Unreal Engine-Based Photorealistic Aerial Data Generation and Unit Testing of Artificial Intelligence Algorithms

Andrew Buck[a], Raub Camaioni[b], Brendan Alvey[a], Derek T. Anderson[a], James M. Keller[a], Robert H. Luke III[b], and Grant Scott[a]

[a]Department of Electrical Engineering and Computer Science, University of Missouri, Columbia MO, USA
[b]U.S. Army DEVCOM C5ISR Center, Fort Belvoir, VA, USA

SPIE Geospatial Informatics XII
April 6, 2022

Presented by Andrew Buck

Motivation

- **UAVs make great research platforms!**
  - Move freely in 3D space
  - Variety of sensing modalities (cameras, LiDAR, …)
  - Useful for developing autonomous behaviors (AI)



- **Many applications would benefit from UAV autonomy**
  - 3D scene reconstruction, surveillance, EH detection, S&R, …
  - Even some limited autonomy could help human operators

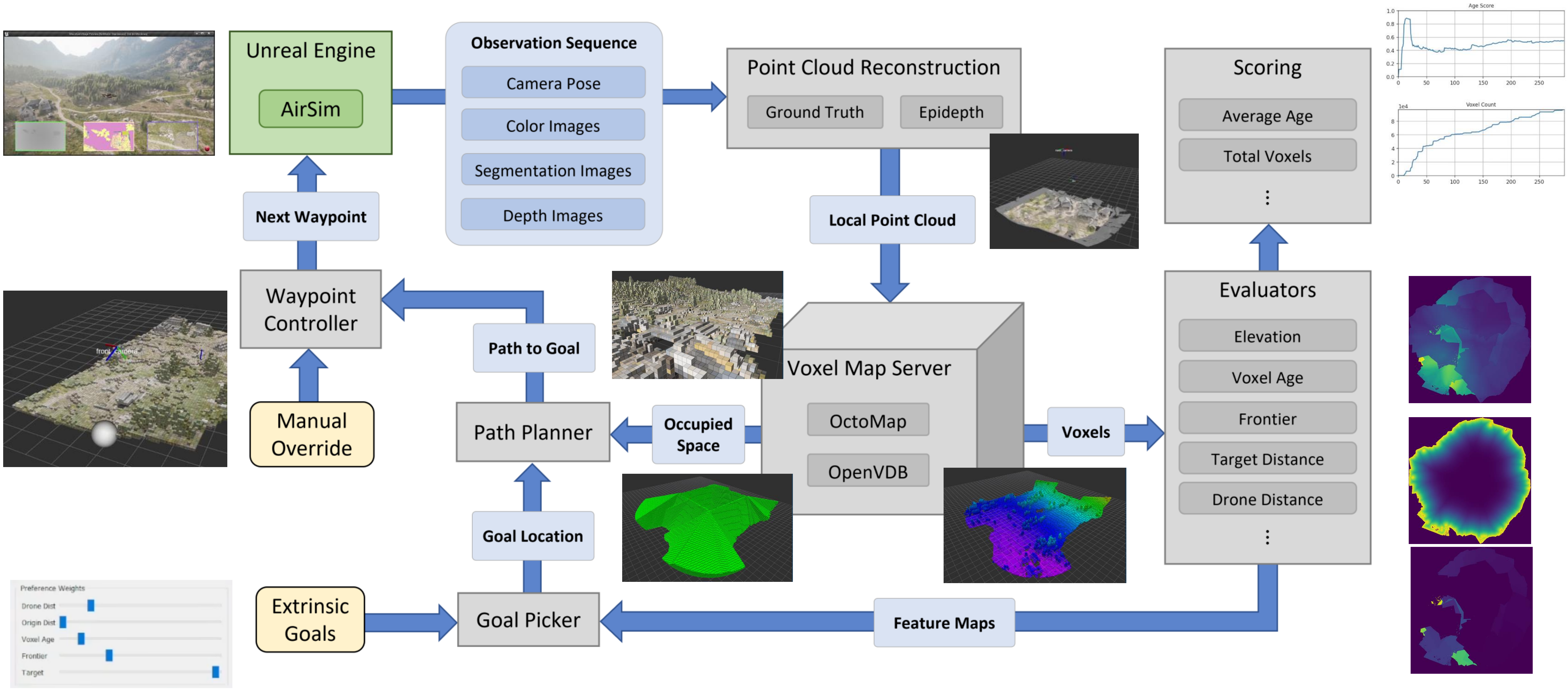
- **Our Goal:**
  - Develop a modular and extendable framework for studying UAV autonomy with photorealistic simulation tools

Why Simulation?

