NavX:
Final Design Review

Team:
Alvin O'Garro
Jacob Jeong
Antony Samuel
Kartik Sastry

Final Design Review Scope

System Overview
- –Concept of Operations (CONOPS)
- –System Integration/Usage
- –Mission Level Description
- –System Block Diagram
- –Interfaces
- –Success Factors & Verification

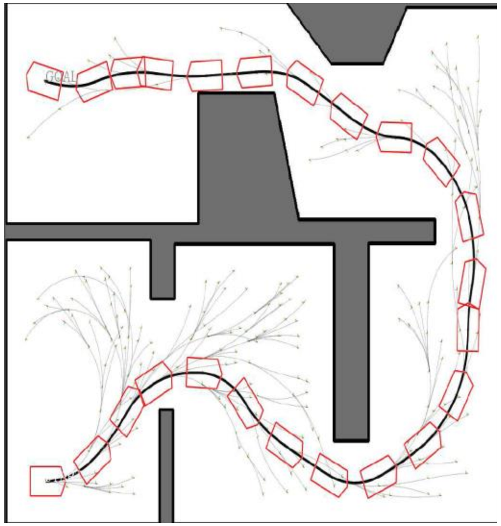
Requirements
- –Requirements Allocation
- –Performance Metrics and Analysis

Implementation
- –System Trade Offs
- –Software Design Analysis
- –Risk Analyses
- –System Performance
- –Schedule

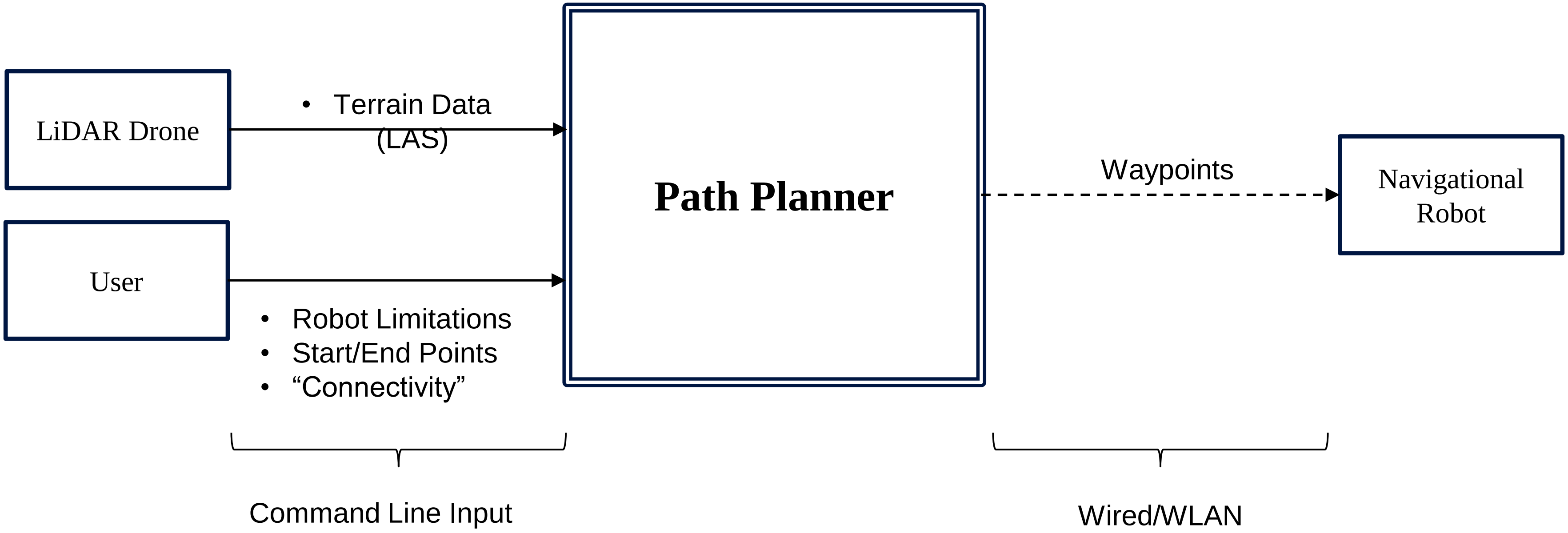
Conclusions
- –Results
- –Future Work

Concept of Operations (CONOPS)

- **System Purpose / Objective**
  - Plan a ’high-level’/’global’ path for an autonomous vehicle through austere terrain
    - Minimize the total distance travelled
    - Provide waypoints (sequence of coordinates) that define the path
    - Account for mobility limitations of the particular autonomous vehicle
- **System User**
  - Operators of the autonomous vehicle
    - In ECE 4012 – Design Team B
    - Outside of ECE 4012: Harris Corp. and/or potential clients



