

CSE 4310: Introduction to Computer Vision

Summer 2023

Program #4: Overhead Box Dimensioning

In this assignment, you will implement a simple 3D box dimensioning system using PCL. You will be given a set of PLY files taken from an RGB-D camera looking straight down at a tabletop scene. In this scene, there will be one or two small boxes. You will write a program that will locate the tabletop and individual boxes, and print the dimensions (width, length, height) of any identified boxes.

The program should take a single command line argument, which will contain the file path for the input point cloud (assumed to be an XYZRGBA PLY file). The program will only operate on a single input file when executed (i.e., loading a different image will require a restart).

Once the program is loaded, the file will be processed and a result will be displayed in the console and display window(s). There will be no runtime user interaction other than zooming around in the 3D viewer. The program should display the 3D window until the user terminates execution by pressing *ctrl + c* in the console.

The result point cloud in the 3D viewer should be colored to represent what is in the scene. The table surface points should be colored blue, and any identified boxes should be colored green or red, with each box colored differently (in other words, each of the 2 potential point clusters over the table surface should have a unique color). If only one box is present in the scene, it should be colored green.

In the console window, you will display an inventory of what is detected in the scene. The length, width, and height of each box, in meters, should be given in the following format:

```
BOX 1: 0.0535 0.1235 0.0255
BOX 2: 0.0455 0.2235 0.0385
```

Your program should run on the class development environment using PCL 1.12.1. You must include a working CMakeLists.txt file with your source code. The program should be able to be compiled and executed by running the following set of commands in the program directory if C++ is used:

```
cmake .
make
./program4 <PATH TO FILE>
```

For example...

```
cmake .
make
./program4 test1.PLY
```

You may develop your application on your own system / OS, but it must run properly on the class development environment at the time it is evaluated. Submit your source code and all other necessary files to blackboard in a single zip file by the deadline. Late submissions will incur a penalty of 10 points per day after the deadline.

Points will be assigned as follows:

1. Program finds and colors the tabletop successfully – 40 points
2. Program finds and colors the boxes successfully – 40 points
3. Program prints correct box heights – 10 points
3. Program prints correct box width – 5 points
3. Program prints correct box length – 5 points

Partial credit may be given, at the discretion of the grader, for items which are not fully functional or contain bugs. You may be asked to demo your program to the grader if errors occur during the initial run. Write your code as cleanly as possible with proper formatting and comments in order to bolster your case for partial credit on non-functional features. It is highly recommended to follow a standard coding style, such as ANSI C++, Google C++, etc. in your program.