- **Real-world data collection can be challenging**
  - Costly and time-consuming
  - Risk losing hardware
  - No available ground truth (object detection, segmentation, depth, …)

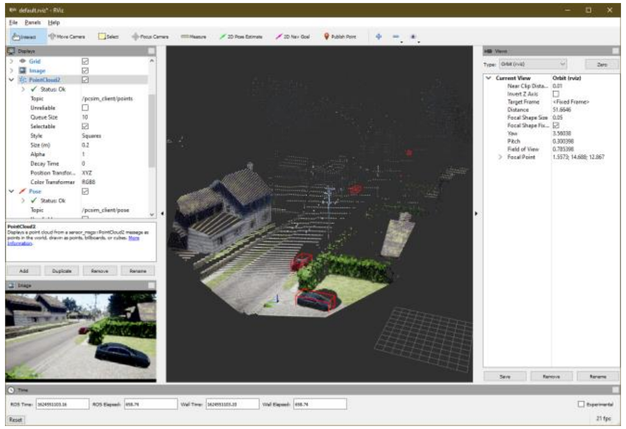
- **Simulation environments have matured**
  - Unreal Engine can produce photorealistic imagery (RGB, depth, …)
  - Large marketplace of maps and assets
  - AirSim and ROS provide a link to connect back to the real-world

Simulation Architecture

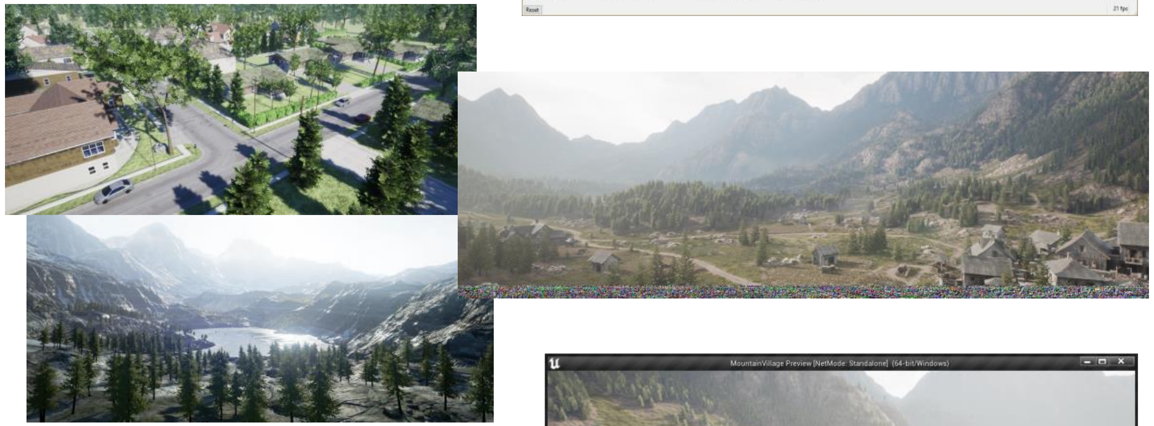


ROS, Unreal Engine, & AirSim

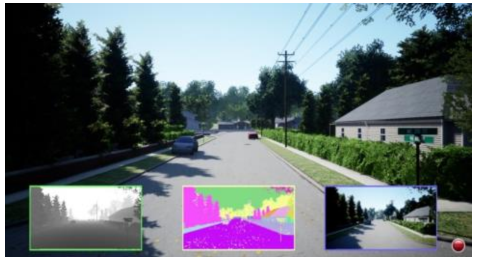
- **ROS (Robot Operating System)**
  - Provides a message passing framework
  - Distributed architecture
  - Several built-in message types and visualization tools (RViz)

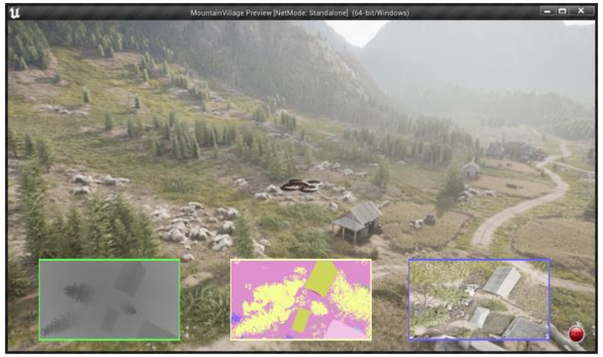


- **Unreal Engine (UE)**
  - Photorealistic rendering and simulation
  - Large marketplace with custom maps and assets