System Integration/Usage

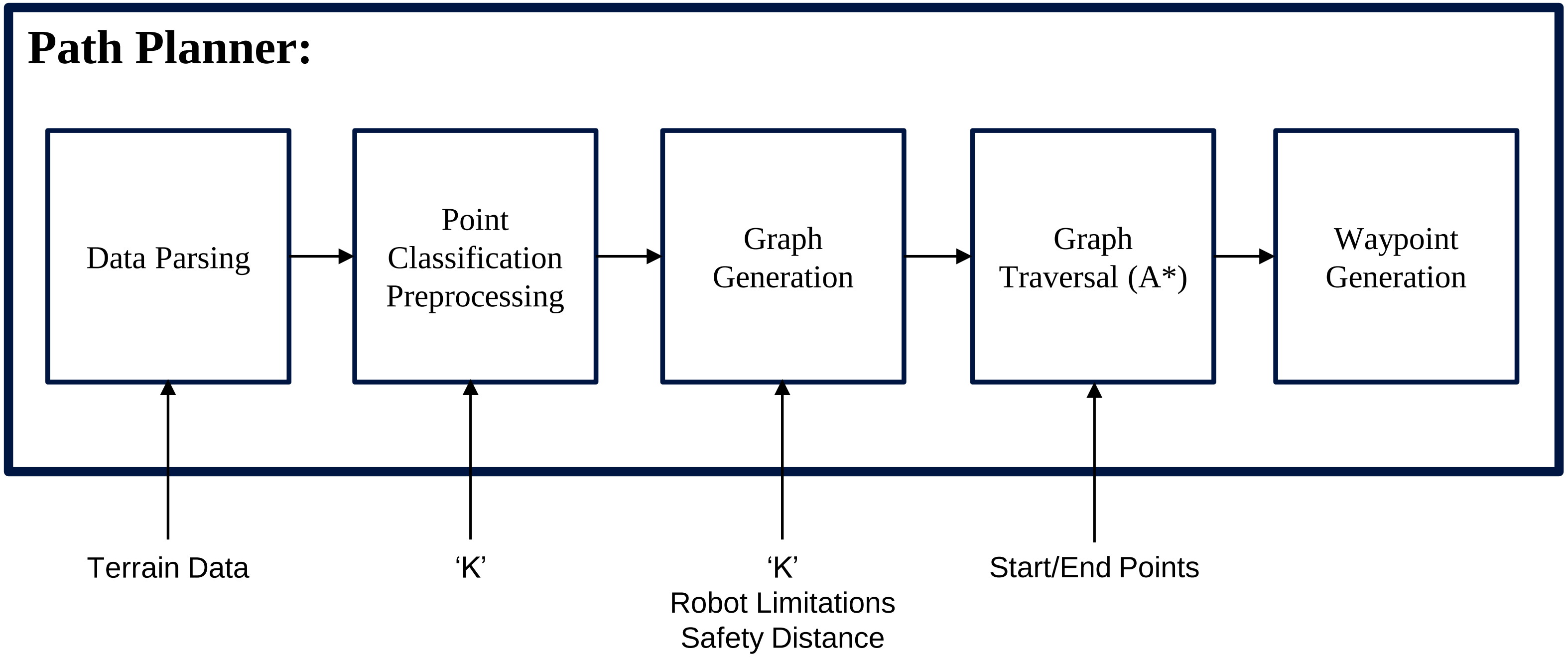


Mission-Level Description

HARRIS®

- **Inputs**
  - "Terrain Data"
    - An .las file characterizing the terrain
      - Relevant components: 3D point cloud + point classification
      - Reference coordinate system + units from file metadata.
  - "Robot Limitations"
    - Maximum tilt the robot can experience
      - In direction of motion and laterally
    - Limitations on the type of terrain the robot can drive in
      - Water, mud, etc.
  - "Start, End Locations"
    - Provided either as coordinates in the reference coordinate system, or by identifying points in the .las file.
      - If an exact node is not specified, the closest node (by Euclidean distance) will be selected.
  - "Connectivity Parameter"
    - The graph representation connects each node to its k-nearest neighbors. We leave integer k as an optional user parameter, with a default value to be set later.
- **Output**
  - Waypoints
    - Of the form: $\{(x_0, y_0, z_0), (x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)\}$
      - N x 3 array of `double/float` 's, or a file.
      - Will communicate with Team B to transform this into the appropriate reference frame.

System Block Diagram



Interfaces

- **External Interfaces - With User**
  - Input data must adhere to the LAS standard
  - Output waypoint format - list of (x, y, z) tuples
    - Allows flexibility for user to localize in 2D or 3D
- **Internal Interfaces - Data Exchange Between Software Components**
  - **Parser - Preprocessing Interface:**
    - LAS File parsed into array of Point objects and input to classifier
  - **Preprocessing - Graph Generation Interface:**
    - Preprocessing maintains the representation of points, passes to graph generation
  - **Graph Generation - Graph Traversal Interface:**
    - A graph
  - **Graph Traversal - Waypoint Generation Interface:**
    - Graph Traversal (A*) outputs a list of nodes. Waypoint Generation is a coordinate transformation.

