

Individual lab report #11

4/18/2017

Jin Zhu
Team E

Teammates: Michael Beck, Akshay Bhagat, Matt Lauer, Leo Lu, Jin Zhu

1. Individual Progress

1.1 Lighting

Inconsistent lighting condition, such as glare and shadow shown in Figure 1, will cause our neural network to fail on recognizing objects. My major task this week was to get a good lighting condition inside of the self. During the competition, we will be putting a tent around the self, so the ambient light would be blocked off. Thus, we can have the controllable lighting inside of the self (Figure 2) matching what we used to collect training images for neural network. I tried several types of lighting conditions, and the final design is to point LED strips upward, the light will reflect on the fabric on top, bounce back and pass through the fabric underneath, as shown in Figure 3.

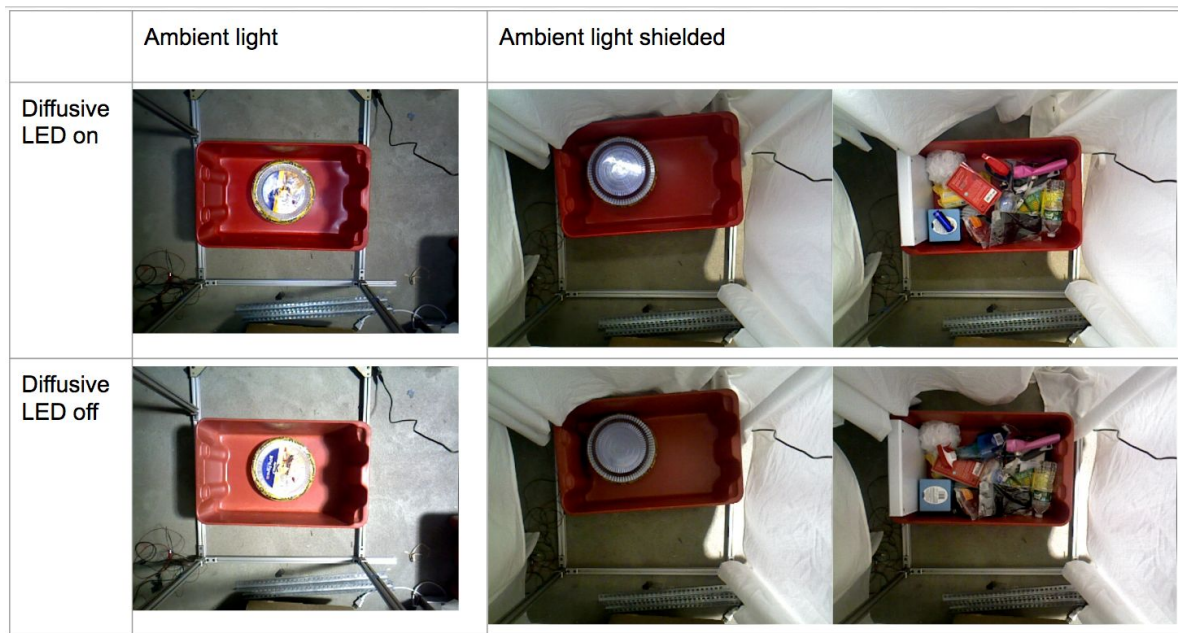


Figure 1. Images taken in different lighting conditions. This setup is intended to be used for data collection for items in the tote

The two sets of LED on the two sides are similar distances to the bins, so they are connected in series. The set of LED strips in the middle are closer to the top bin, so they are powered separately and connected to a dimmer. Figure 4 shows how the LEDs are mounted and powered.