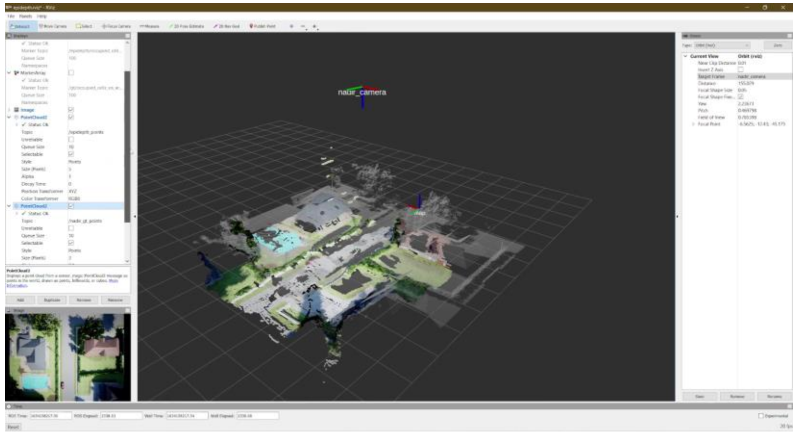
- **AirSim**
  - UE plug-in that simulates a virtual drone
  - Provides pose, depth, and image sensors
  - Responds to commands via ROS

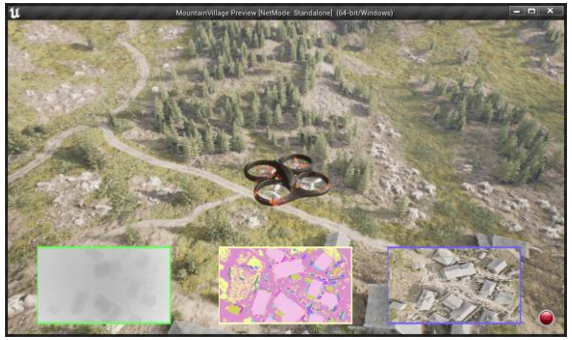


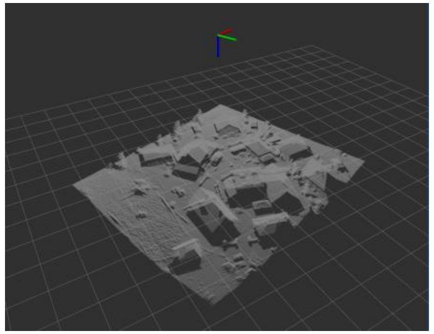


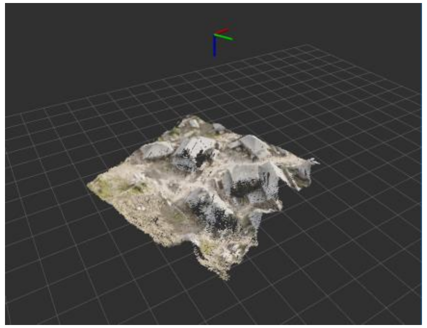
Point Cloud Reconstruction

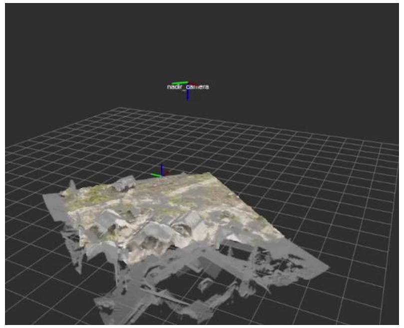
- **Local point clouds**
  - We build a 3D representation of the scene from the current drone location
  - Used to construct a map of the environment
  - Created from images, but could also use LiDAR
- **Ground truth point clouds**
  - Uses the known depth image from AirSim
  - Accurate reconstruction as ground truth
  - Less realistic (perfect pose and depth not usually available)
- **Estimated point clouds**
  - Epidepth algorithm uses image pairs and pose to predict depth
  - Can be applied on real-world drones
  - Use known ground truth to evaluate accuracy





