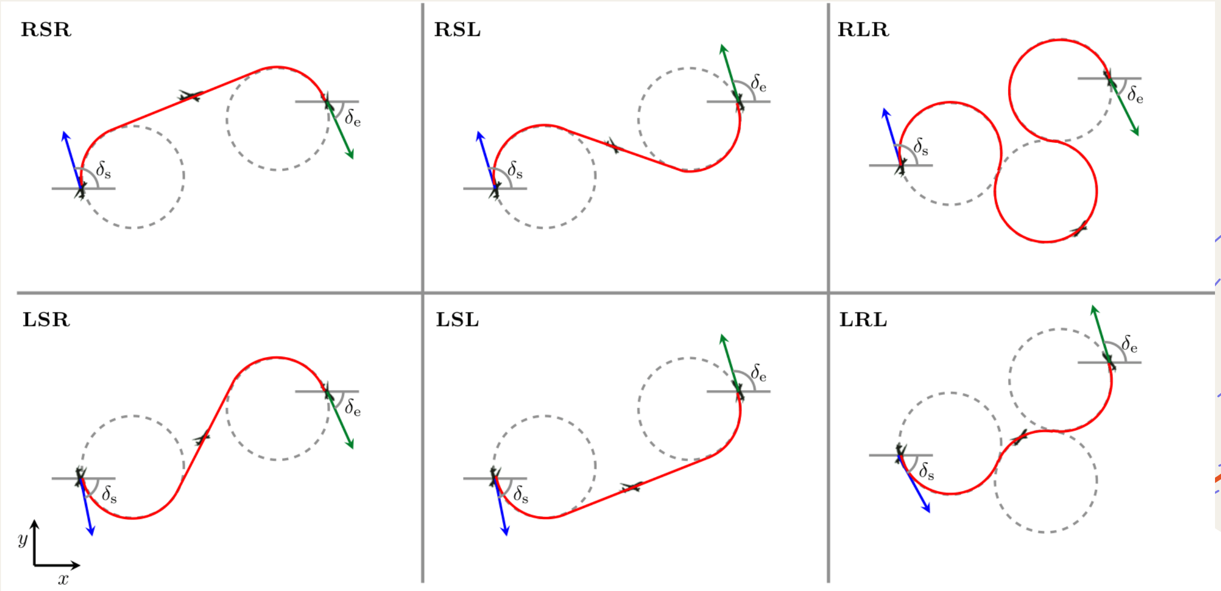
- Small aircraft flown without onboard pilot, controlled remotely
- Used in navigation, photo surveillance, combat, transportation etc.
- Rotary-wing:
  - Generate lift by forcing air downwards by rotating the wings
  - Navigate tight environments
  - Remain stationary mid flight
- Fixed-wing:
  - Generate lift via large wingspan & small propellor
  - Can carry large payloads e.g. cameras for surveillance
  - Longer battery life

UAV Path Planning

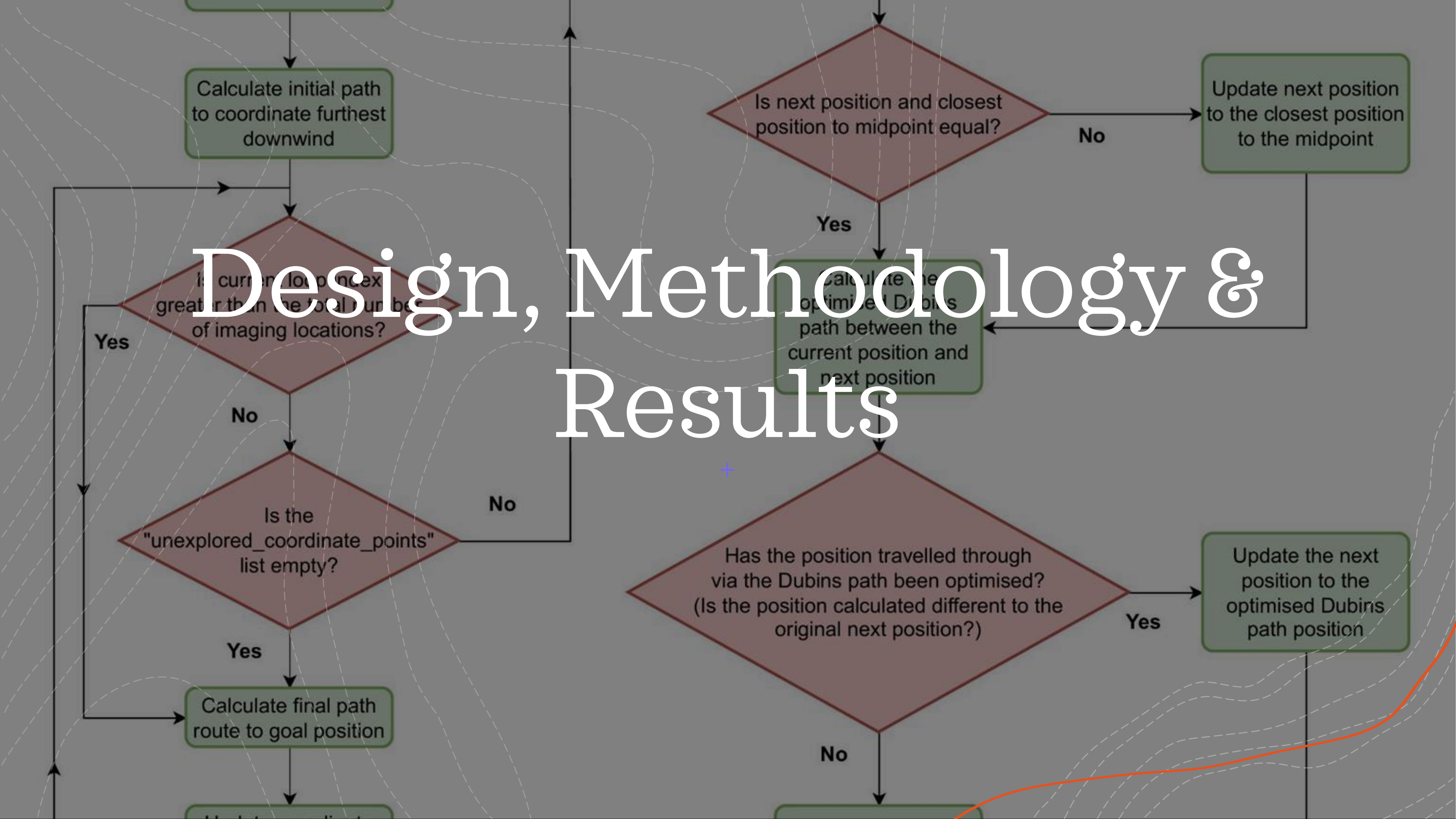
+ Lawnmower path shape
+ Dubins path turns
+ Image overlap requirements
+ UAV height AGL



