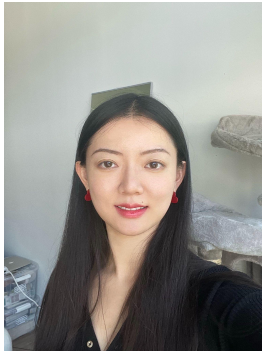
Multi-Cam Fusion

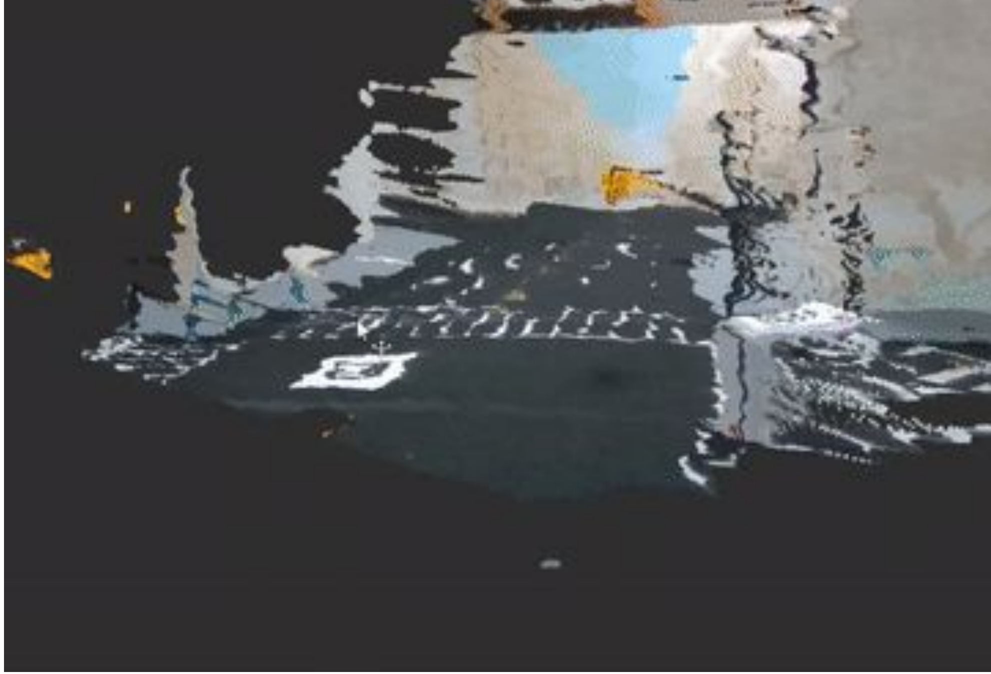
Group Members:
Arnuv Batra, Yue Wang, Peter Wilmot, Varun Kota, Jesse Lerner



Motivation for Multi-Cam Fusion

The point of this project is to be able to combine multiple camera views in a smart intersection.

<u>Why do we need multiple views?</u>
To see things that may not be visible from one perspective and avoid blindspots.