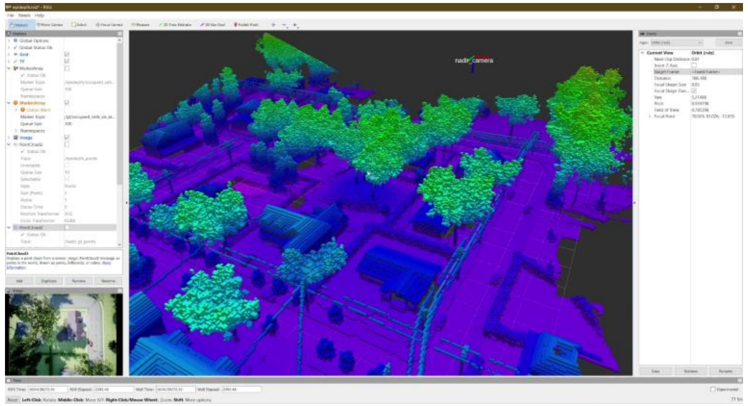


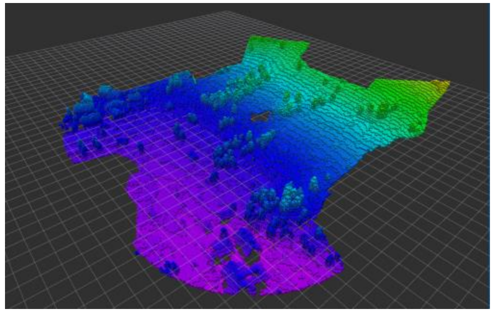


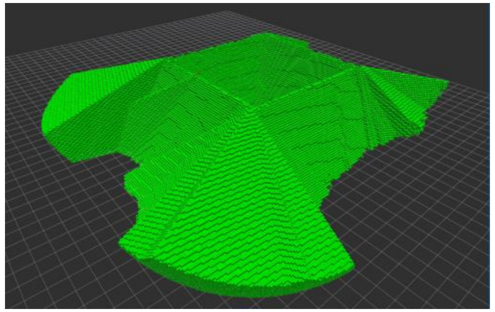


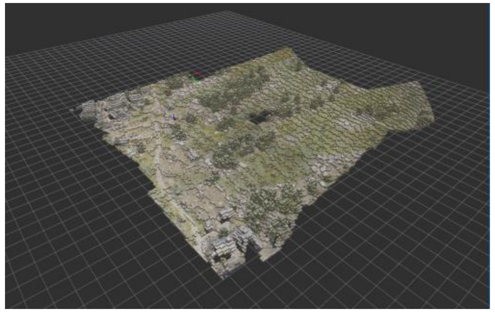
Map Server

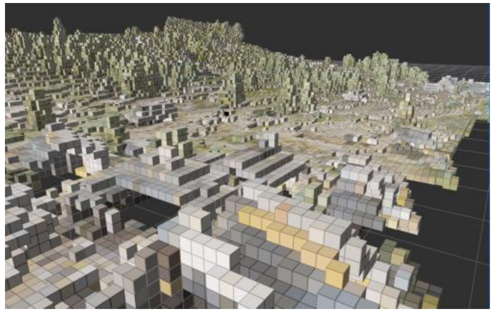
- **Voxel Grid Knowledgebase**
  - Point cloud observations are aggregated in a large world-space voxel grid
  - This map represents everything the agent has observed in the environment
  - Can be queried and processed to decide goals and evaluate performance
- **OctoMap**
  - Probabilistic occupancy grid based on octrees
  - Uses camera pose to determine free space
  - Commonly used with ROS
- **OpenVDB**
  - Library for managing large sparse voxel grids
  - No inherent modeling of free space
  - Efficient for applying global functions to all points
  - Commonly used for simulation and effects in the film industry









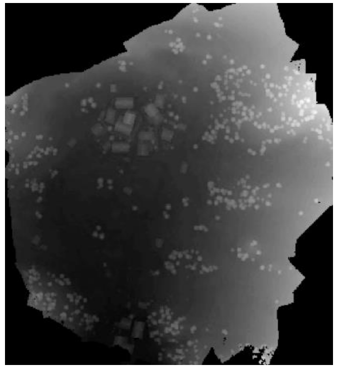


Evaluators

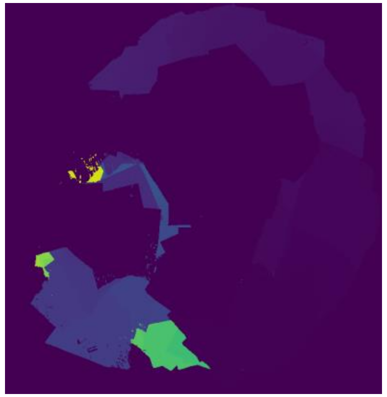
- **Process the voxel map**
  - Understand features of the space
  - Interpret to determine goal locations
  - Use to evaluate overall performance

