Stereo Vision for Object Detection and Distance Ranging

wgvb97

6^{th} December 2019

1 Approach

1.1 Object Detection

For object detection, I edited the implementation of YOLOv3 provided by Toby Breckon (2019) based on Redmon's and Farhadi's (2018) original description.

The most significant addition I made is to apply contrast limited adaptive histogram equalisation (CLAHE) to enhance the contrast in local pixel neighbourhoods of the image. This approach is preferred over regular histogram equalisation as it works over local areas to redistribute the lightness over the global image (Yadav, Maheshwari and Agarwal, 2014). In my implementation, I apply CLAHE to the value channel of HSV space of the frame inputted to YOLOv3 to brighten darker areas of the image to reveal more objects in the shadows (Figure 1). However, this approach risks losing information in brighter areas of the image, causing certain objects to go undetected (Figure 2).

Figure 1: From left to right: Before and after applying CLAHE to a man in the shadow of a building.

Figure 2: From left to right: Before and after applying CLAHE to a white van in the sun.

1.2 Stereo Disparity Map Calculation

My stereo disparity map creation is also based on an example from Breckon (2017).

1.2.1 Pre-processing

As well as the pre-processing included in the original code, I performed global histogram equalisation on the greyscale image pairs. This technique is a simple way to improve the contrast of low visibility images, improving accuracy of the disparity map for low-light images (Aditya, Reddy and Ramasangu, 2014) (Figure 3).

Figure 3: From left, to right:
Original image, unfiltered disparity map, histogram equalised map

1.2.2 Post-processing

In addition to the thresholding and speckle filtering provided, I produced two disparity maps with respect to both left and right images. To these maps I applied the weighted least squares filter to produce a single, smoothed map, with respect to the left image (Figure 4). This approach allows for fast global smoothing, whilst preserving edges (Min *et al.*, 2014).

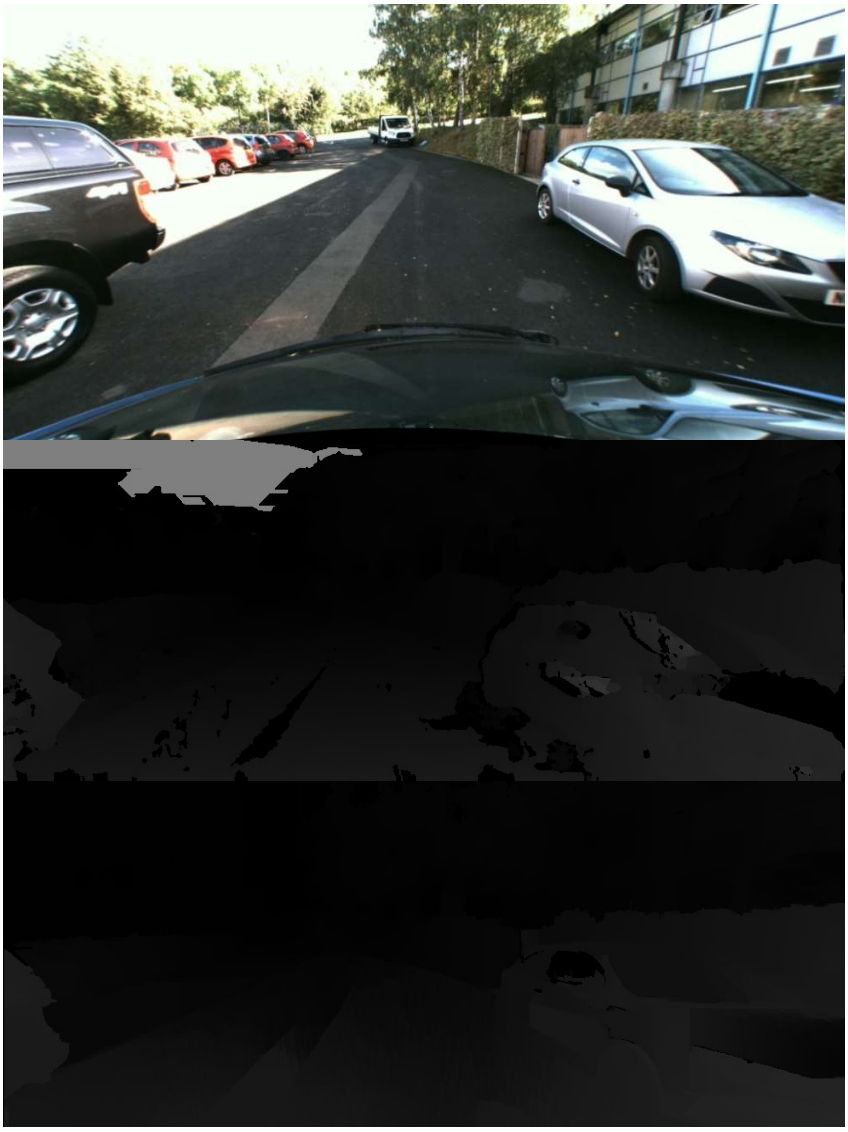


Figure 4: From top to bottom: Original image; unfiltered disparity map and WLS filtered disparity map

1.3 Distance Calculation

The challenge in this section is reducing the disparity map to a single value representing an object.

My first instinct was to use the mean of the area drawn out by the object's bounding box. However, this was susceptible to the influence of the background, leading to the distance being over-estimated.

Figure 5: Distances calculated using the mean of disparities.