Critical Success Factors

HARRIS®

- **What is important to the Customer:**
  - **Schedule**: Project completed by Senior Design Expo ✓
  - **Technical**: Project successfully generates and illustrates waypoints ✓
  - **Sponsor Relationship:** Maintaining Harris communication for project direction ✓
  - **Systems Engineering:** Maintaining System Block Diagram ✓
  - **Engineering Management:** Assigning a Responsible Engineer for every requirement and subsystem ✓

- **Methods of Technical Evaluation**
  - LiDAR data parser output validated with Geographic Information System (GIS) Software ✓
  - Employ an existing path-planning tool to verify generated path (verified)
  - 3D Visualization ✓
    - Illustrates terrain and calculated waypoints (sanity check)
    - Visually verify generated path conformance with robot limitations

Requirements Allocation

(Formation of sub-tasks): See System Block Diagram

Data Parsing

- The project shall correctly read .las file and convert data fields to usable data

Point Classification Preprocessing

- The project shall remove unnavigable points (ie. canopy, water), k-nearest-neighbors classification of unclassified points

Graph Generation

- The project shall generate graph representation for points

Graph Traversal (A*)

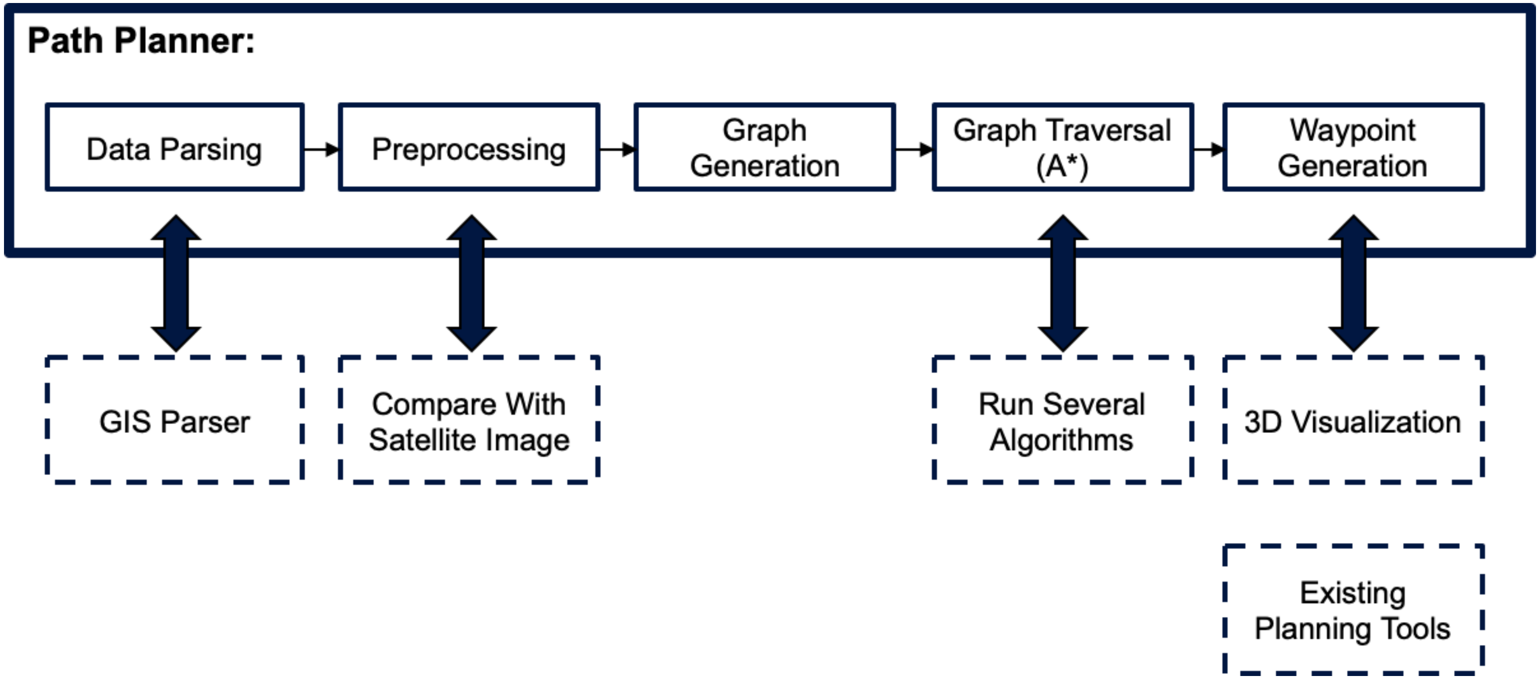
- The project shall compute shortest distance path from user-specified start/end points

Waypoint Generation

- The project shall transform graph nodes into geographical coordinates

Performance Metrics and Analysis

- **"How good is the path?"**
  - The project shall characterize the 'deviation' of our path from an ideal or reference path
- **"How long does it take to compute?"**
  - *Note: Important if runtime becomes critical*
- **Subsystem validation depicted below:**



System Trade-offs

- **Significant Tradeoffs**
  - Spatial density of data vs computation resources.
    - Higher spatial density allows finer path control and better terrain representation, but requires a more memory intensive graph.
  - Path optimality vs computation time.
    - More accurate paths can be generated, but at the expense of computation time.

Software Design Analysis

- **Critical Timing**
  - No timing constraint
- **Memory Utilization**
  - Point storage requirements alone scale linearly with *k*
    - Bulk of memory required
    - Points stored as (X, Y, Z, Class): 3*sizeof(float) + sizeof(int)
    - Nodes stored as (Point, Point[k]): (*k*+1)*sizeof(Point)
  - On the order of MB for 100,000 points in .las file
- **Processor Loading**
  - Single Core - Single Thread

Risk Analyses

- Since the project is purely software-related, all associated risks were results of software malfunctions/miscalculations.