- **Feature maps**
  - **Elevation:** Max voxel height at a location
  - **Age:** Time since the location was last observed
  - **Frontier:** How close to the edge of the explored space
  - **Target Distance:** Distance from some specified target
  - **Drone Distance:** Distance from the current drone position

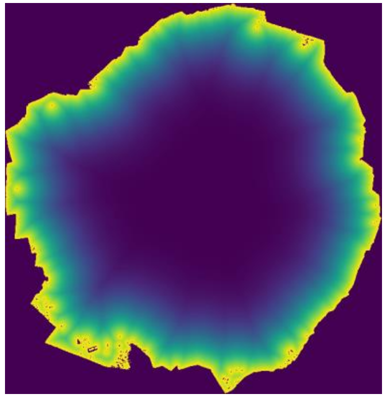
Voxel Map



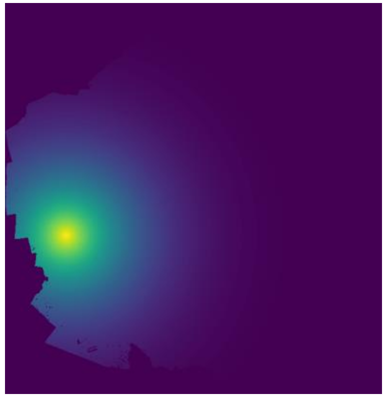
Elevation



Age



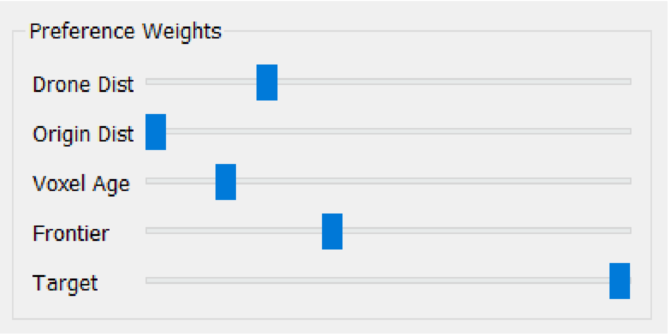
Frontier

