UR5 Obstacle-Avoidance Pick-and-Place with Arm Assistance
Weekly Progress Report 1
Yucheng Kang, Zhiyi Ren
April 12, 2018

- This Week's Goals
  - Implement UR5 control with MoveIt!
  - Implement arm tracking with Kinect
  - Write the node that converts arm poses to UR5 poses
- This Week's Progress
  - UR5 control with MoveIt!
    - We followed moveit tutorials to implement UR5 kinematics and planning. It works for basic planning on the real robot.
    - As we start looking into implementing customized planning, we realized that we need to use more functions in the API. So we start to write new codes for future integration.
  - Skeleton tracking with Kinect
    - We used openni_tracker (https://github.com/ros-drivers/openni_tracker) package for skeleton tracking and it works perfectly on our laptops (see Fig. 1).
    - However, on the desktop in Wyman it always crashes in a few seconds. The reason seems to be the unstable usb connection. If we type lsusb command, we can see that Kinect intermittently disappears in the device list.
  - Converting poses
    - The node kinect_teleop is done (see Fig. 1).

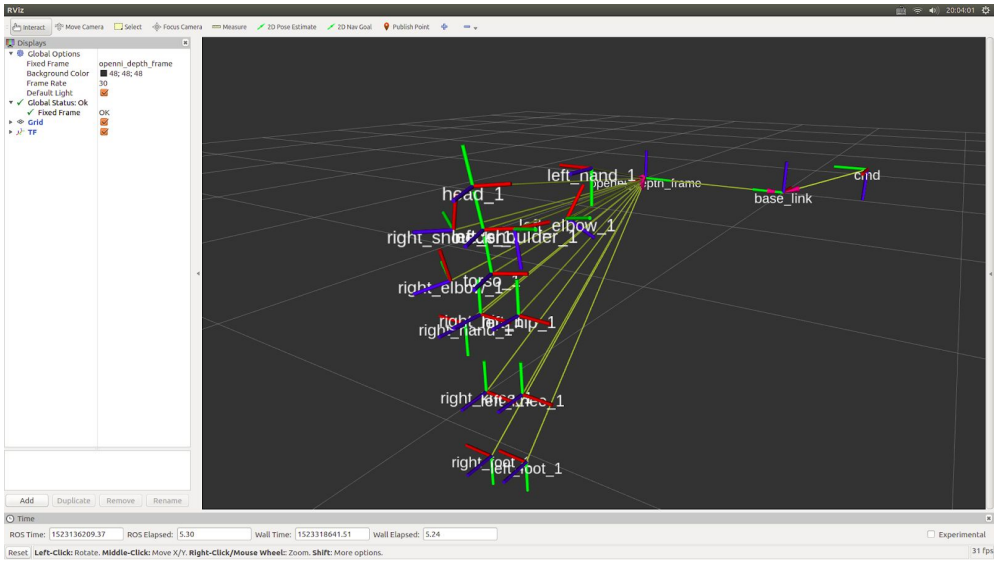

Fig. 1 Testing kinect_teleop in RViz

- Camera mount
  - We made a 3D-printed camera mount that fixes the R200 and the gripper to the UR5 (see Fig. 2). This was in the 2nd week's goals.