- **Risk 1:** *The generated waypoints guide the robot through unnavigable terrain or unseen obstacles*
  - User maintains option to go into “manual mode”

- **Risk 2:** *Map generation takes longer than preferred*
  - User inputs a ‘K’ values correlating to path options
  - Downsample input data

System Performance

- System performance reached our initial goals. Specifically, our process for generating waypoints effectively provided appropriate coordinates for the navigating robot

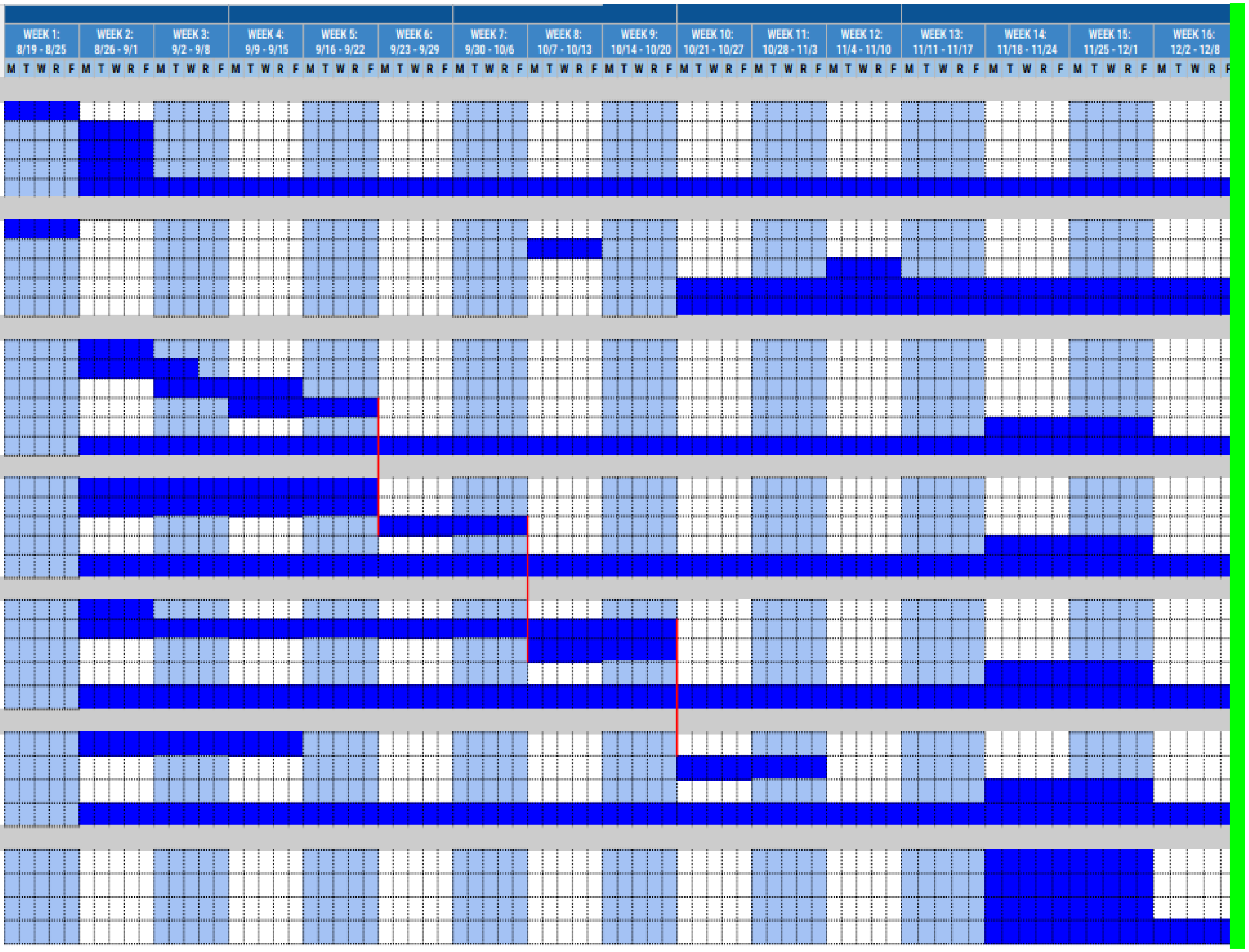
- **Performance Analysis:**
  - **Produce Output**
    - Input LiDAR data in LAS format
    - Parse LiDAR data to get point coordinates/point classification information
    - Input point array into path-planning algorithm
    - Produce waypoint Array

- **Evaluate**
  - Illustrate Waypoints using 3D Visualizer
  - Overlay Waypoints on top of terrain visualization for path verification
  - Compute deviation from a baseline / ideal path, if one exists