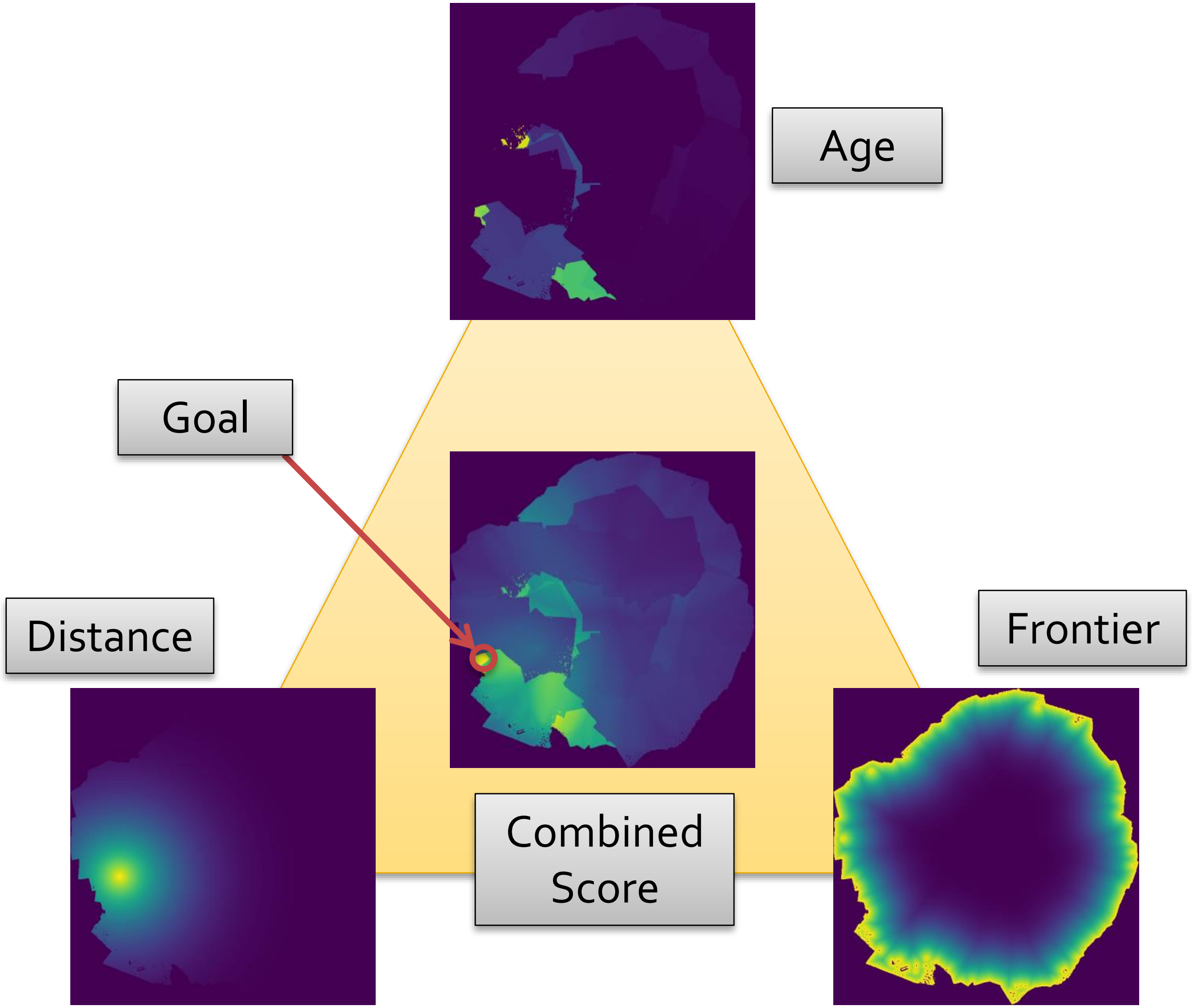


Distance

Multi-Criteria Goals

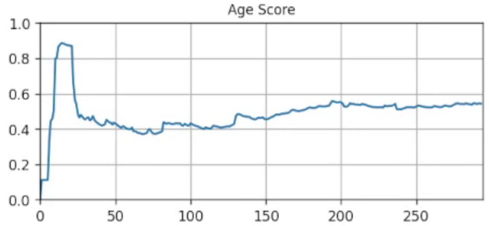
- **Goal selection**
  - Feature maps are aggregated with user-defined linear weights
  - For each feature map $k \in [1, \dots, N]$ containing pixels $p_{ijk} \in [0, 1]$, we define a weight $w_k \in [0, 1]$
  - The score of each location is given as $S(p_{ij}) = \sum_{k=1}^{N} w_k p_{ij}$
  - The location with the maximum score is chosen as the current goal

