Vision Subsystem Description

Base System

(Excludes Features for "Unknown" Competition Objects)

The proposed base configuration for the vision system will utilize 4 sensors. Three sensors will be mounted above three of the system bins, while the fourth will be eye-in-hand on the robotic arm. Each sensor will be controlled by a separate CPU as outlined in Figure 1.

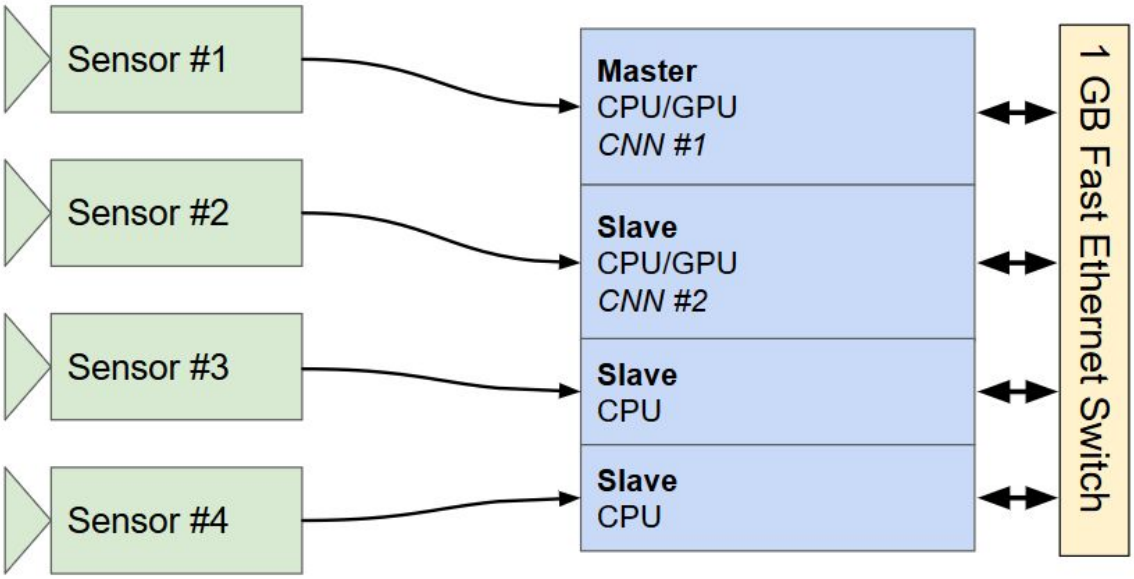

Figure 1: Vision System Hardware Diagram

Each CPU will be connected to one another via a switch. One CPU will act as the Master while the others act as Slaves, with the Master CPU dictating communication within between the CPUs via the state machine. The Master CPU and one Slave CPU will include GPUs in order to support the ability to run two separate CNNs simultaneously. No streaming data from the sensors will be passed between CPUs in order to reduce communication time and complexity, and instead each CPU will process the relevant sensor feed into a set of usable matrices. Those matrices will include raw RGB data and correlating segmented depth images. The segmented depth images will be created through the use of sensor calibration data and PCL (PassThrough Filter). The raw RGB data will be passed to relevant CNNs for processing (in the base configuration this will be Faster-RCNN). The output of the RGB CNN will be an object classification matrix which will correlate to the segmented depth images. Both of these matrices will then be fed as inputs to PERCH, along with the target object to be picked/stowed and a library of the object models. The input/output diagram for PERCH can be seen in Figure 2.