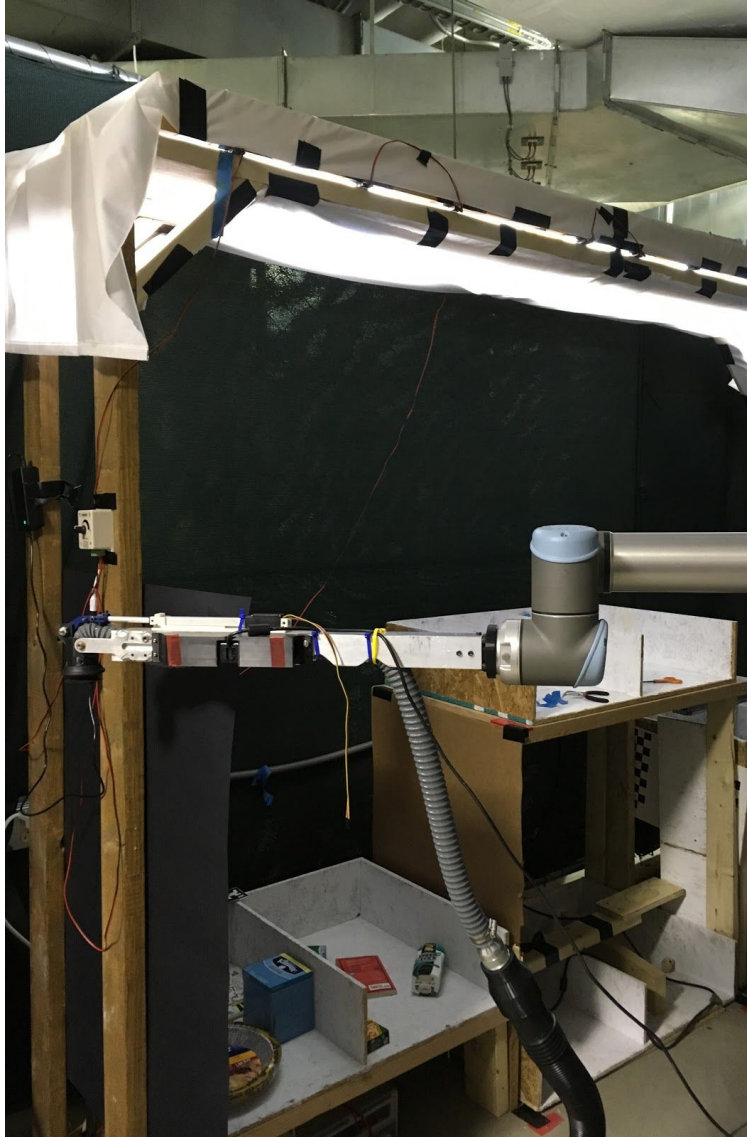


Figure 2. New lighting condition

During fabricating the new lighting fixture, I learned some useful tools, including drill, band saw to cut aluminum, wood saw and soldering to power LED strips.

Our system will need to recognize items placed on the shelf and inside the red plastic tote for picking and stowage separately. Thus, I also tried to make an isolated environment (Figure 5) to collect separate images for the red plastic tote that we will use during stowage. Things I tried including placing LED at different locations and compare the reflection result. We might not use this in the end if we decide to only collect data for items in the shelf because of the shortage of time.

