

## 1 What I've done

- Modified code to capture the depth image associated with each RGB image and point cloud
- Had a look into the Vicon code
- Covered cube box in pictures to add texture
- Collected more data (checkered box and cube box as scene), didn't manage to get a full run without crashing though. It got about halfway around the boxes both times.

## 2 Parts of report to look at

- Nothing new.

## 3 Questions

- 

## 4 Comments

- The quadcopter is actually publishing its commands, so I'm recording a rosbag of that topic now
- The top box is not really in the image being captured by the RealSense (see Figure 1). We might need to make it fly a bit higher or closer to the boxes.
- The depth image doesn't look great (see 2), but it's the same as the depth image in the example code.
- My quad crashed pretty badly and the frame slipped out. It might take me a while to fix it, and I think I need some screws.
- The quadcopter was drifting a bit towards the far wall when I manually hovered it just off the ground. Not sure if this is related to the crashes.
- The RGB and depth images sometimes don't save properly (they save but can't be opened). Not sure if this is a problem of not getting data in or something in the code.

- The Vicon started detecting an extra marker. I didn't manage to get rid of it through fiddling with the strobe intensity and threshold or through putting electrical tape on the quad. I eventually just deleted one of the quad markers and added the new marker.



Figure 1: Typical image with circle trajectory height and radius both set to 1.2m

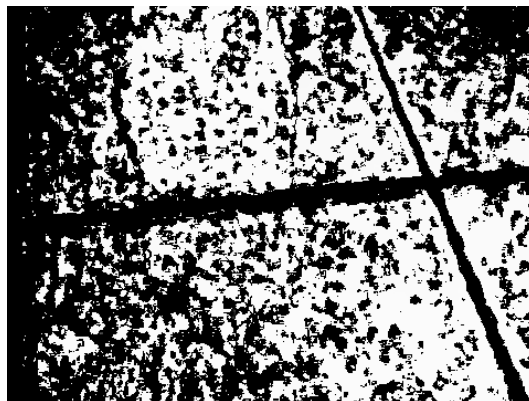


Figure 2: Depth image corresponding to Figure 1

## 5 Stuff to do

- Fix quadcopter
- Adjust quadcopter trajectory so that it can see the top box
- Investigate data

- 
- ~~Investigate boxes to determine which ones can be picked up as point clouds~~
  - Investigate registration algorithm
    - Generate 3D object in MATLAB that I can get points from from various camera poses (may also need to get RGB and depth images if we're going with that approach?)
    - Apply registration algorithm to generated point clouds (ground truth known) – without noise first, then add noise. Get error in true and estimated translation and rotation.
  - Reading
    - Incorporating RGB and depth images – feature matching