Computer Vision Assignment Report

I will first provide a rough outline of the design of my system and then go into more detail about the choices I made. There are three main stages to my system; first, a dense stereo disparity map is calculated; second, objects within the scene are detected; third, information from the disparity map is combined with the detected objects to give estimates to the distance from the camera for each object.

Disparity mapping stage:

1. Perform histogram matching between the grayscale image pairs
2. Compute the disparity using SGBM
3. Apply median filter to the disparity map
4. Threshold disparity to maximum of 128

Object detection stage:

1. Cover up car bonnet with a fixed black polygon (defined by me)
2. Perform YOLO and obtain bounding boxes
3. Ignore objects which aren't one of person, bicycle, car, motorbike, bus, train, truck, traffic light, stop sign

Object distance estimation stage:

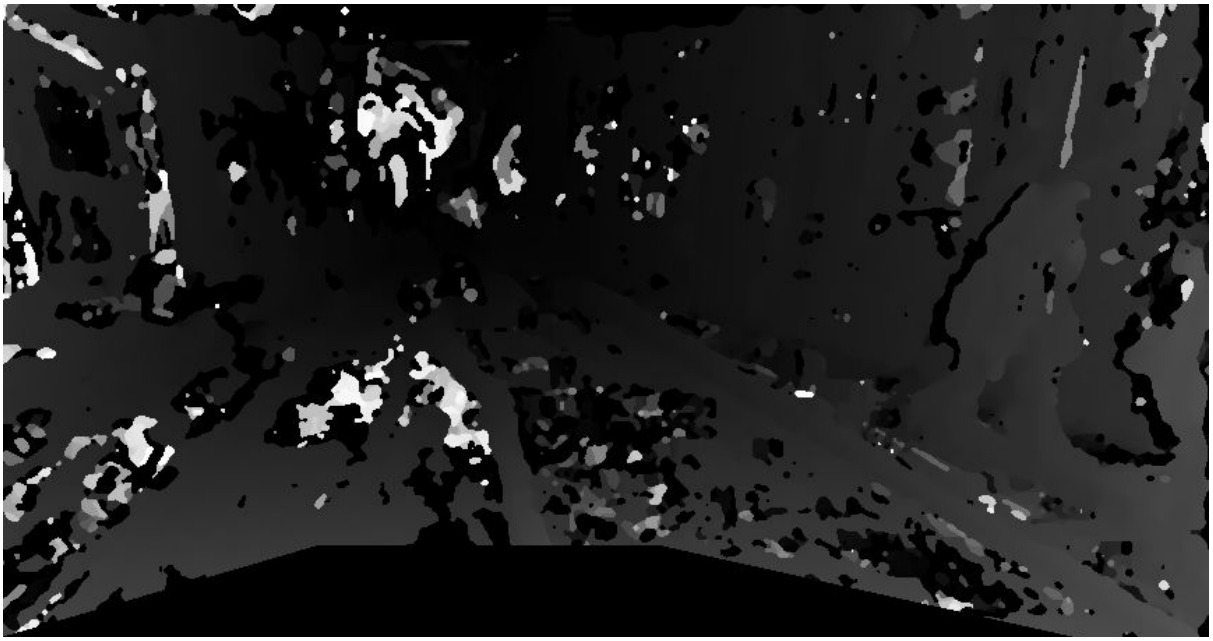
1. For each bounding box, calculate the mean disparity from the ROI in the disparity map
2. Sort the bounding boxes according to their mean disparity in descending order
3. For each box in this sorted list
    a. Compute the median of the defined disparity values (>= 0) from the ROI in the disparity map
    b. Calculate the distance using the formula *camera_focal_length*camera_baseline/median_disparity*
    c. Create a box which is 7/8 the size of the original bounding box (same centre), and set the values in this ROI to -1 in the disparity map (so that they are undefined)
4. Draw the bounding boxes and predicted labels and distances in reverse distance order, so closer objects' labels aren't obscured by more distant ones