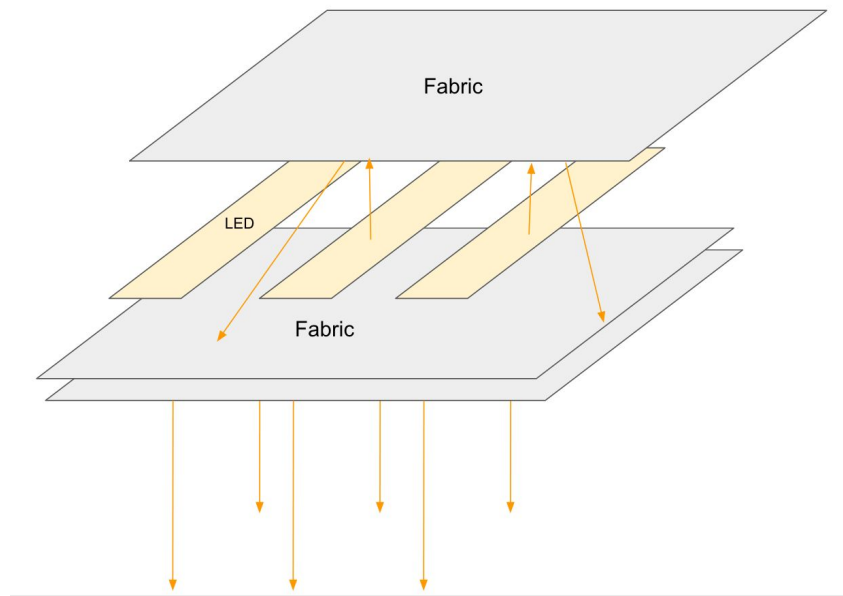


Figure 3. Diffused lighting condition design

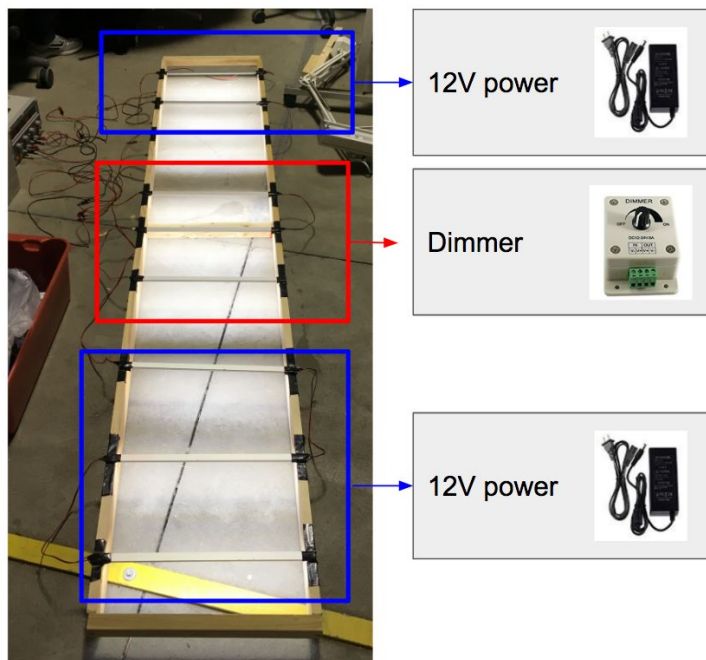


Figure 4. Mount and power of LED strips



Figure 5. Isolated environment to collect data in plastic tote

1.2 Integrate image taking

I integrate the ROS node for taking RGBD images with Matt's code for arm planning. Matt's code loops the arm through a sequence of 2 prep poses outside tote and 4 poses above the tote. After the arm reach a pose, the ROS node will call the ROS service I wrote for capturing RGB images and depth images from the ASUS camera. When the user put in item name, the integrated code can run 4 poses to take pictures at top down, left tilt, right tilt and backward tilt angles. Figure 6 shows an example of the 4 images at 4 different poses



Figure 6. Images captured under 4 poses in the new lighting condition

1.3 Masking

Unknown objects will be released 30 mins before the competition, so we are planning to collect data in 30 minutes and do training on them. I'm working on creating a binary mask that separates out background and objects for individual images, since we won't have time for manual labeling. There were three methods I explored. The first one is use the FCN output result, and the second one uses RGB images.

For method 1, I used the pixelwise labeling from FCN for the given image. FCN labels give a score for each pixel for each of the 41 classes. Out of the 41 scores, the class with the maximum score would be the class of the pixel. I was trying to find a threshold, such that pixels with score for background higher than certain value would be classified as background, otherwise as object. I modified the method to find threshold this week. The new method looks at regions that are object classified as object, background classified as background and object classified as background. Figure 7 is an example of these three instances. White region are pixels identified by FCN as object. The assumption was that if the background score minus the maximum of all the object score is higher than certain threshold, the pixel should be classified as background. To experiment on that, I randomly selected 100 points out of the selected region, as shown in Figure 8. Figure 9 is the histogram of the result. "Object as background" and "background as background" had many overlapping regions for background score

minus maximum object scores. This makes it hard to find a good threshold. Thus, I tried the second method, which uses RGB images.