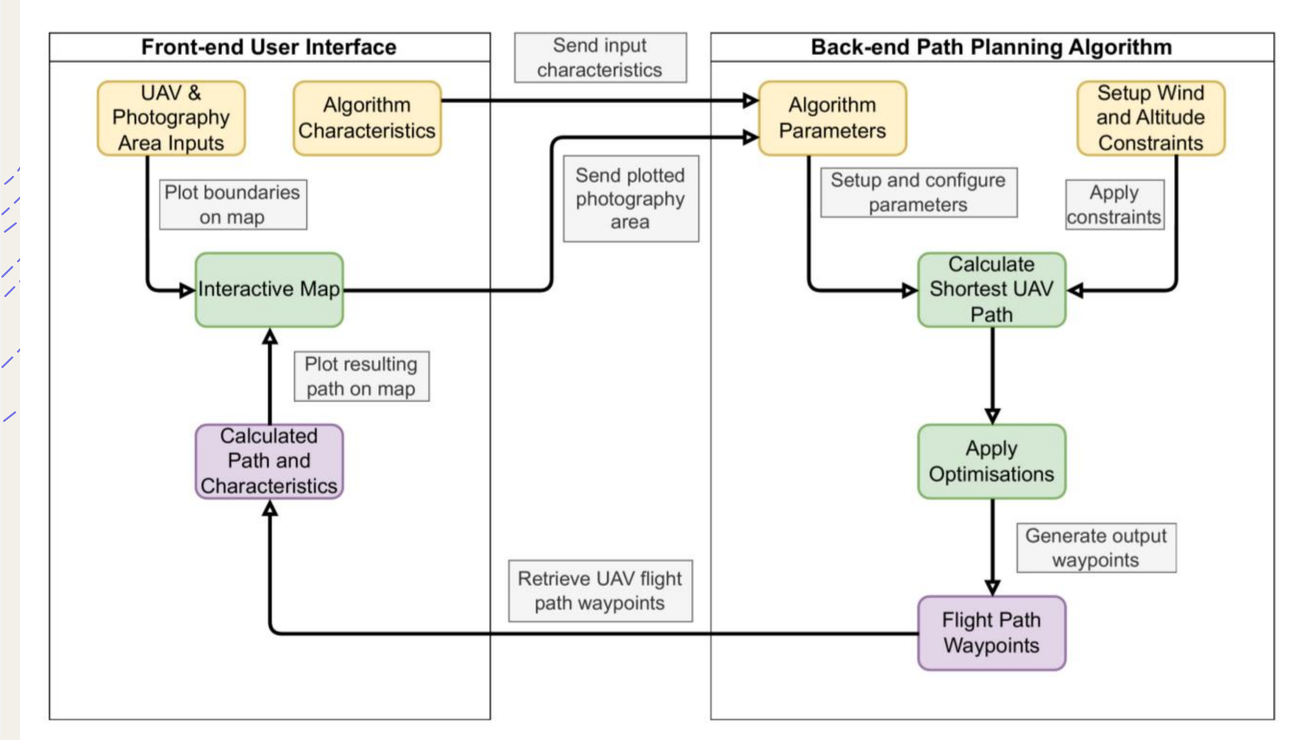


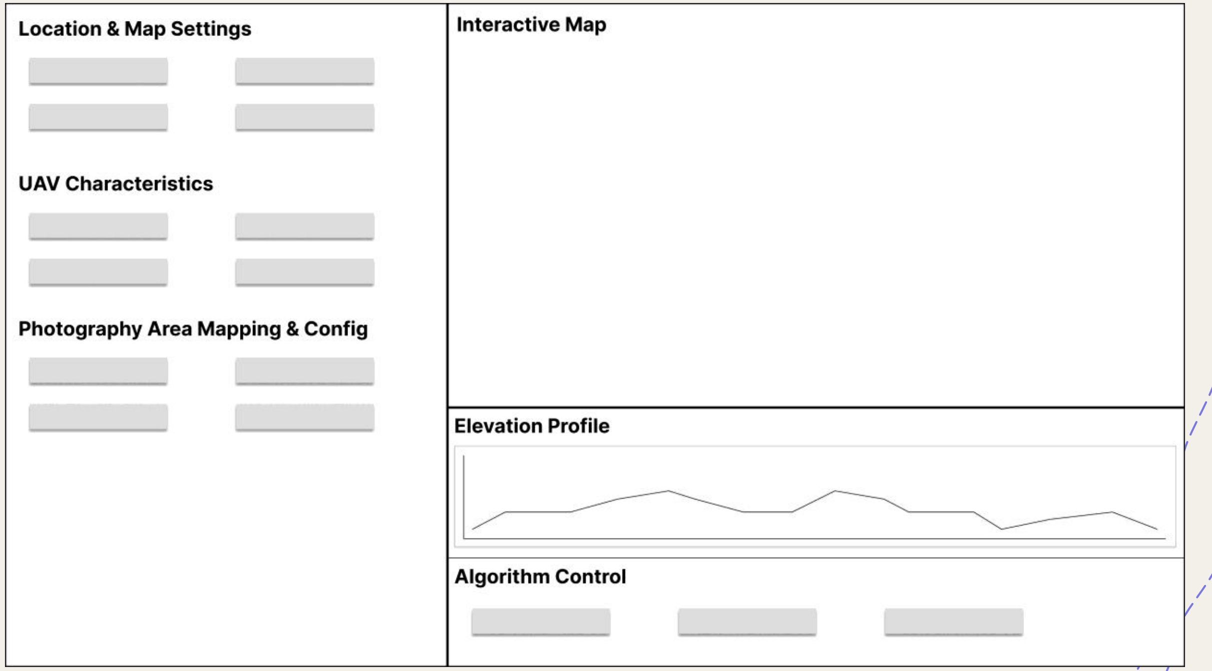


Design, Methodology & Results



Design Overview



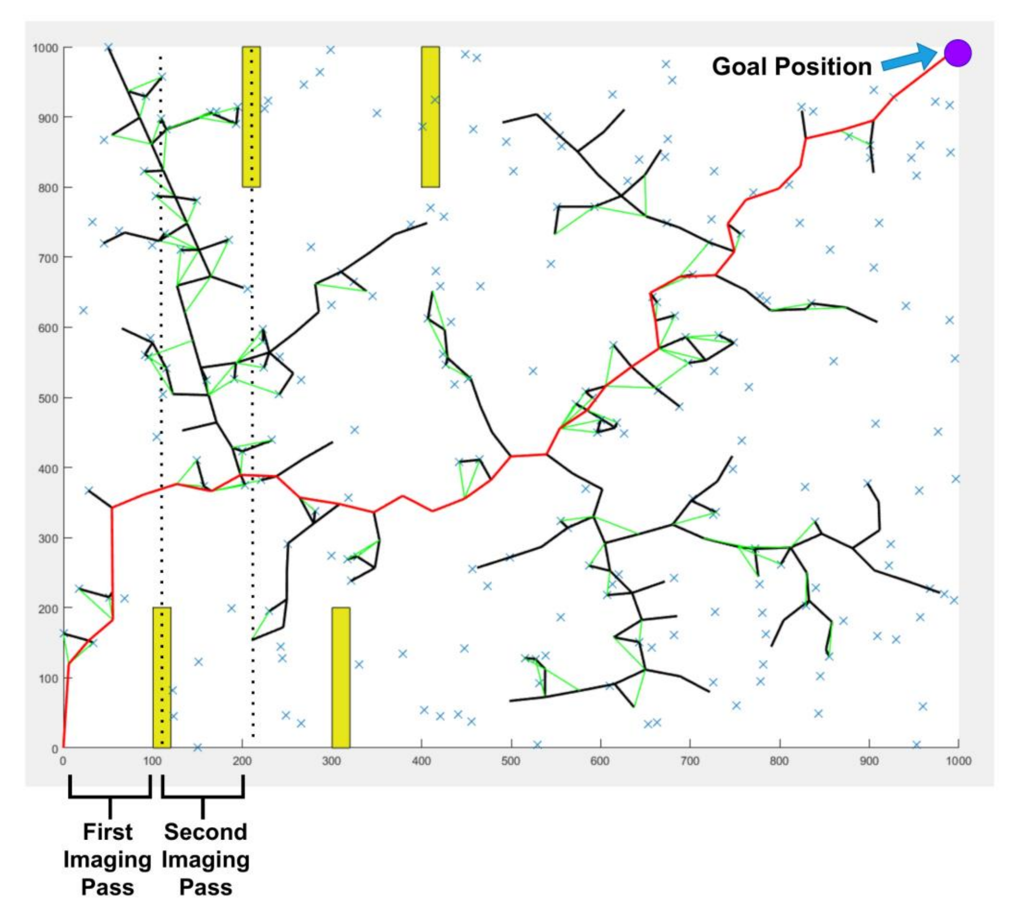