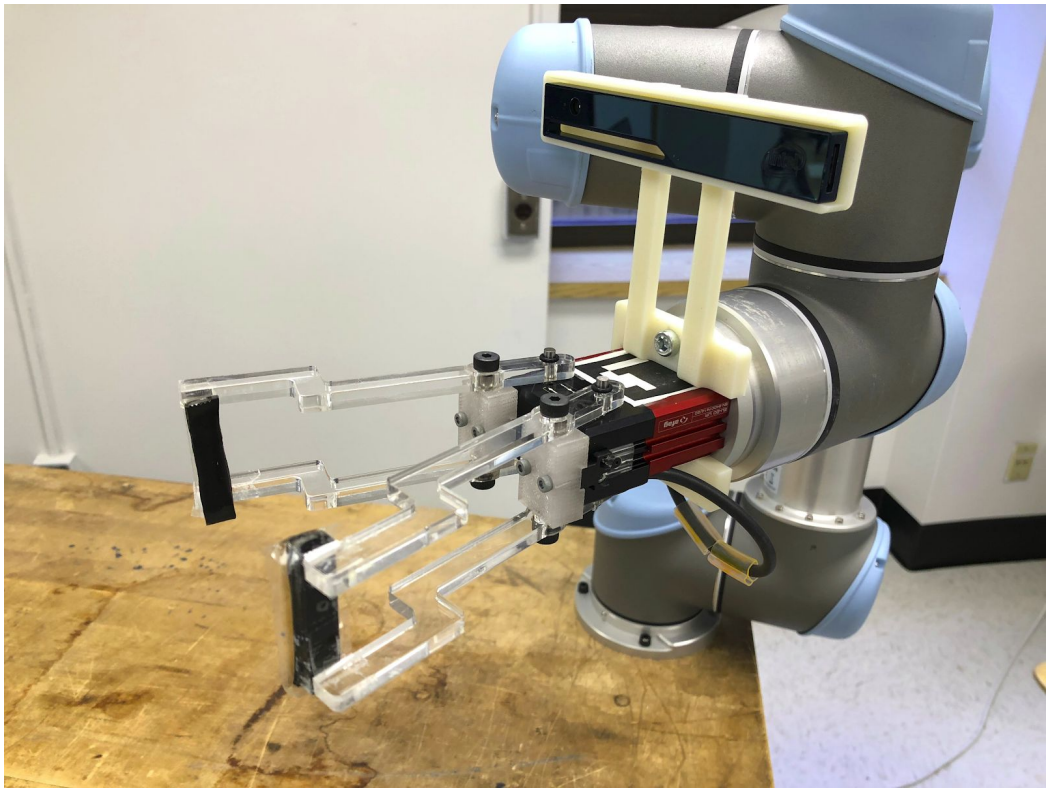

Fig. 2 Camera mount, R200, and gripper at the end-effector

- System overview
  - We spent some time refining the overall design (Fig. 3).

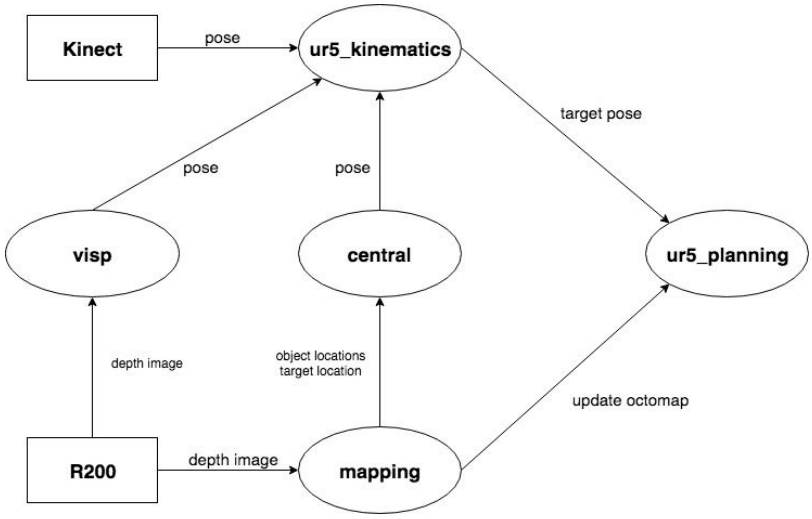

Fig. 3 Preliminary overall design

- We also need to implement hand-eye calibration for R200 cameras.

- Changes in Project Scope/Goals
    - No changes.
- Lessons learned
    - OpenNi package works with Kinect 1414 (the original version of Kinect 1), but not 1473.
    - Using rosbag can greatly simplify debugging process especially when you have to perform silly poses in front of Kinect every time.
- Next week's goals
    - Implement AR tag tracking and calibration with R200.
    - Implement visual servoing with R200.