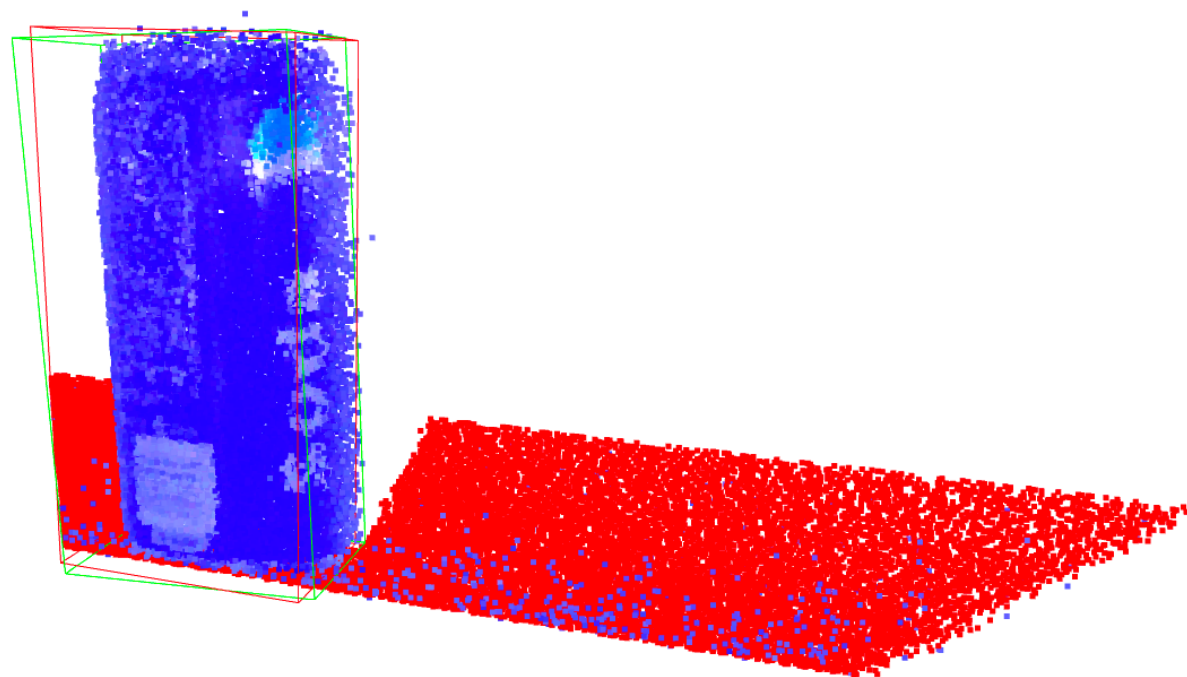


# CSCI 4302 - LAB 2 - POINT CLOUDS

---



The goal of this lab is to get familiar with basic point cloud operations. Point clouds are derived from depth images and allow the robot to perceive its environment in 3D. We will use plane recognition, filtering, and bounding box fitting algorithms to segment and localize objects in the environment. We will then use the state machine from **Lab 0** and the IK solution from **Lab 1** to collect a sequence of cans from random locations on the conveyor belt.

Specific goals

- get familiar with the Open3D library
- understand and implement a basic object detection pipeline
- put MoveL and MoveJ into practice
- understand the limitations of simple FSMs

## OVERVIEW

This lab consists of a slightly modified environment from the previous labs in that cans are placed randomly by a “supervisor”. The robot is equipped with a depth and color camera that are co-located in the robot’s palm, mimicking an Intel Realsense, or similar, stereo-from-depth device. The RGB camera is solely provided for augmenting the point cloud graphically, please do not use it to detect the cans.

You will be furnished with a Python version of the Open3D tutorial. Doing the tutorial requires downloading the Open3D Github repository to gain access to sample data. You will also receive basic code to turn Webots sensing data into Open3D classes.

You will then use Open3D to identify and localize the cans as they arrive on the conveyor belt, then use moveL to move toward them, and finally grasp them.

## REQUIRED FILES

- Download files from **Github**
- To solve this lab, you will need a working implementation of lab0 (state machine) and of the moveL command
- You will need to install the following libraries: open3d, matplotlib. You will also download the Open3D github repo in order to run through the examples.

## PRELIMINARIES

- Install the required libraries and work through the Open3D examples. Not all of the examples are required to solve the lab and you will need additional Open3D functions to solve this lab.
- Navigate to the **Open3D User Manual** and skim through the tutorial, pay close attention to the “outlier removal” section.

## FINDING THE CANS

- Build a recognition pipeline that returns the bounding boxes of the soda cans as they arrive on the conveyor. The following steps are suggested:
  1. Identify the conveyor belt and remove the points associated with it (plane detection)
  2. Filter the point cloud by down-sampling and removing outliers so that only the soda can(s) remain.
  3. Fit a bounding box around the soda cans
  4. Perform a quick sanity check on size and dimensions, then output the bounding box centroids

## MOVING THE ARM

- Use the bounding box centroid to compute the relative motion along the X-axis that the robot has to do
- You can use the current joint angles and forward kinematics as a template for the desired pose
- Think about a way to time hand closure. Don't process the entire point-cloud every time step. Also notice the minimal distance at which the range-finder can see.

## ARM CONTROLLER

- Built up on your statemachine from lab0 to implement a complete grasping pipeline.
- Empty the soda cans into the bin left that is closest to the robot
- The Webots world you were provided with comes with a “Supervisor”. The supervisor is an additional robot controller that has complete access to the world state. Run it in an additional terminal alongside with your robot controller and it will reset the cans to a random location on the conveyor belt every 5 seconds. It will also count the number of cans in the bin after 100s.

## SUMMARY

1. Install and get a feeling for Open3D, a library to manipulating point clouds
2. Turn the range image provided by the range finder into a point cloud and filter it to detect the soda cans
3. Compute the bounding box of the soda can to compute its centroid
4. Use this information to grasp the soda can and throw it into the bin

## DELIVERABLES

- A single python file that detects soda cans using the range finder. You will need to disable all Open3D graphical output prior to submission. Use of the camera for detecting the cans is not allowed. The supervisor file will be used for scoring. (Don't submit your supervisor.)