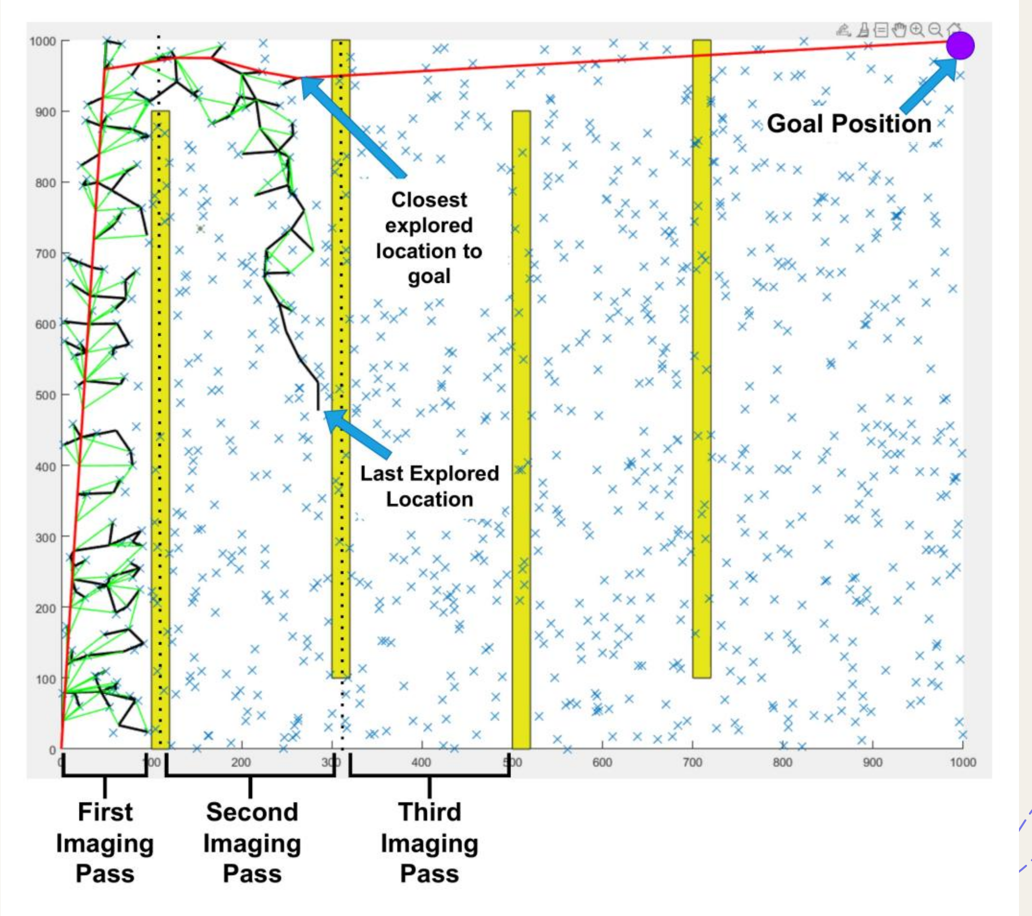
Key
Input
Output
Process

+System Architecture

+GUI Mockup

Rapidly Exploring Random Tree (RRT)