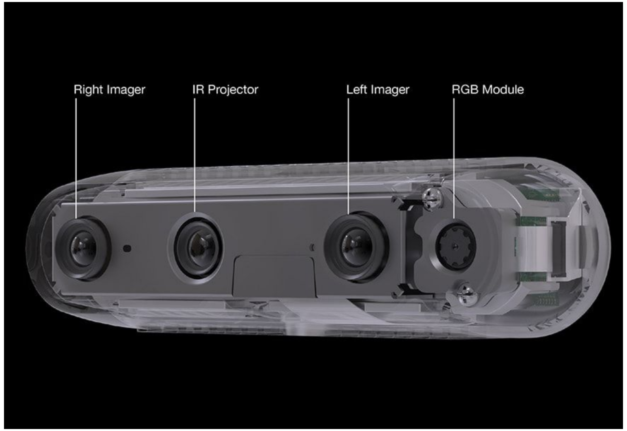
3D camera 1

Cam 1

Cam 2

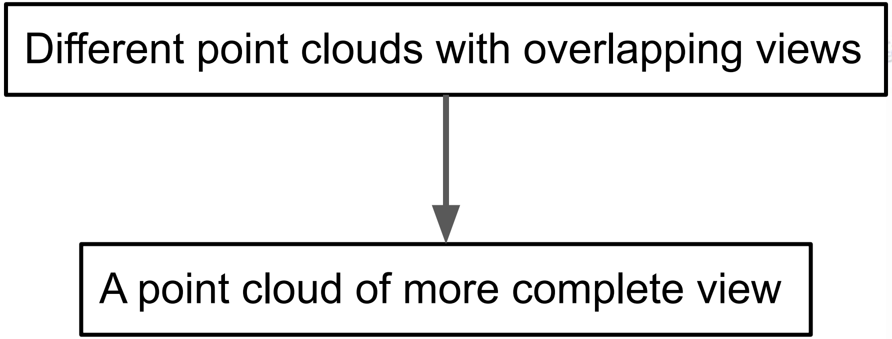
Project Objectives

- 4 Intel RealSense 3D Camera
    - Uses Point Cloud Data type
- Stitch projections to one model



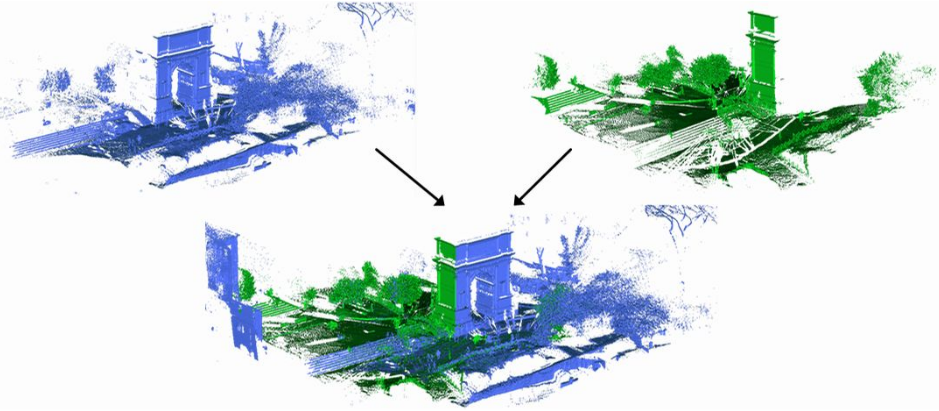
- Possible uses:
    - Intersection monitoring
    - Assisted driving
    - Traffic control
    - Crash prevention