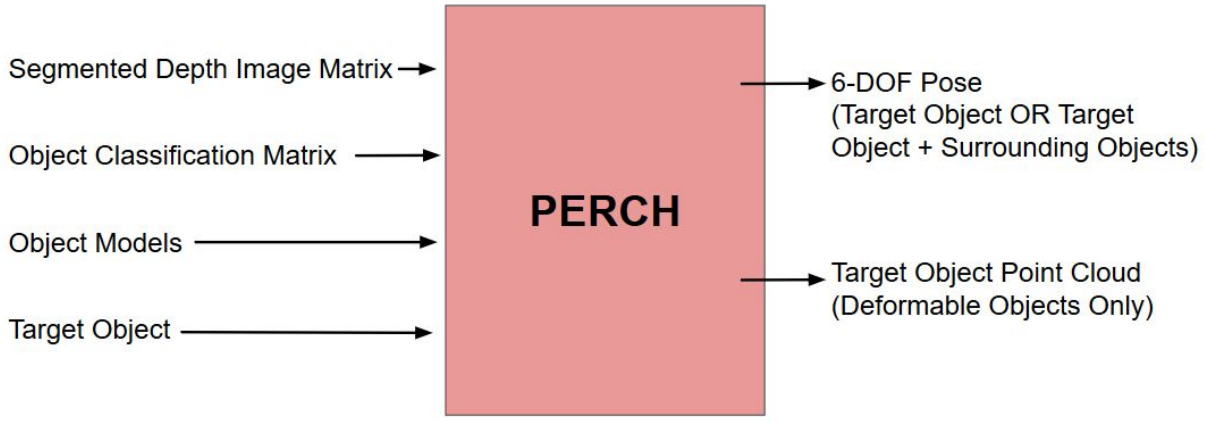

Figure 2: PERCH Input/Output Diagram

PERCH will use the inputs in order to output 6-DOF poses for either the target object, or the target object and surrounding objects based on its configuration. These 6-DOF poses will then be passed to Grasping. In the case of a target object which is deformable PERCH will simply pass the point cloud of the object to the next stage in the pipeline (again PERCH may also pass the 6-DOF of any rigid objects surrounding the deformable item if it is so configured). In the event of a target object which is deformable the point cloud will then be passed to a PCL function which will attempt to identify surface normals on the object, and the points of those surface normals will then be passed to Grasping. The overall software diagram can be seen in Figure 3.

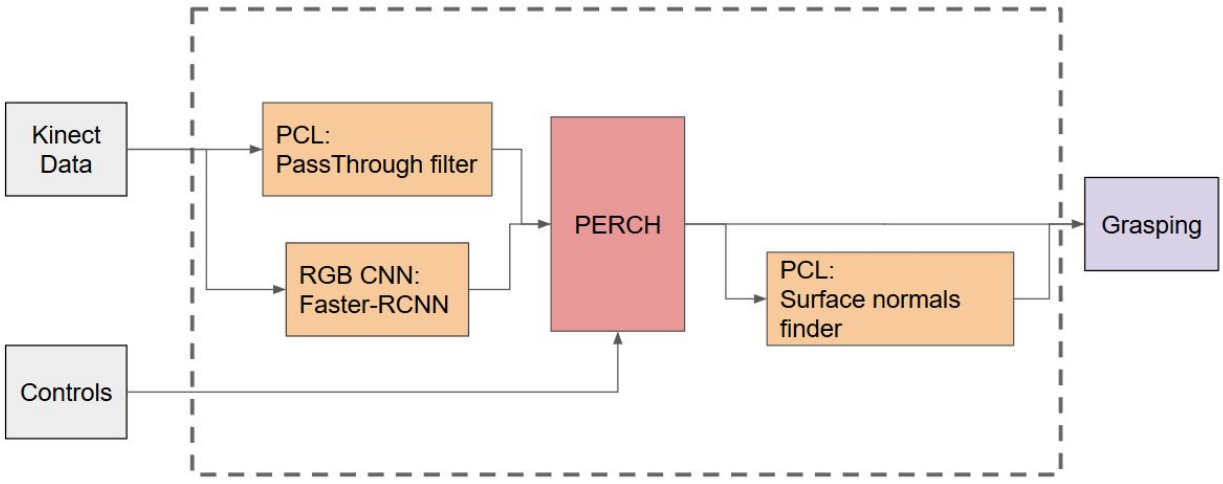

Figure 3. Vision Software Diagram

Potential System Improvements

Future RGB CNNs may be run in parallel with Faster-RCNN to see if there are any performance gains of one over the other. Sensor fusion may also be employed in order to acquire higher definition point clouds as input data.

Unknown Items

Approaches for unknown items are still being considered. A point cloud CNN may be employed for unknown items, and would exist within the system diagram between the PCL PassThrough Filter and PERCH. In the event that a point cloud CNN is employed the second GPU will be utilized for it once the appropriate RGB CNN choice has been finalized (once an RGB CNN choice has been made there is no longer a need to use two GPUs for CNN comparisons).

Localization

April tags will be employed for bin localization. A strip of tags will be placed on the back/top of the shelves for each static Kinect, while another strip of tags will be on the bottom/front of each shelf to localize the eye-in-hand Kinect. The april tag placements can be seen as the checkerboard pattern in Figure 4.