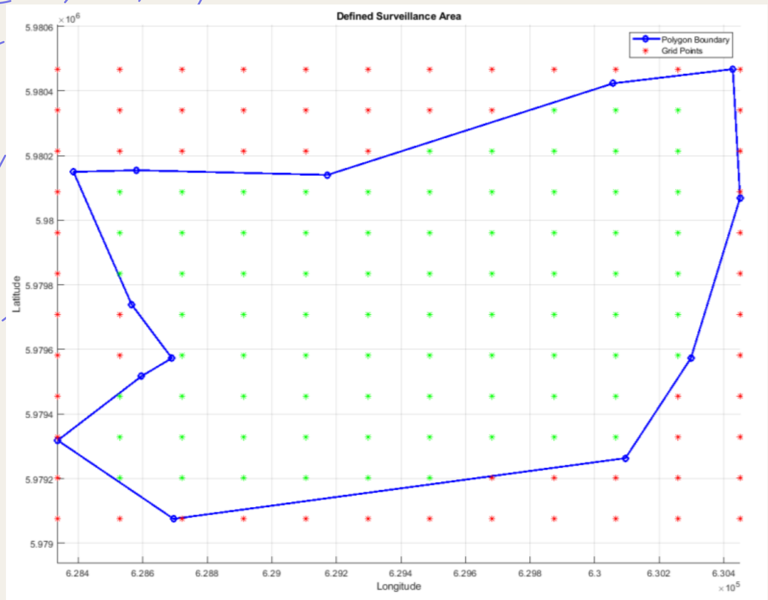


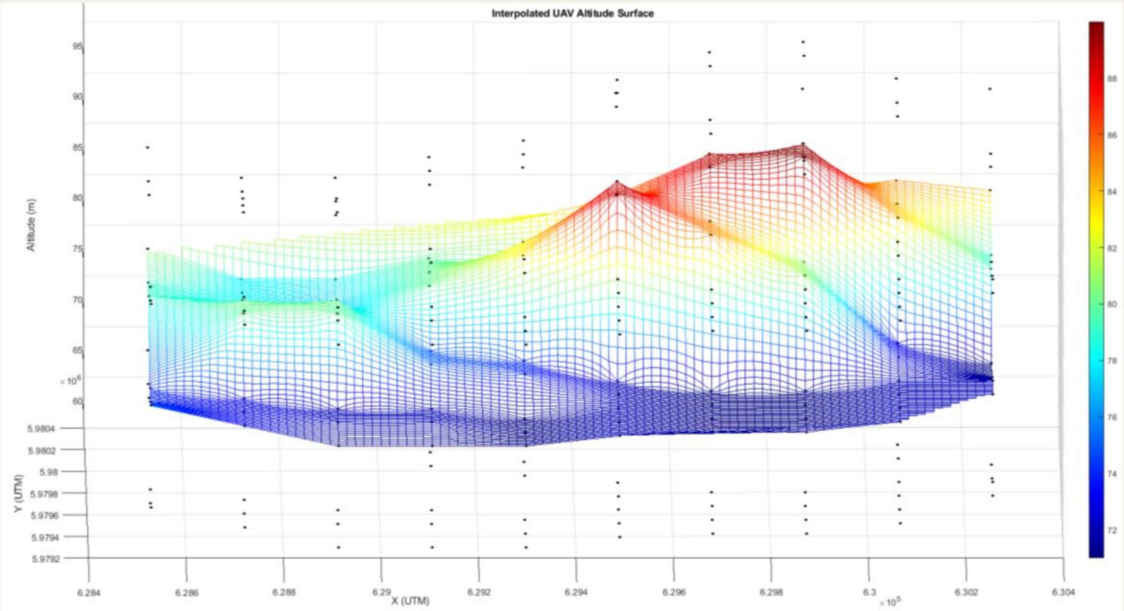


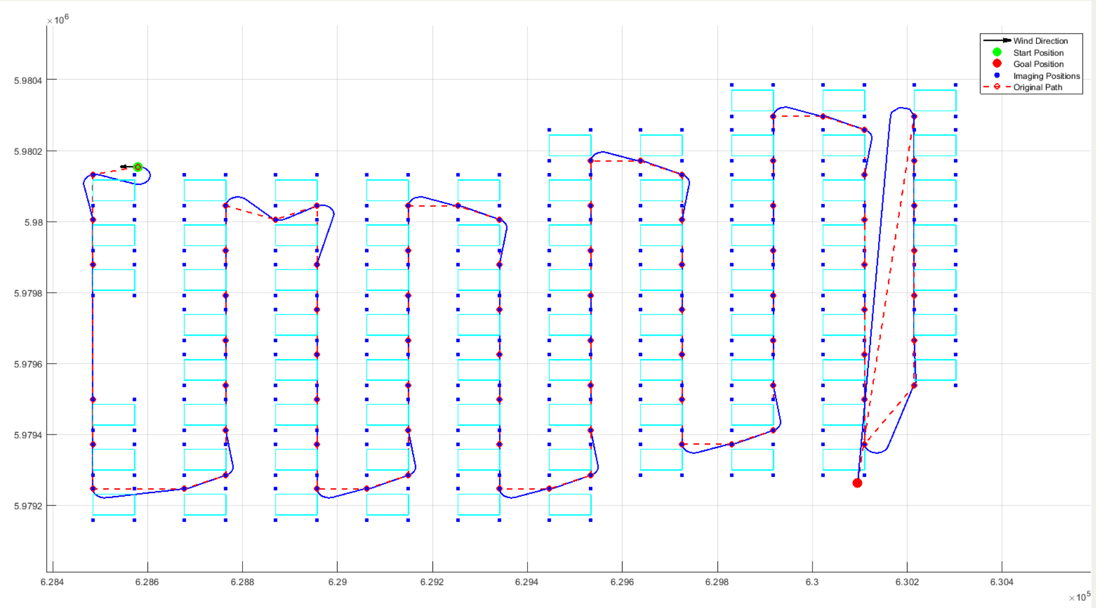
+ Short obstacles don't provide enough "guidance"