High-Level Solution Design

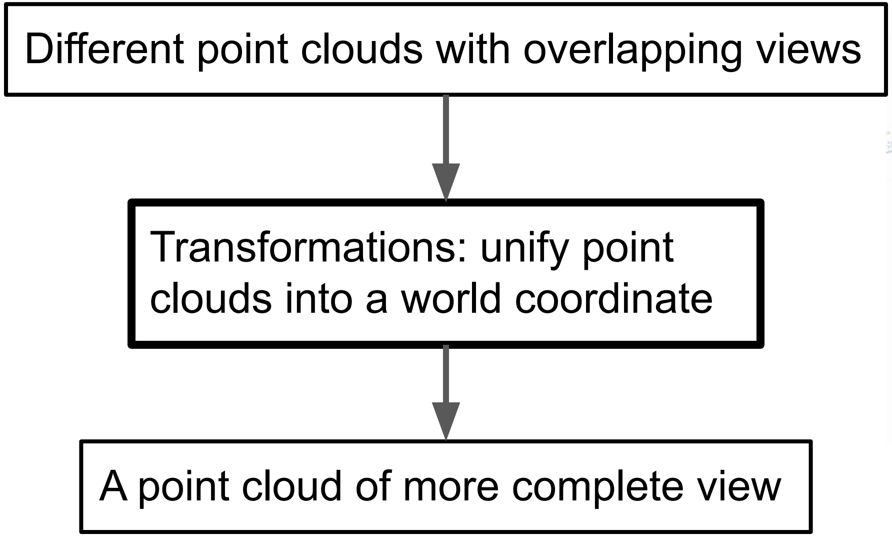


Different point clouds with overlapping views

A point cloud of more complete view



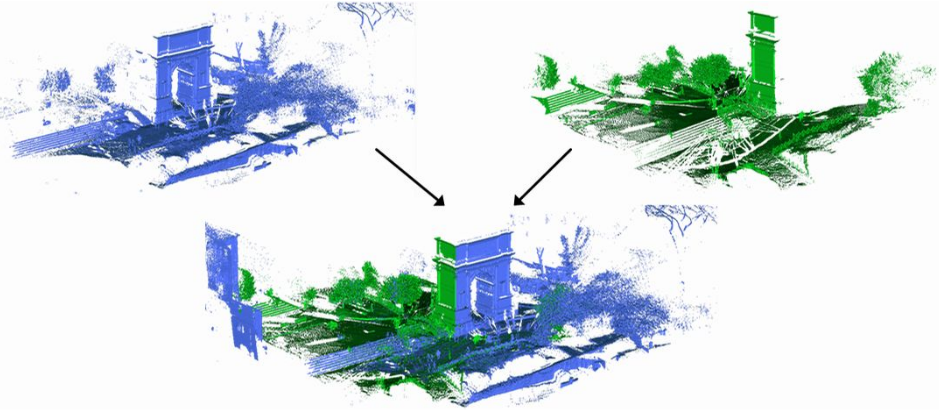
High-Level Solution Design



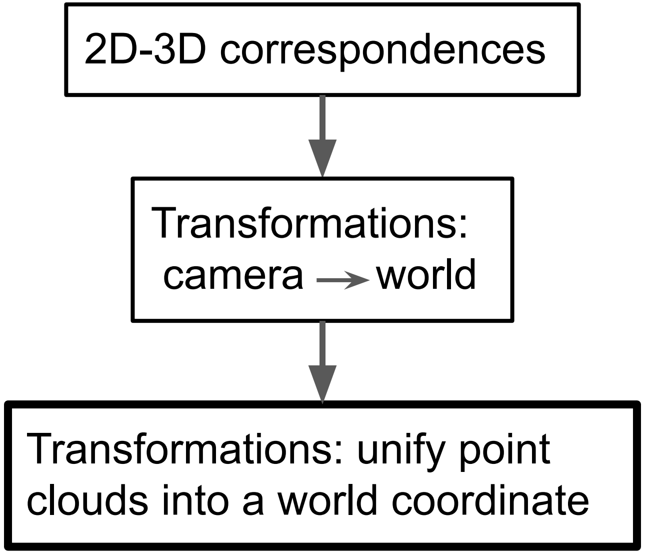
Different point clouds with overlapping views