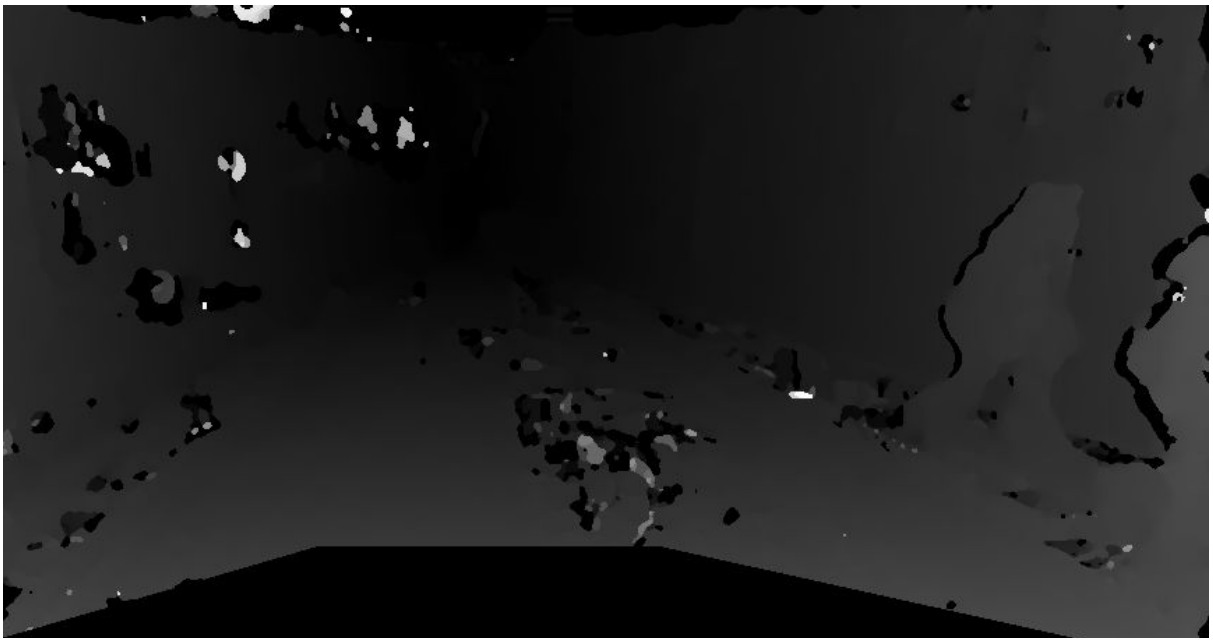
Disparity Mapping Stage

In my experiments I found that performing histogram matching between the left and right image pairs resulted in much less noise in the resulting disparity map. Histogram mapping is the process of transforming the pixel values of one image so that the histogram of these values matches a target histogram as closely as possible. In the two images below, the grayscale left and right image pair, you can see that the exposure of the right image is much higher than the left image.

This happens because the two cameras can pick up different amounts of light and so the sensor hardware compensates by adjusting the exposure. The problem is that the exposure of each camera might be different. In order to get the best results from dense stereo matching algorithms like semi-global block matching, the pixel values corresponding to the same object should be very similar between the two images. After histogram matching, the right image looks like this:

And here is the difference in the computed disparity maps, before and after:





Object Distance Estimation Stage

In order to overcome the problem of obtaining a distance estimate when an object is partially occluded, I did two things. First, I sort the objects according to an initial distance estimate, calculated from the mean of the disparity values in the bounding box. Then, I remove these disparity values from the map so that they aren't used to calculate the distance for object which are behind the objects already processed. Secondly, I use a robust measure of average which is the median of the disparity values. This helaps reduce the influence of bad patches of the disparity map (where the values are undefined or extreme). In the example below, you can see the distance to the truck remains the same when a person walks in front of it.