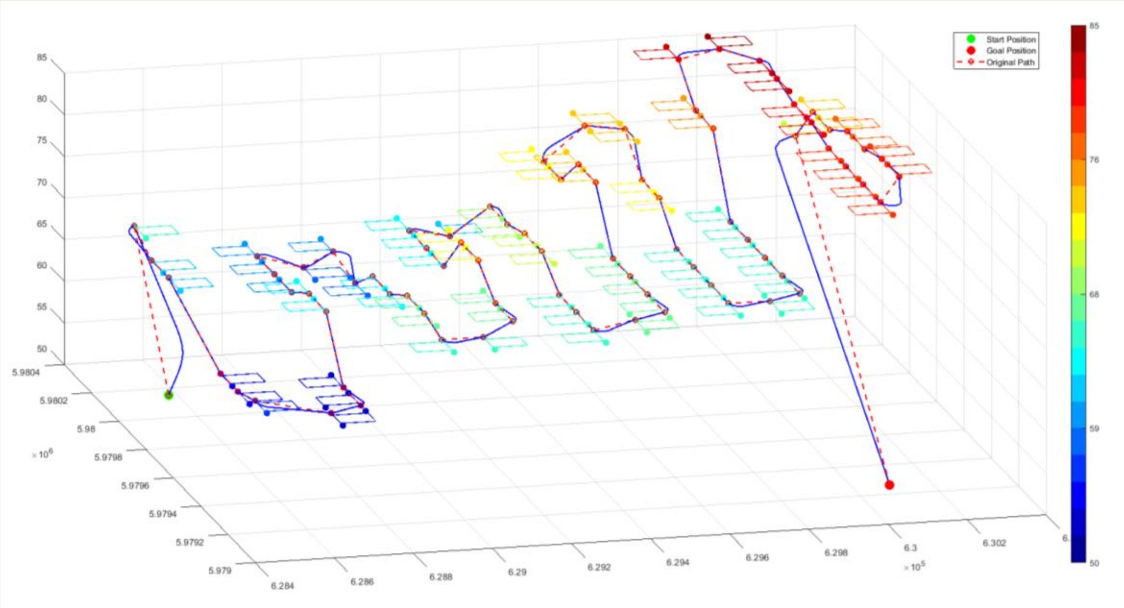
+ Long obstacles prevent algorithm from pathfinding