Schedule

HARRIS®

TASK NUMBER	TASK TITLE	TASK LEAD	START DATE (Week)	DURATION (Weeks)
1	*ECE 4012 Milestones*			
1.1	Integrate Advisor Feedback	All	1	1
1.2	Oral Presentation	All	2	1
1.3	Revise Proposal	All	2	1
1.4	Revise Project Summary	All	2	1
1.5	Website Maintenance	AO	2	15
2	*Harris Corp. Milestones*			
2.1	Preliminary Design Review	All	1	1
2.2	Critical Design Review	All	8	1
2.3	Technical Readiness Review	All	12	1
2.4	PowerPoint Presentation	All	10	6
2.5	White Paper	All	10	6
3	*Data Acquisition, Preprocessing with GIS (Lead: AS, JJ)*			
3.1	Preprocessing/Generate Point Cloud Data	JJ	2	1
3.2	Extract Point Classification Information	JJ	2	1.5
3.3	Acquire and Fuse Additional Data	AS	3	2
3.4	Transfer Data to Implementation Level	AS	4	2
3.5	Create Visualizations for Presentations	JJ, AS	14	2
3.6	Documentation	JJ, AS	2	15
4	*Developing a Representation (Lead: KVS)*			
4.1	Develop Efficient Storage, Access, and Update Mechanisms	KVS	2	4
4.2	Develop Code to Populate Representation	KVS	2	4
4.3	Evaluate Against Performance Specifications, Revise	KVS	6	2
4.4	Create Visualizations for Presentations	KVS	14	2
4.5	Documentation	KVS	1	15
5	*Implementing Path Planning Algorithm (Lead: AO)*			
5.1	Define Interface Between Representation and Algorithm	KVS, AO	2	1
5.2	Develop and Refine Implementation	AO	2	8
5.3	Evaluate Against Performance Specifications, Revise	AO	8	2
5.4	Create Visualizations for Presentations	AO	14	2
5.5	Documentation	AO	2	15
6	*Integration and Output (Waypoints) Generation*			
6.1	Reach Agreement with Harris Team B on Waypoint Information	All	2	3
6.2	Produce Waypoints at Agreed Upon Resolution/Format	All	10	2
6.3	Create Visualizations for Presentations (Block Diagram?)	All	14	2
6.4	Documentation	All	2	15
7	*Testing and Expo Preparation*			
7.1	Test With Several Sets of Data	All	2	14
7.2	Evaluate Against Performance Specifications	All	2	14
7.3	Document Testing Results	All	2	14
7.4	Develop Expo Presentation Materials	All	3	14