Transformations: unify point clouds into a world coordinate

A point cloud of more complete view



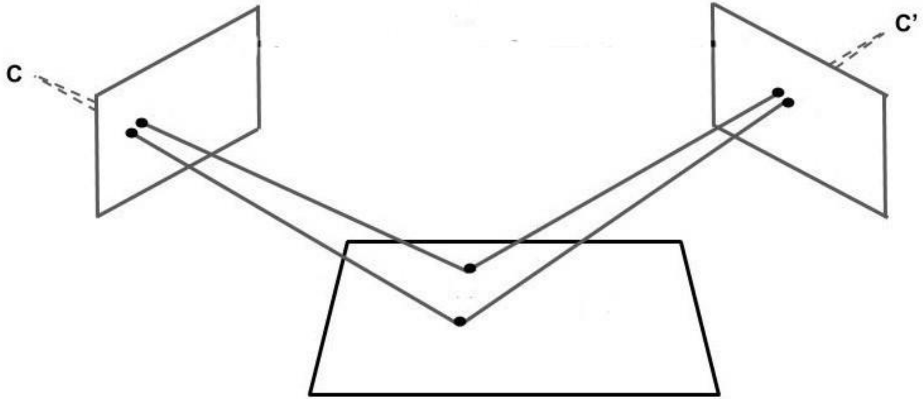
High-Level Solution Design



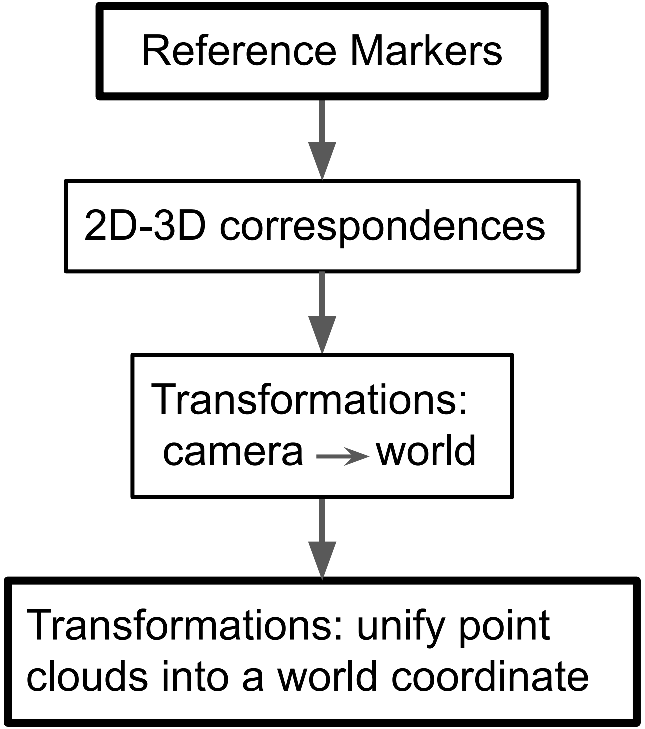
2D-3D correspondences

Transformations: camera → world

Transformations: unify point clouds into a world coordinate



High-Level Solution Design

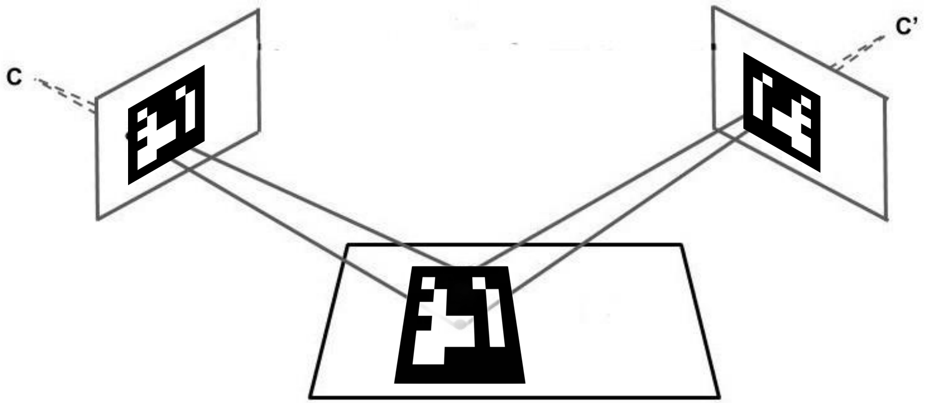


Reference Markers

2D-3D correspondences

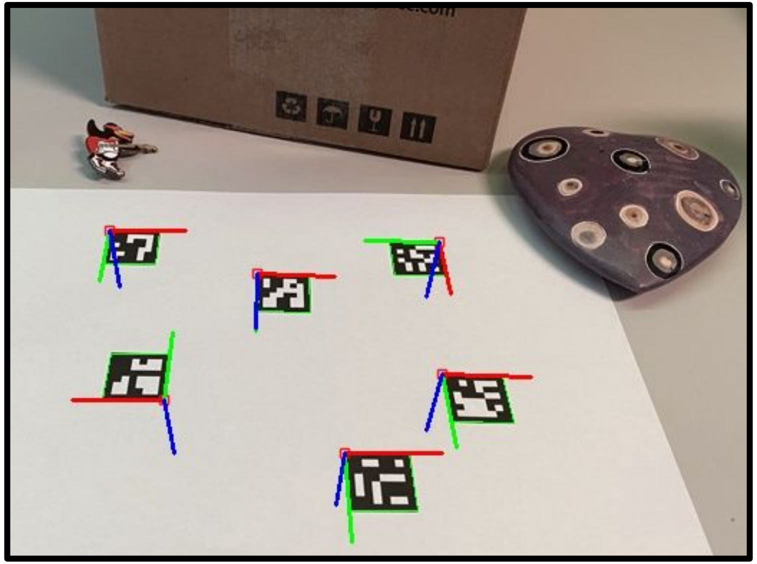
Transformations: camera → world

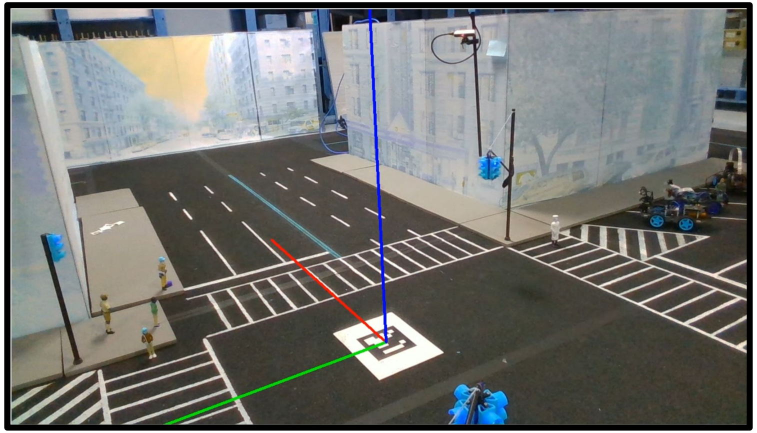
Transformations: unify point clouds into a world coordinate