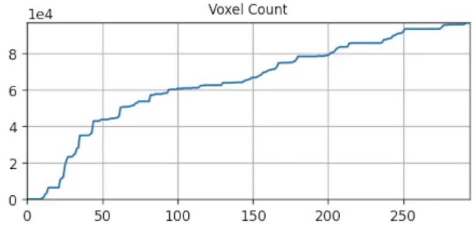


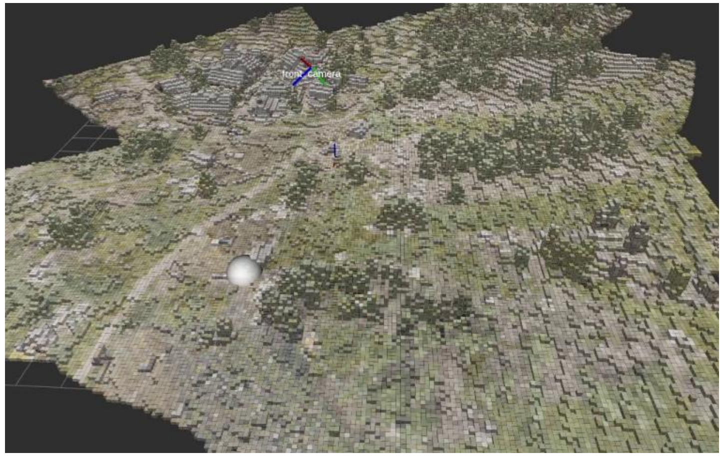


Moving and Scoring

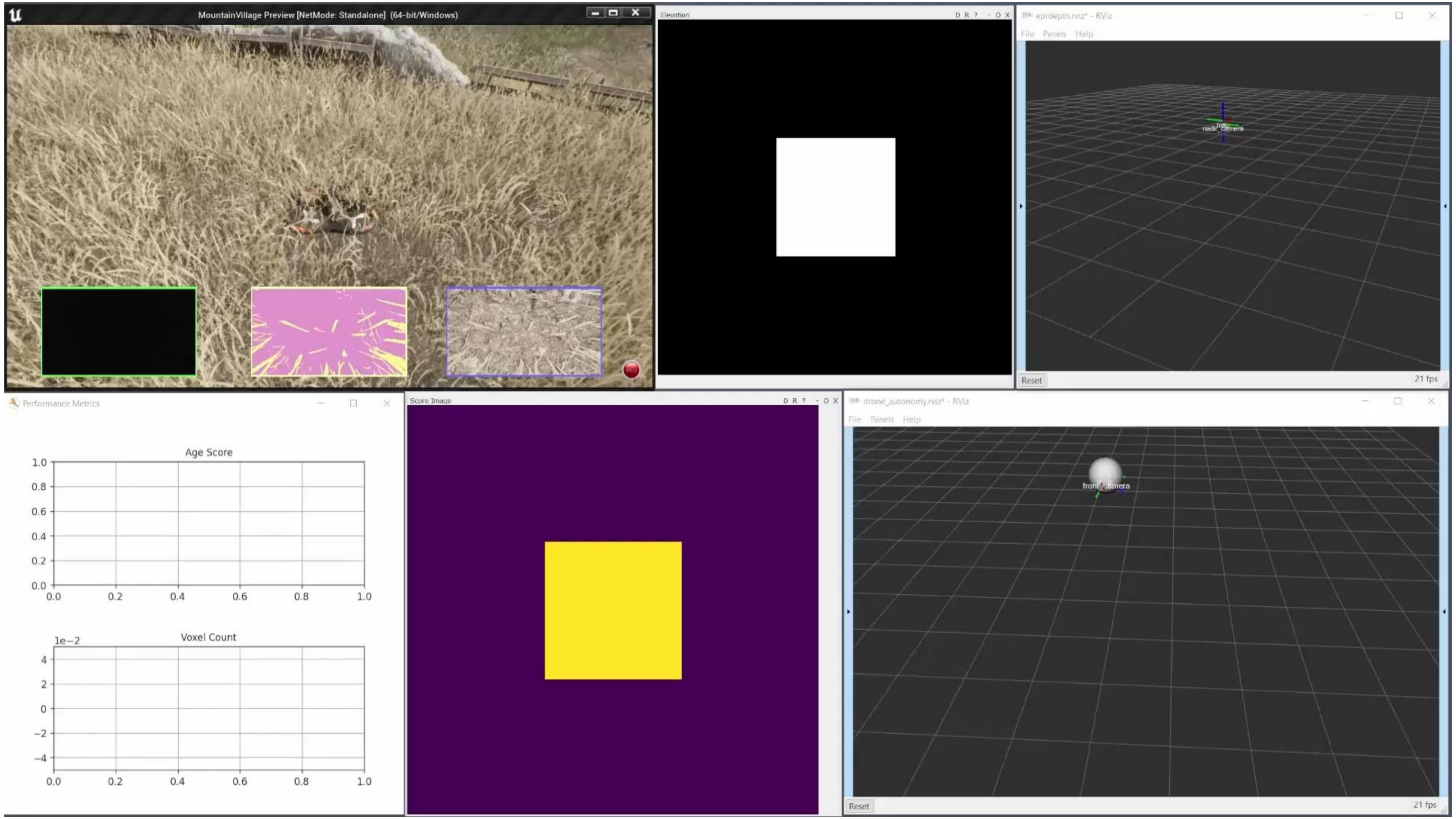
- **Path planning**
  - The current goal is set as the next waypoint
  - The voxel grid is used to determine an appropriate path to the goal
  - Can use elevation map to find min elevation or use known voxel free space
- **Waypoint control**
  - The ROS controller commands the drone to fly along the path to the next waypoint
  - An operator can override control at any time or begin a new mission phase
- **Scoring metrics**
  - Can track various features as the simulation progresses to evaluate performance
  - E.g., total voxel count, average voxel age, time to complete task, drone battery, elevation profile, number of collisions, …