Results

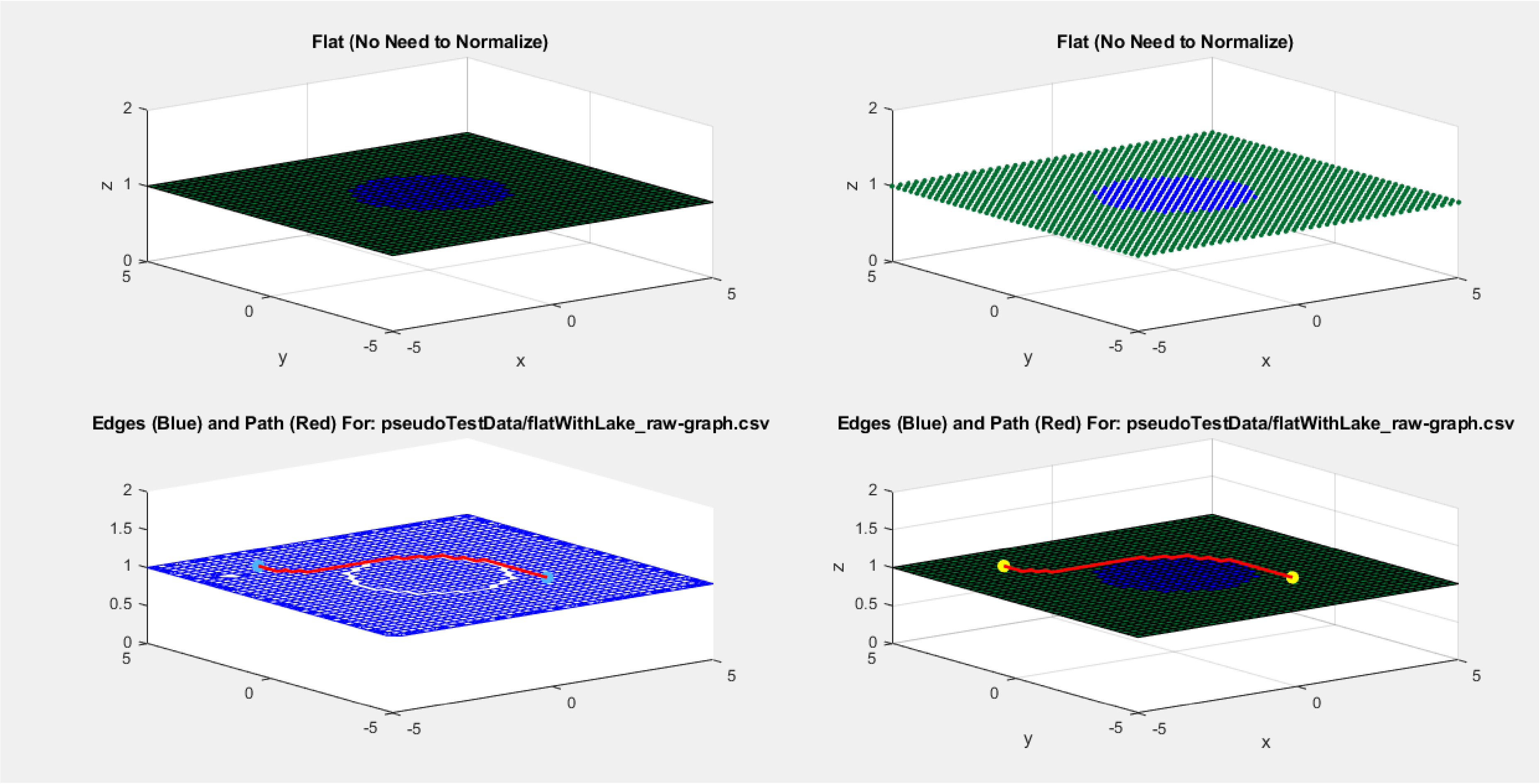
Intuitive Test Cases

- Generated a variety of “intuitive” test cases
  - Sanity check for common scenarios
  - Useful in debugging

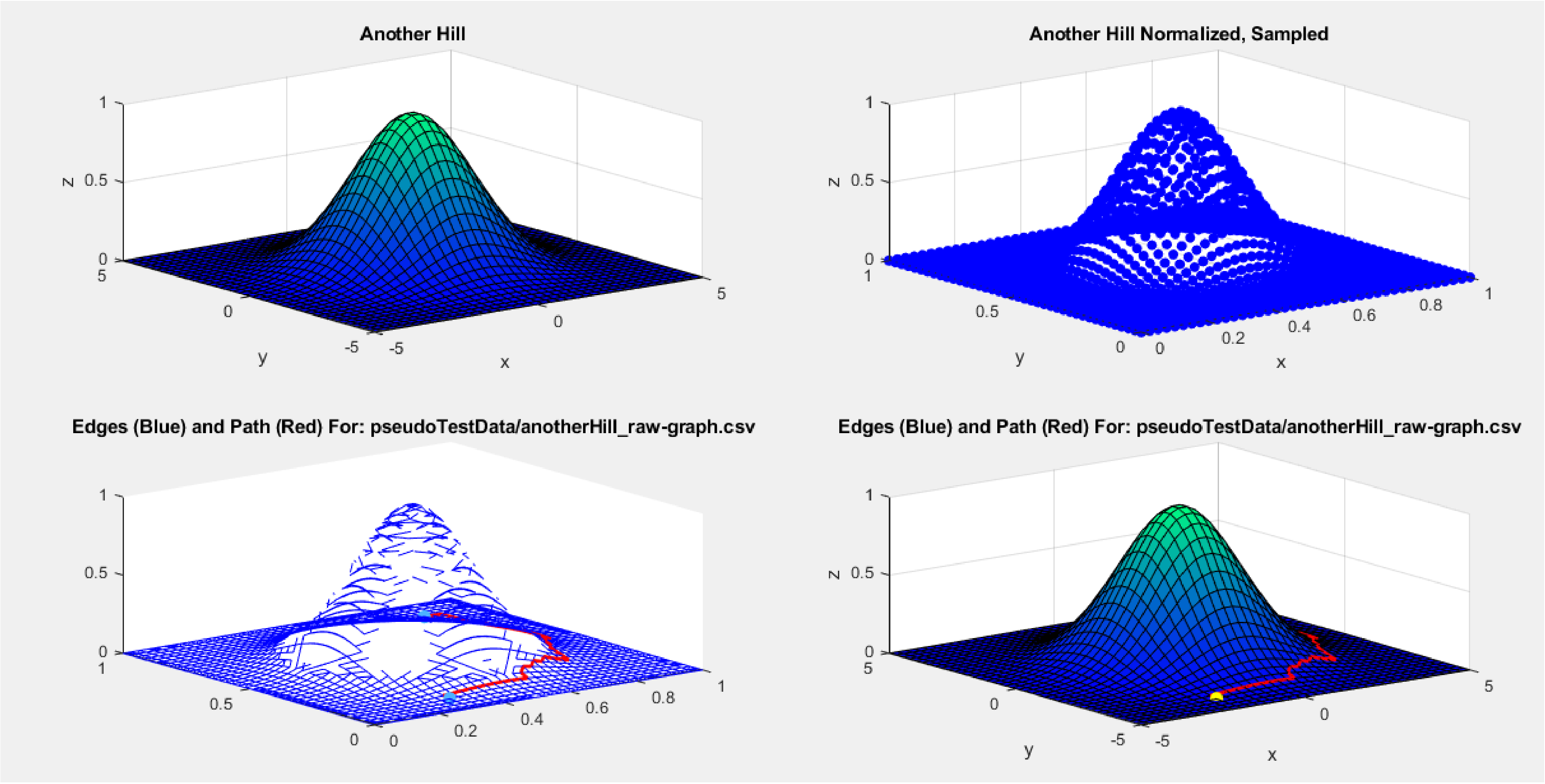
“Real” Test Cases

- Superimposed path output for samp11_gnd.las on satellite image to visually verify the result.
- Graph generation runs for 5-8 minutes on 38K datapoints
- Path computed in 4-5 seconds

