Smart Intersection Cameras

Heneil Patel, Eleonore Pichon, Peter Wilmot

This work was supported in part by the NSF REU program and the donation from nVERSES CAPITAL
Project Inspiration/Idea

- Create a 3D model of an intersection
- 3D cameras on each corner
  - More coverage
  - 2D can only see front of object
- Why?
  - Detailed traffic/pedestrian info
  - Send details to smart cars
    - Allows “sight” around a corner/blockage
    - Allows for advanced tracking/prediction
Project Methods/Equipment

- Stitch multiple point clouds together
  - Cosmos scale intersection
  - Intel RealSense D415 Depth Cameras
Aruco Markers

- OpenCV
- 3D point projection
- Kabsch Algorithm
ROS/Rviz

- Allowed for easy streaming
  - pre-configurations for realsense camera
- Many pointcloud viewer options
- Inconsistent transformations
- Time/Resource consuming to install

90 degree rotation
Point cloud data transfer

- **Ethersense**
  - low resolution
  - hard to decode

- **OpenCV / Pickle files / Sockets**
  - smaller file size
  - easy to decode
Improving Marker Detection

- Different markers detected each frame
- Our solution:
  - Cache markers from last frame
  - Challenges with data types
    - Multidimensional arrays
    - Tuples
    - Lists
  - Up to 20 marker increase
More ArUco markers

- Almost perfect calibration
- Not practical for real life
  - Need realistic reference points
Sending multiple point cloud frames

- Loop:
  - Clients send files
  - Server downloads files
  - Viewer reads files

- Issues
  - Error when read and download same file at same time
  - Stops after few seconds of loading local and streamed files
Implementation after calibration

- Use PyTorch to analyze/modify images
  - YOLOv8
    - Deep Learning
    - Object Recognition
YOLOv8 Car Detection

- Train custom model for DIY cars
- 19 manually labeled images
- Many false positives
- Deployed on ultralytics hub mobile app
- Slower than regular YOLO (memory leak?)
Image Segmentation

- Coordinates of segmented “mask” in 2D
- Able to draw segmentation mask independently
- Allows for 3D object mapping
Future Improvements

- Fix point cloud video viewer
  - Lock/Unlock files for receiving/reading
  - Send frames directly
  - Implement queue/buffer
- Translate masks to 3D
- Combine masks into 3D mesh or bounding box