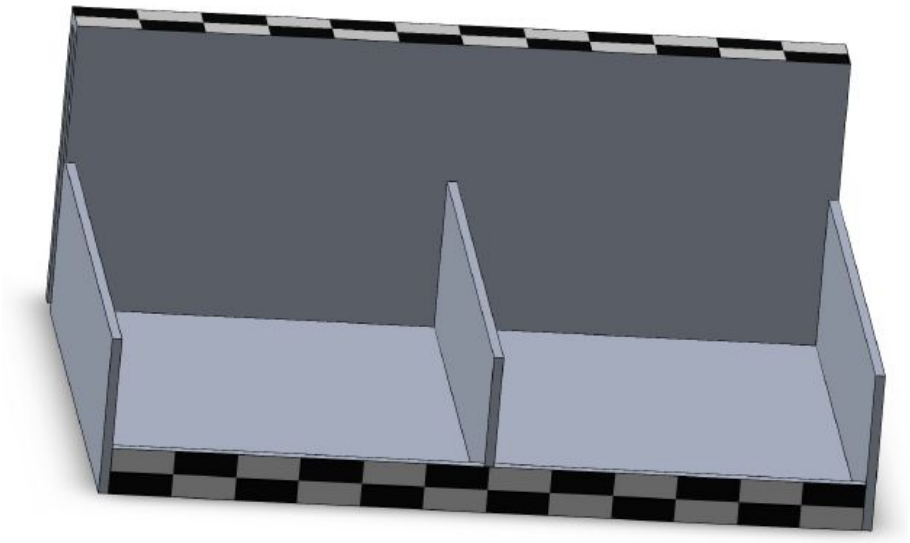

Figure 4. Bin april tag placements

ROS Architecture

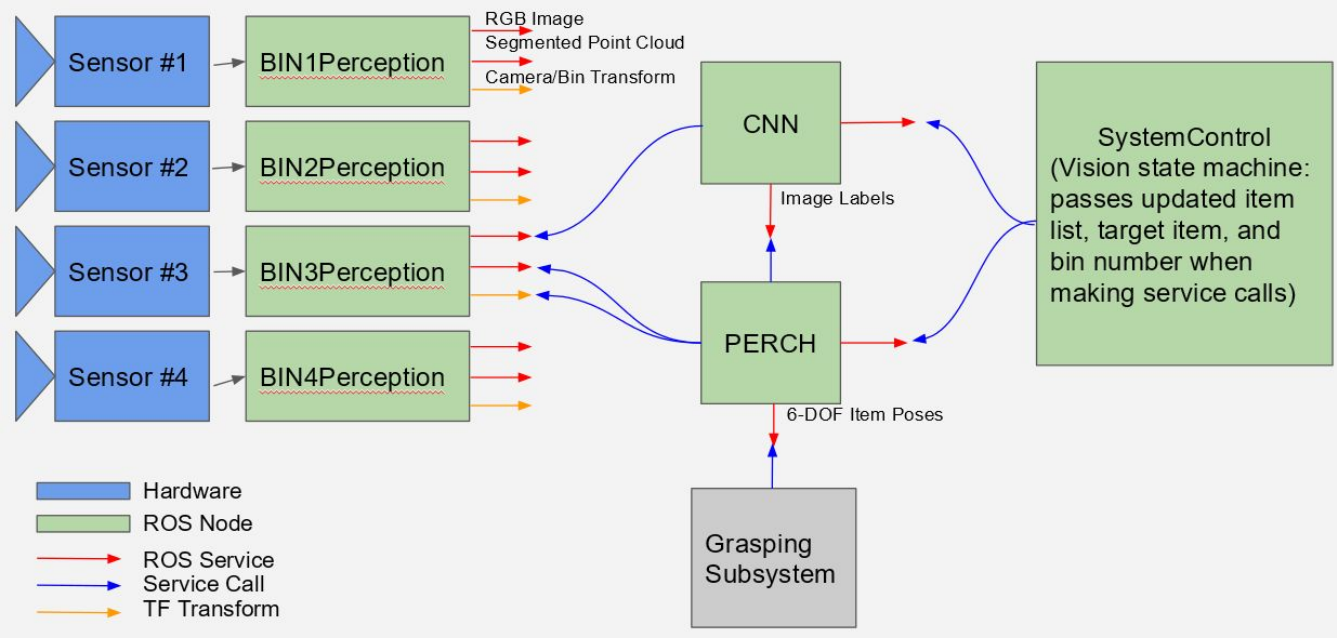

Figure 5. ROS Nodes and Interfaces

The perception pipeline begins with the BIN#Perception nodes. These nodes will perform any sensor fusion for their respective bins, as well as any image and/or point cloud pre-processing (such as segmenting the bin edges out of the point cloud). The vision system will proceed in the following order for each bin:

1. The SystemControl node will call the CNN command ROS service, passing the desired bin number and the updated bin item list to the CNN.
2. The CNN node will call the appropriate BIN#Perception ROS service in order to obtain the RGB image of the bin. It will then process the image and label each pixel with the appropriate item.
3. When the CNN has finished processing the image the SystemControl node will call the PERCH command service, passing PERCH the bin number, target item, and item list.
4. PERCH will call the appropriate BIN#Perception ROS service in order to obtain the segmented point cloud or depth image and the camera transform. It will then process the point cloud in order to determine the 6-DOF for each non-deformable item (deformable items will be left as their original point clouds)
5. The 6-DOF for the items will then be available for the Grasping Subsystem through a service call.