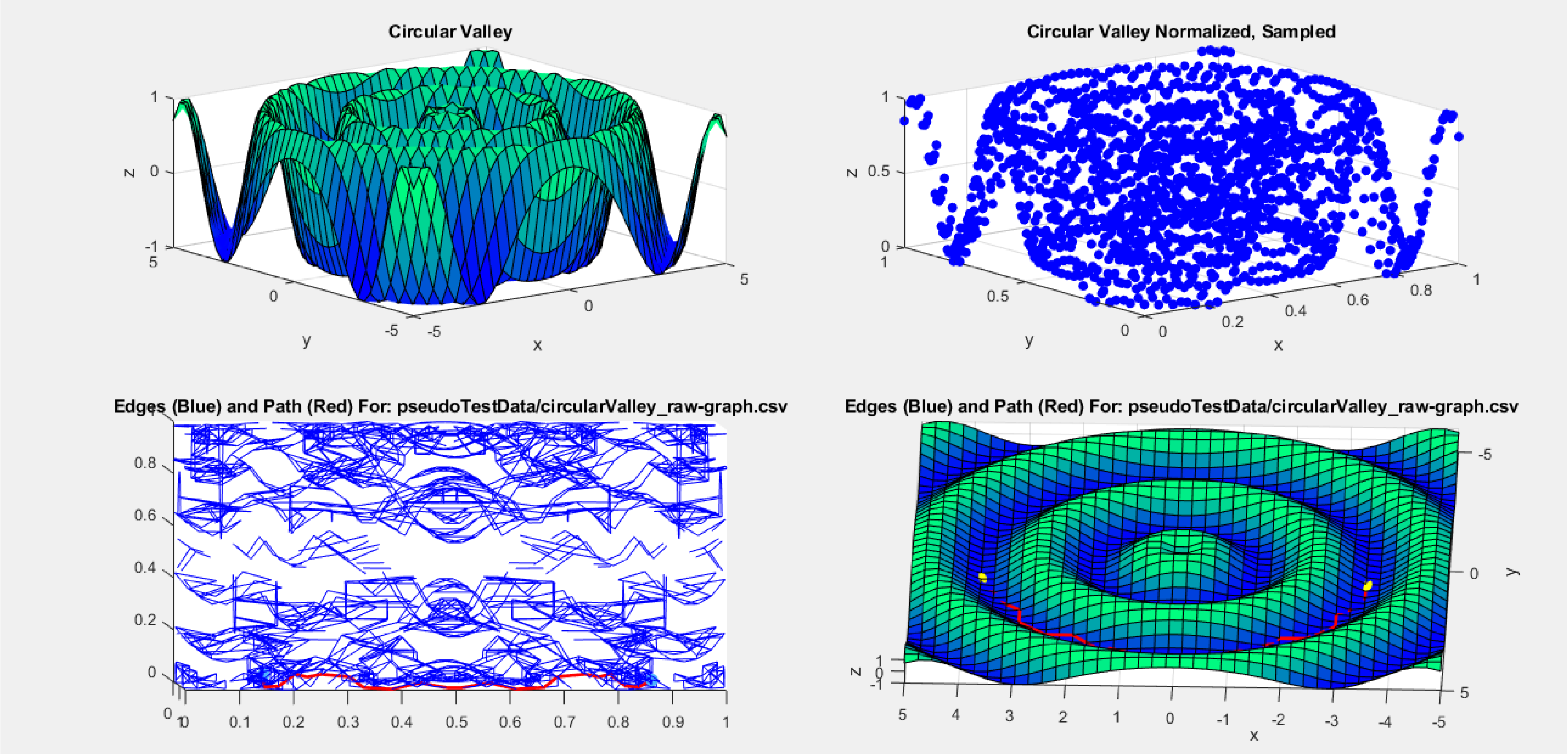
Results (Test Case)



