Dynamic Obstacle Avoidance and Path Planning

Nathan Brown, Martin Deegan, Gregory Meyer, Zhihao Ruan

Mapping

- We calibrate the position and orientation of three depth cameras pointing at the arm using AprilTags
- Using the calibration data, we combine depth information from our cameras to generate a point cloud of obstacles around the arm
- We remove distant obstacles and the arm from the point cloud, then reduce its density before using it for motion planning

Motion Planning

- We implemented a GPU-accelerated stochastic trajectory optimizer that minimizes collisions and acceleration of the resulting trajectory
- Each trajectory consists of n waypoints (joint angles) separated by equal timesteps.

$$p_1, \ldots, p_n$$

Velocity and acceleration can be computed from positions

- We minimize

$$J(p_1, \ldots, p_n) = \sum_{i=1}^{n} \text{collides}(p_i) + \frac{1}{2} \|a_i\|_2^2$$

Over the points

$$p_2, \ldots, p_{n-1}$$

We keep the first and last waypoints fixed to the start and goal positions.

The collides function penalizes collisions in the map.

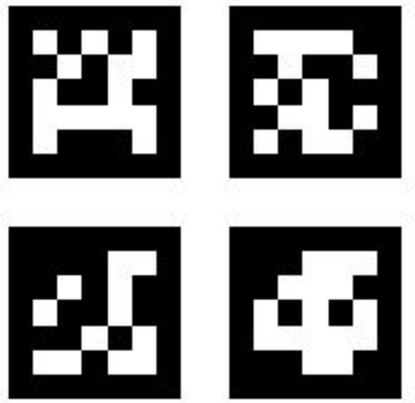
Control

- The Rexarm is configured with six degrees of freedom, including a gripper and wrist rotation
- Servo target positions and velocities are set by the motion planner and executed at 100Hz

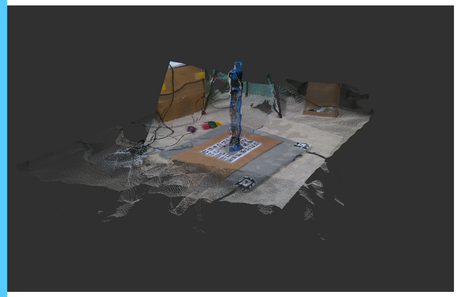
We maintain a **3D map** around a robotic arm, then use a **GPU-accelerated** trajectory planner to enable **safe operation** in **dynamic environments** in real time.

References

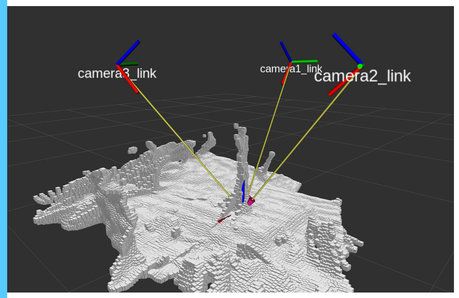
[1] C. Park, “Real-time Optimization-based Planning in Dynamic Environments using GPUs,” in International Conference on Robotics and Automation, 2013, pp. 4090



We use AprilTags to localize the cameras.



Combined depth clouds after camera calibration.



Point cloud from 3 cameras. Calibrated poses are shown as coordinate frames.