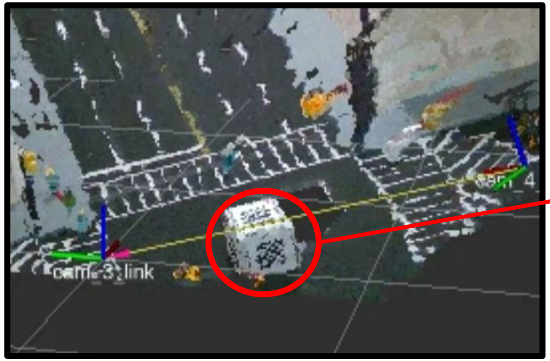
What is ArUco?

- ArUco Markers are known objects that are easy to find with computer vision
- The OpenCV library has a method to find ArUco markers and calculate relative camera position
- We use this to calculate the cameras relative positions

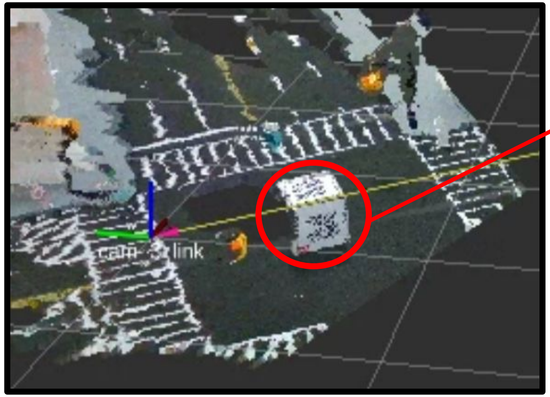




The Idea:

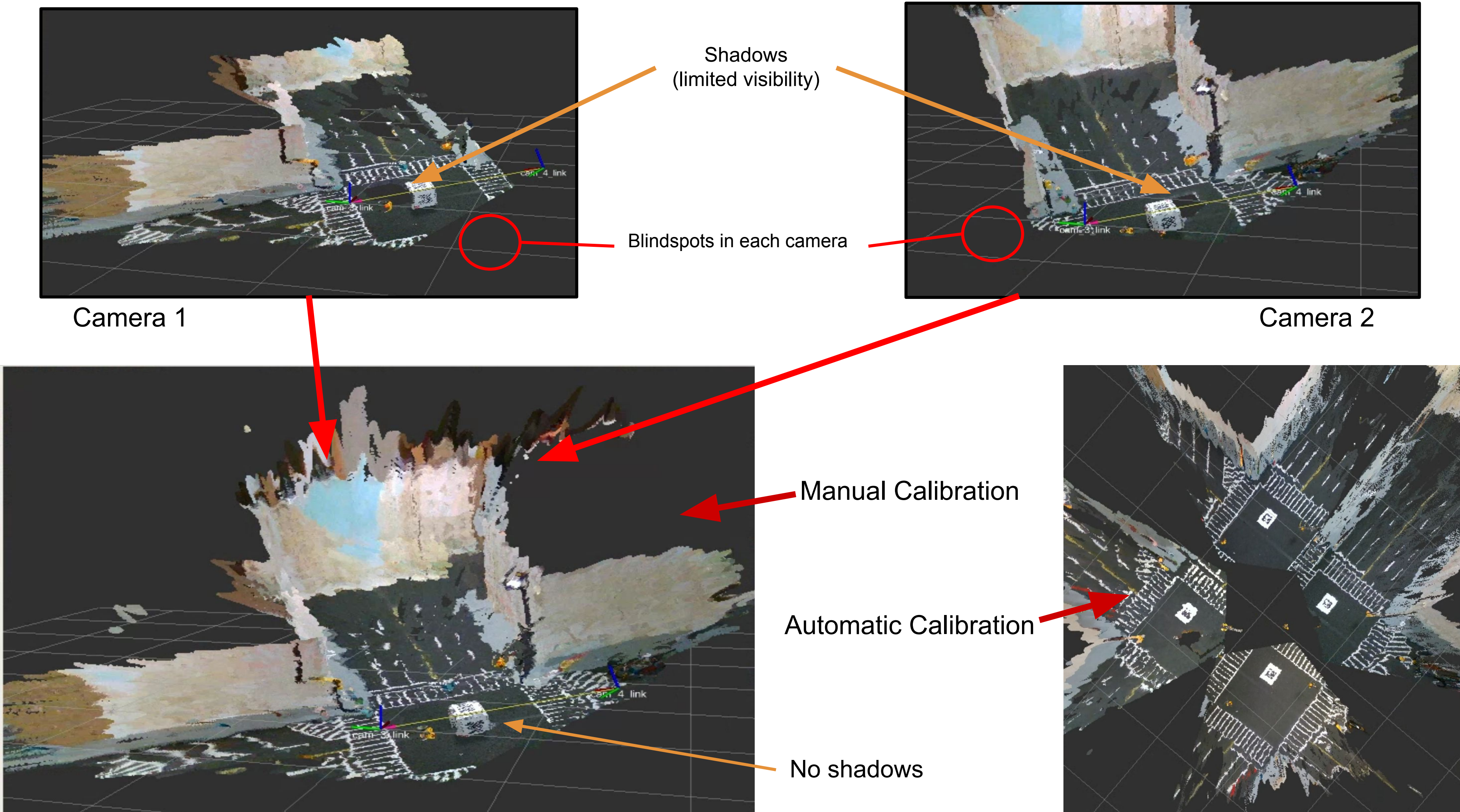


Camera 1 sees the ArUco tag and identifies its position relative to the ArUco tag's four corners



Camera 2 sees the same ArUco tag, finding its own position relative to the four corners of the same tag

Camera 1 and Camera 2 both know their positions in relation to a set of common points, so we should be able to estimate the transform between them.



Shadows (limited visibility)

Blindspots in each camera

Camera 1

Camera 2

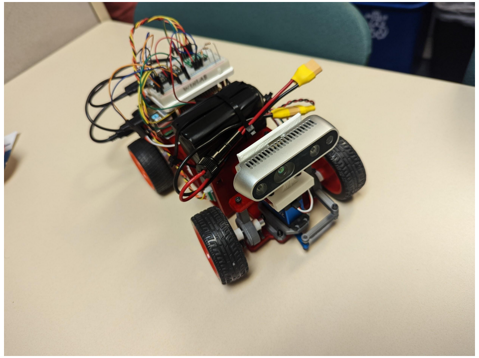
Manual Calibration

Automatic Calibration

No shadows

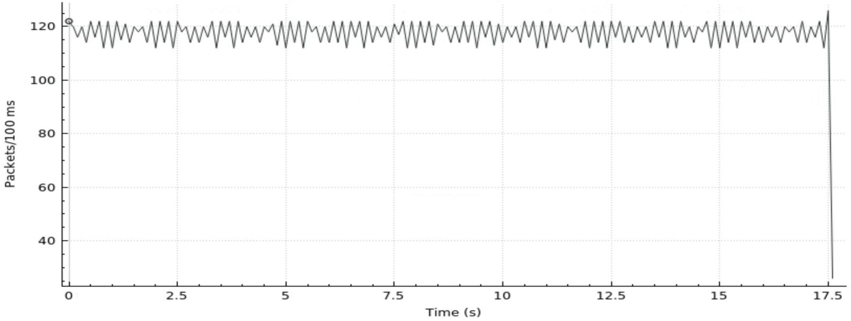
Future Work

- One big issue with using our work in a real-world setting is latency. Point Clouds are large and slow to process.
  - Figure out latency and whether it can be improved
  - Find faster ways to display or process information
- Fix automatic calibration
- Incorporate smart car camera and position
  - Challenges with moving camera and wireless connection
  - ArUco tags would not be able to help calibrate moving cameras, and wireless connections have much more latency



Hardware group car

Car driving at 60 mph
-> 27 meters per second
Example program latency: 0.5s
-> 0.5s x 27 m/s
Uncertainty: 13.5 meters



Any Questions?