Figure 7. FCN classification result. Blue box is object classified as object, green box is background classified as background and red box is object classified as background



Figure 8. 10 randomly sampled points per case. Red pixels are object classified as background, blue dots are object classified as object, and green pixels are background classified as background

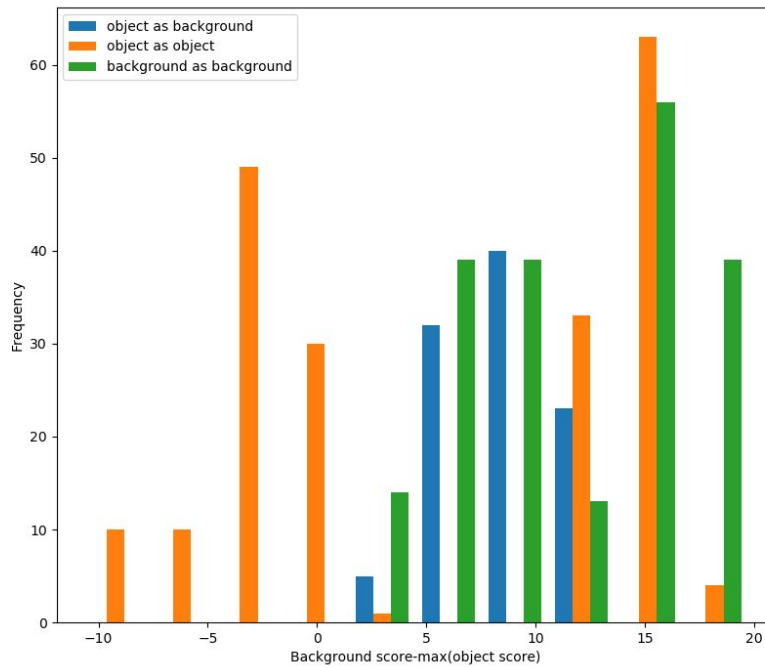


Figure 9. Histogram for the each type of pixels having certain values for background score minus maximum object score

The second method subtracts the empty tote image from the image with items. The result is an image with items and very low intensity for background, as shown in Figure 10. I will work on improve this result and generate a good mask from this.



Figure 10. Masking using RGB images

2. Challenges and problems

The challenge for model training now is not enough time to label enough training data. The old model was trained on Kinect with different lighting condition and background. To have good accuracy for SVE, we need to retrain models. The most time consuming part would be label the images using LabelMe. One way to outsource labeling would be using Amazon Mechanical Turk (AMT). However, AMT doesn't provide direct support for LabelMe, and it might take many days to figure out the API for it. I emailed AMT customer service and talked with some students with experience of using Amazon Mechanical Turk for some other purposes. Basically, we need to write a javascript that does the same job as LabelMe, so that people can draw polygons around objects and provide a name for the object. Also set up the html link to Amazon Web Service (AWS) for collecting the data and generate code for finished jobs. This seems like a complicated process, if we couldn't figure this out, then we have to label images by ourselves, or try to find a lab at CMU which already had it setup.

3. Teamwork

All the teammates worked a lot for this progress review. Michael worked on shelf, bin fabrication, and fixed quaternion issues for grasping. Akshay also helped with grasping. Leo worked on change the system to bin mode, improved the state machine, and worked on fusion of point clouds. Matt worked on slider, 7-DOF, planning scene, and camera poses for taking training data.

Michael helped me with brainstorming the lighting setup, purchase of the materials, taught me the use of the band saw to cut mount for LED mounting, the use of wood saw for wood frame and how to make good soldering. Matt helped me with understanding how to power the electronics and control LED with the dimmer. Akshay helped me with putting together the frame and taught me how to use drill. I worked with Matt to integrate the camera with arm. Michael also helped me with masking by thinking of new methods to find the threshold to separate out background and the object. Leo provided me a function to segment out object using point cloud depth information and april tag localization result, so there is a second way I can compare my segmentation using FCN with.

4. Plan

Michael and Matt will work on planning for 7 DOF, use point cloud as obstacle. Michael will work on shelf and 80/20 aluminum ordering. Akshay will work on collecting training data, and IK for grasping. Leo will work on integrate FCN to the current system, so that FCN result can be used for point cloud segmentation and the objects point cloud can be passed to grasping.

I will work on masking for unknown items and assist Akshay to collect training images. I'm using opencv for image processing right now, and I'll try to find some good module for background subtraction.