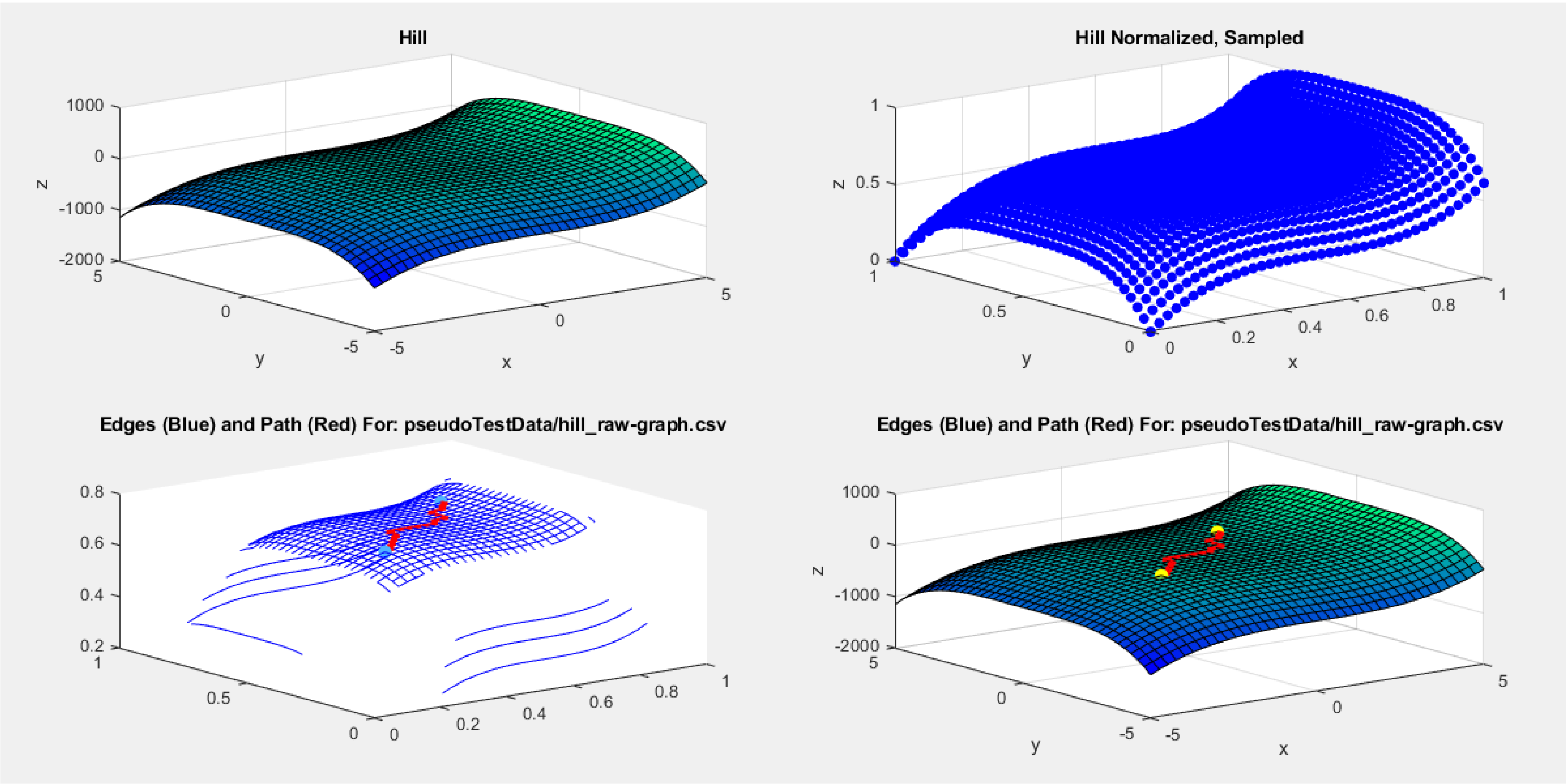
Results (Test Case)



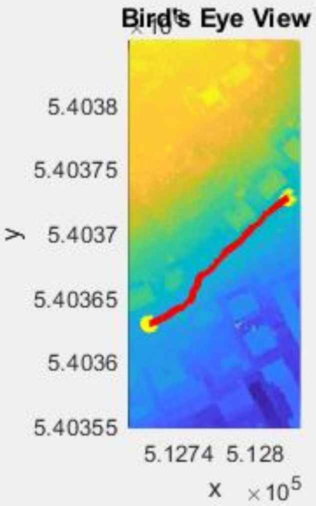
Results (Test Case)

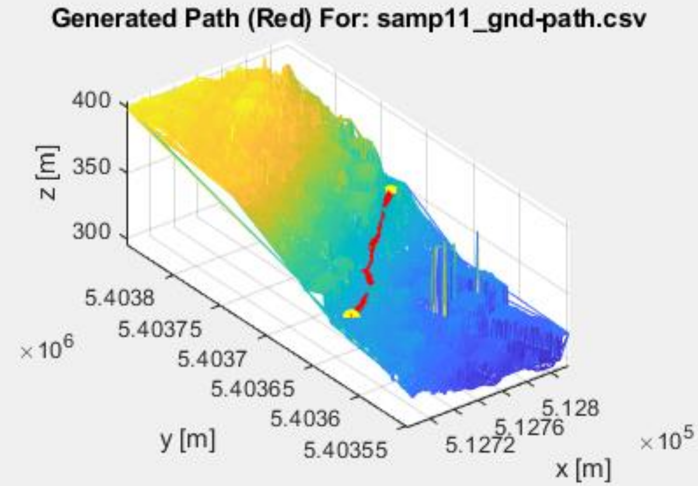


Results (Test Case)



Result (Real LiDAR Data)





Real-World Location: Stuttgart, Germany - Bird's Eye

Real-World Location: Stuttgart, Germany

Conclusions, Future Work

- The shortest navigable path through a given segment of terrain (described by an .las file) is successfully produced.
  - Accounted for tilt and terrain type limitations of the ground vehicle
  - Incorporated a safety distance

- Waypoints generated are with respect to the coordinates implicitly defined in the .las file.

- After graph generation, paths are computed on the order of seconds
  - 4-5 seconds for samp11_gnd.las - dataset of 38,000 points
  - Thus, solution can be applied to moving targets (stretch goal)

- Future work:
  - Account for robot size
  - Optimize graph generation code for performance
    - Multithread, evaluate tilt of edges in batches