TSP Nearest Neighbour

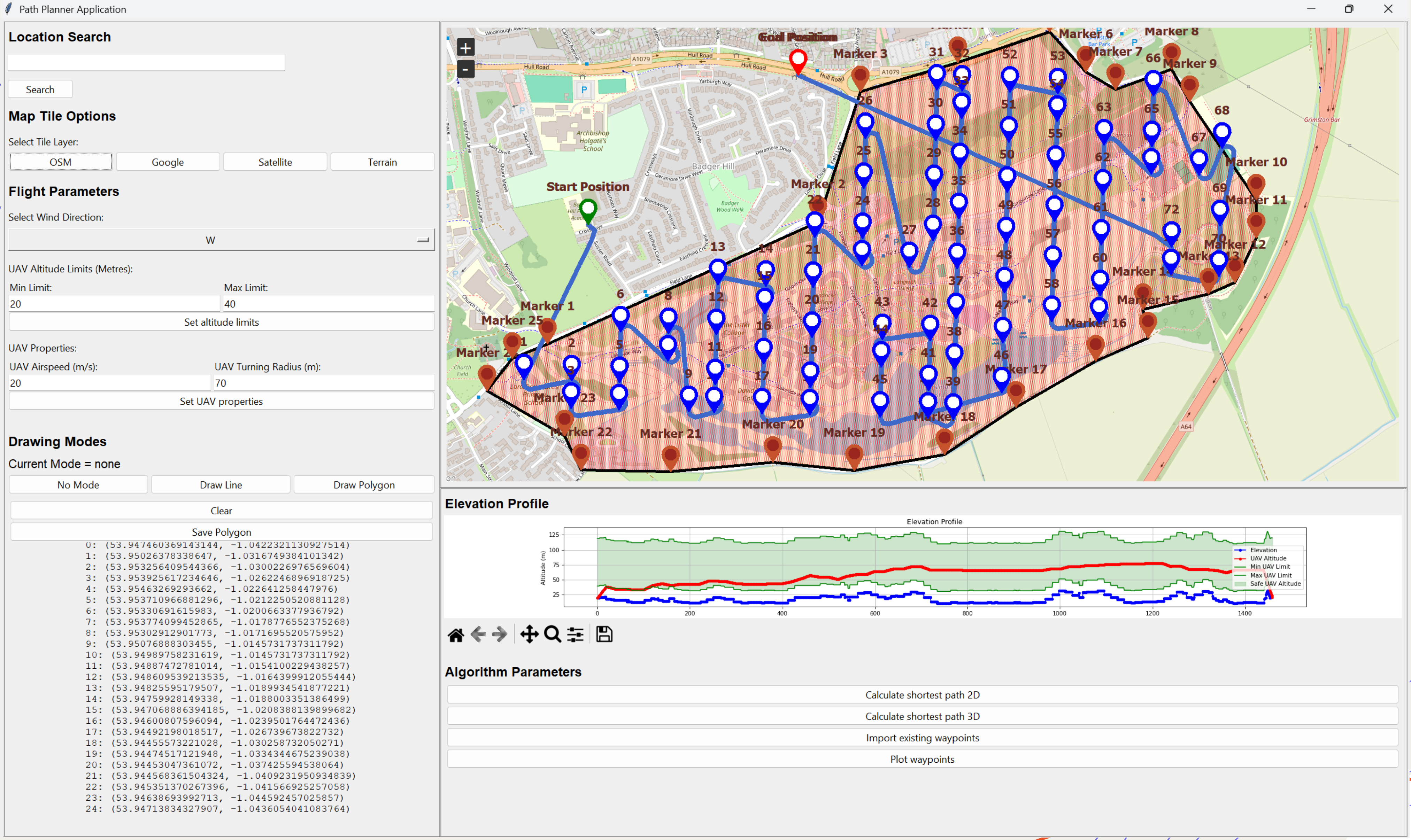




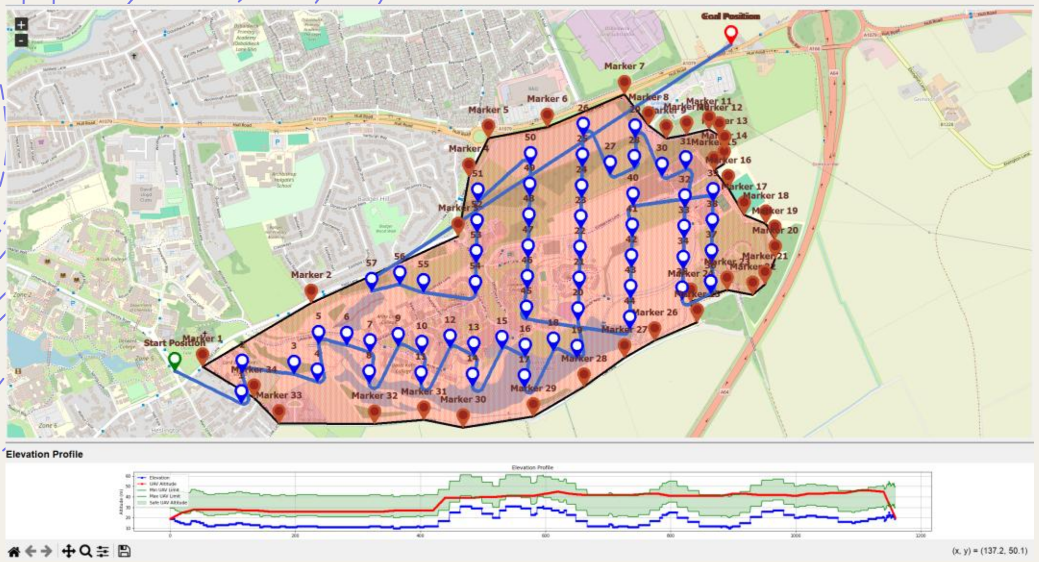


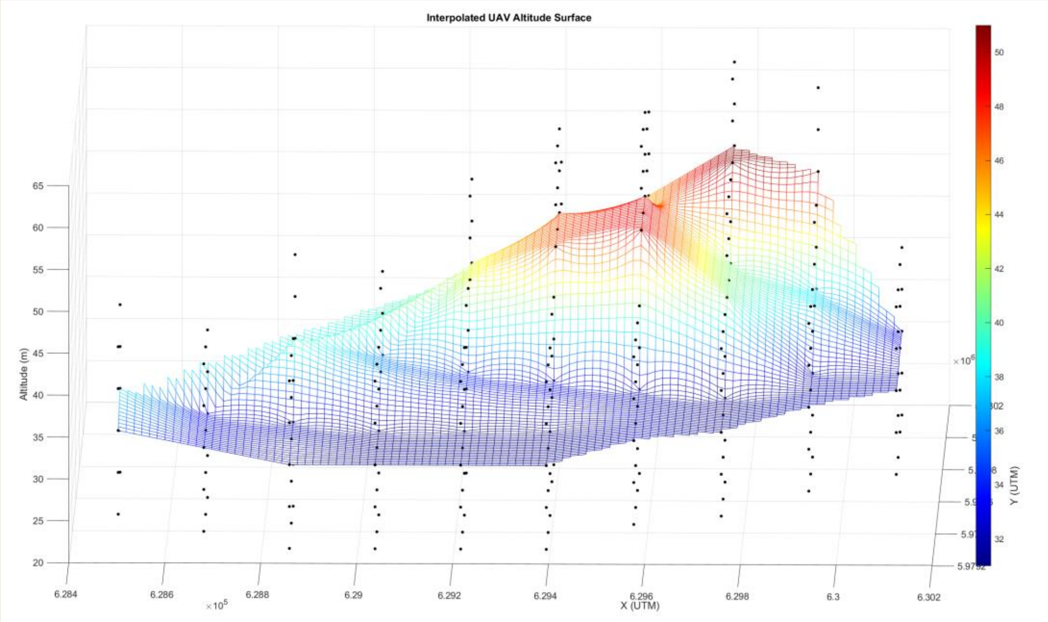


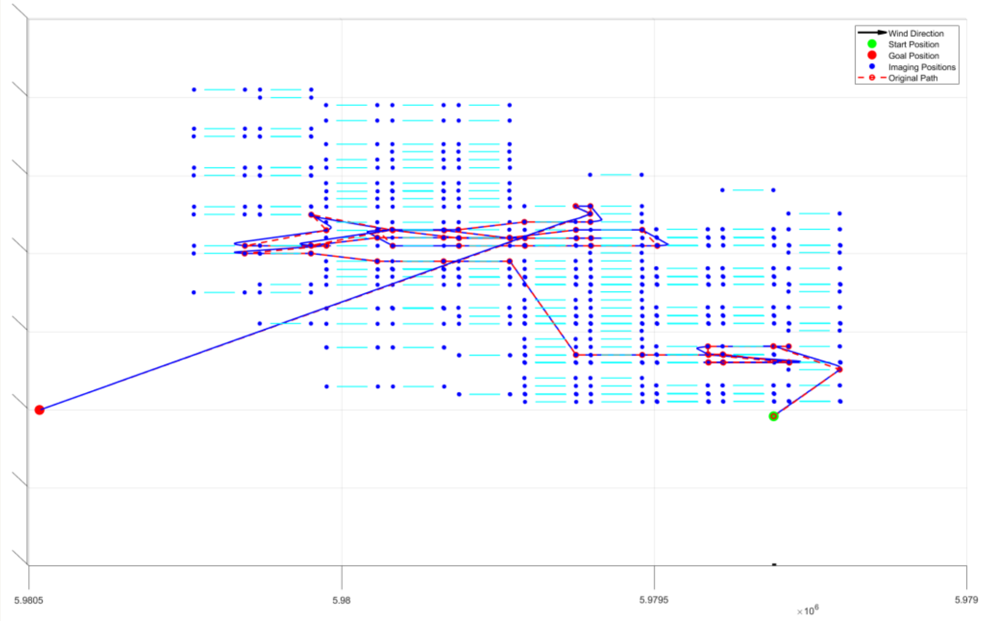
User Interface & Mapping

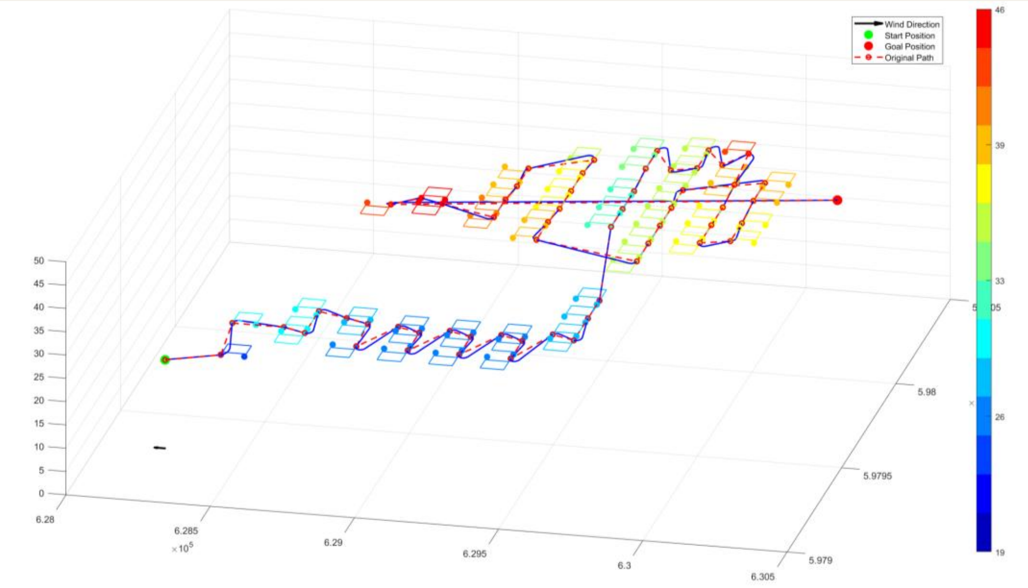


Campus East Results – Flatter Terrain

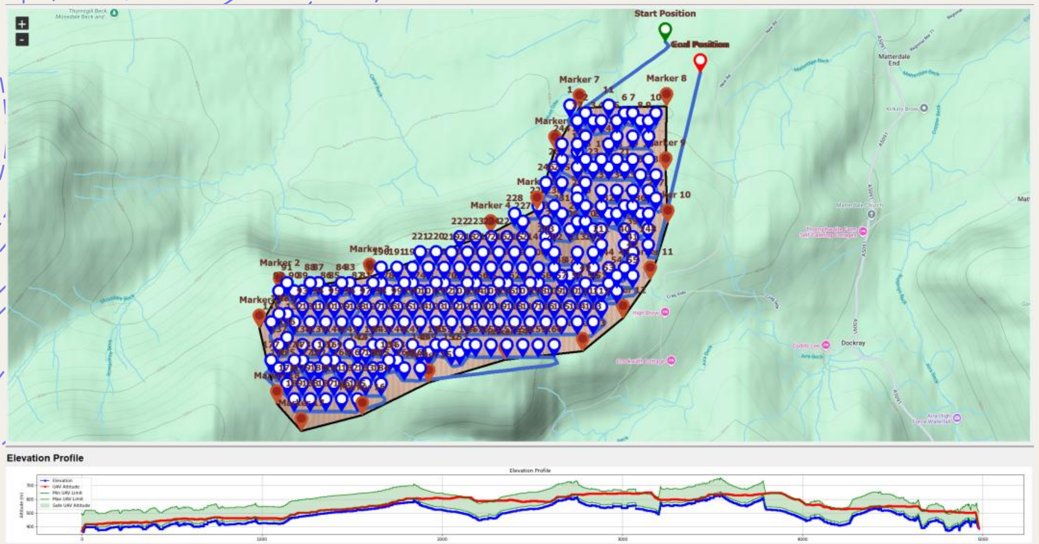


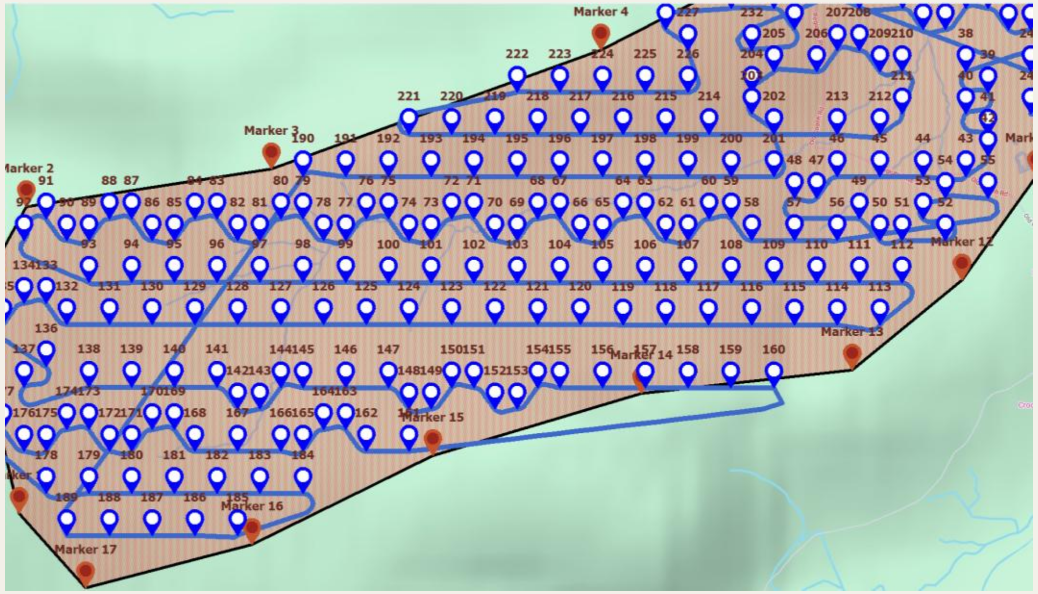


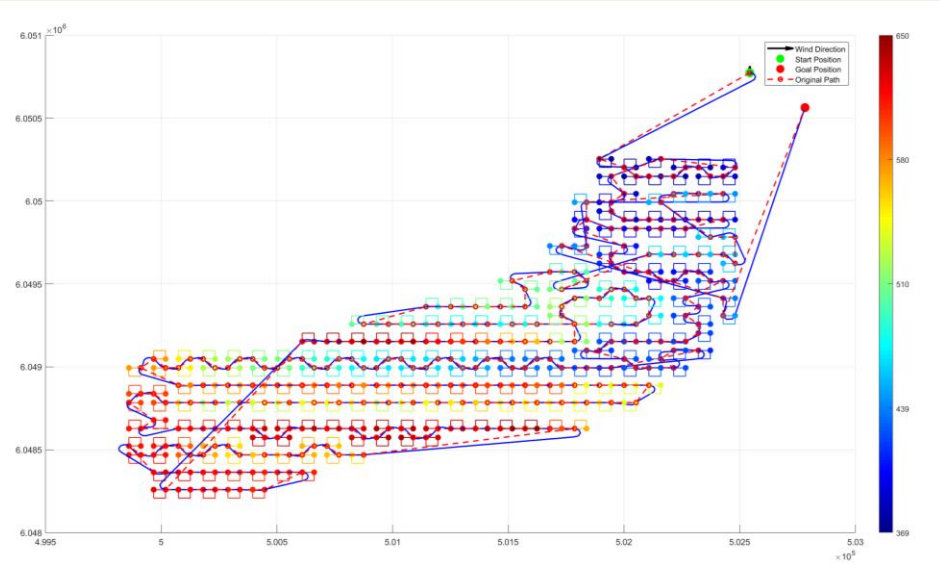