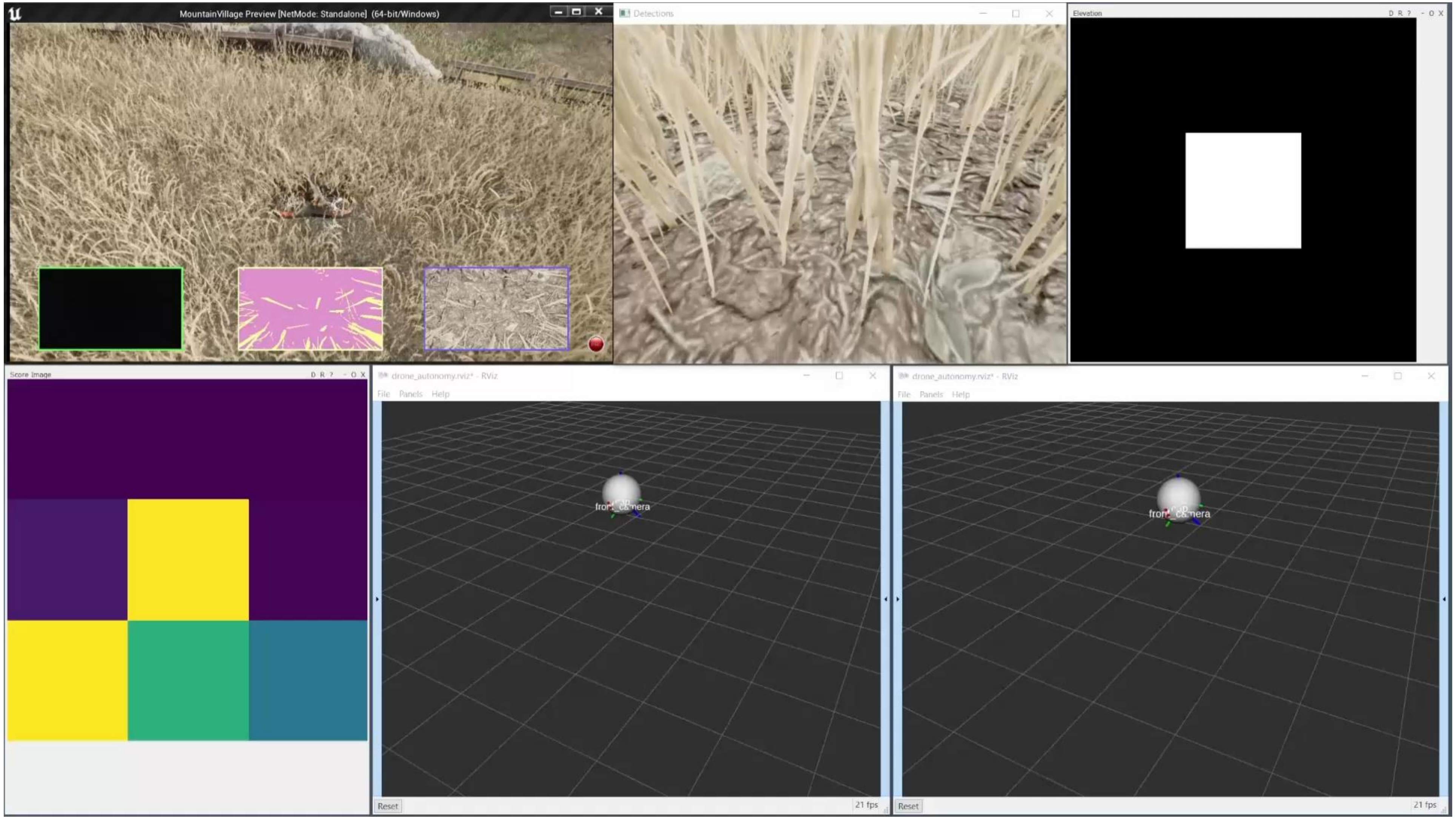




Loiter Scenario



Search and Rescue Scenario



Conclusions and Next Steps

- **Benefit of using simulated environments**
  - Useful for developing and testing AI algorithms for UAVs
  - Available ground truth allows for evaluation of performance
  - Modular design provides “unit testing” of individual components
- **Voxel map represents agent knowledge**
  - Many ways to encode information (observations, prior knowledge, …)
  - Provides a way to perform spatial reasoning (linguistic, semantic, …)
  - Interesting test bed for multi-criteria path planning
- **MCDM framework can be extended**
  - General strategy can be applied to many different problems
  - Adaptable to incorporate more human/robot teaming
  - Learn which features are important and develop optimal behaviors (RL)