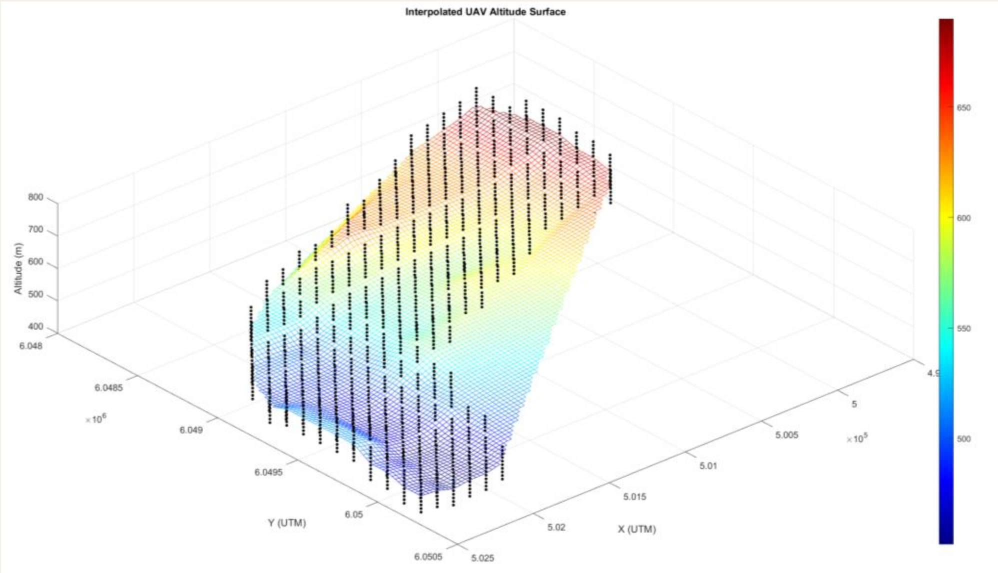


Groove Beck Results – Hilly Terrain









Conclusions & Further Work

Conclusions

Successful Outcomes

- Achieved core objectives.
  - 3D flight path with constant height AGL
  - Intuitive UI
  - Environmental constraints implemented
- Robust proof-of-concept for autonomous 3D path planning for aerial photography.
- Successfully reduces user interaction required to fly a UAV.

Problems & Drawbacks

- Calculates flight path using one iteration, so does not calculate *the* most efficient path
- No battery cycle management and landing predictions

Further Work

- Improve TSP heuristics – look ahead more than 2 points
- Battery life estimation – notify user when battery is expected to run out and land to swap battery and continue flight.

Thank You

Any Questions?