I instead used the median of this area but found the distance to be smaller than expected. So, to lower the impact of the distant background, I calculated the median after filtering out disparity values below a threshold. This brought the distances closer to the expected values but were still smaller than I had expected, especially when partially occluded as the distance measured would instead be the distance to the obstacle. To remedy this, I experimented with values around the lower quartile, before settling on the 15th percentile. This provided strong estimates for the distances in most cases but was still subject to a lot of variance as some bounding boxes are tighter to the object than others, leading to the distance to the background being returned.

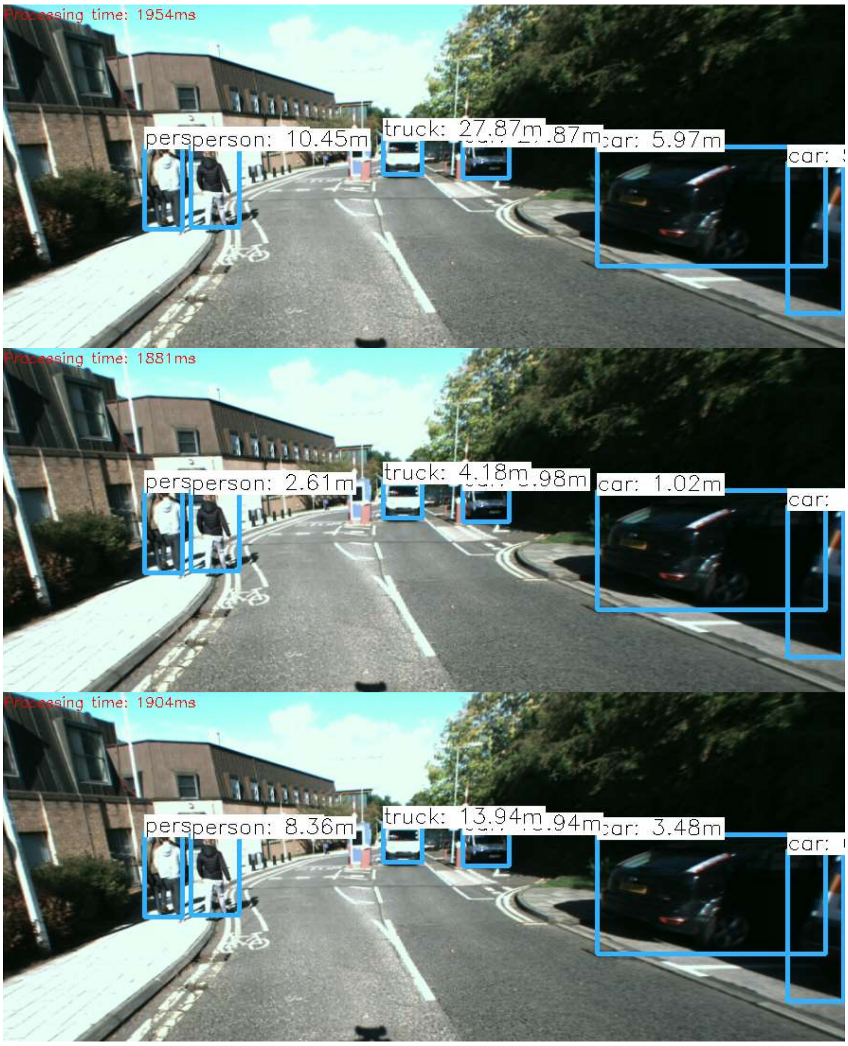


Figure 6: From top to bottom: Distances calculated from the median of disparities; the median of non-zero disparities and the 15th percentile of non-zero disparities.

Finally, I decided to use the mode of the non-zero disparities in the bounding box. This method was effective as the detected object takes up the largest area of the bounding box meaning that the modal disparity is the disparity of the object. The only occasion where this is not the case is when the object is occluded and the obstacle takes up more of the bounding box than the object. However, in this case, I find the desired object is seldom detected.

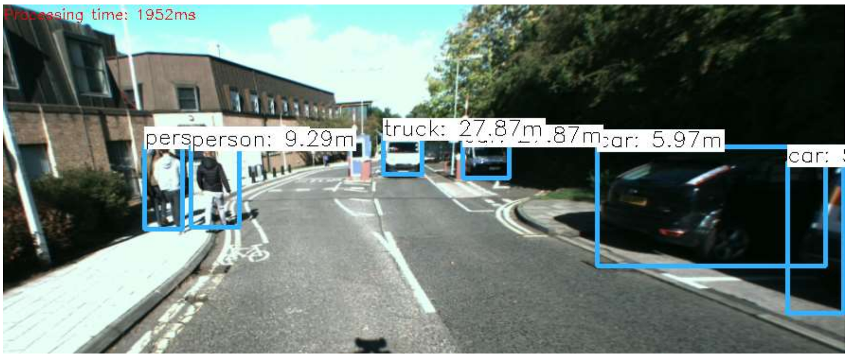


Figure 7: Distances calculated from the mode of the non-zero disparities.

1.4 Sparse Stereo Ranging

In conjunction with the dense stereo ranging program specified in the assignment, I have developed a sparse stereo ranging program for comparison.

2 Results

2.1 Accuracy of Dense Stereo Ranging

My implementation of dense stereo ranging passes the 'eye test': distances seem feasible and decrease as the car approaches. Furthermore, as a result of the filtering applied, my solution is robust in challenging conditions like shadows and occlusion. However, simply passing the 'eye test' is insufficient proof of success so I have measured the true distance in cases where the car and object's positions are easily estimated by examining the surrounding landmarks:

Figure 8: Dense estimation of the distance to the truck is 27.87m, estimate provided by Google Maps is 24.93m.

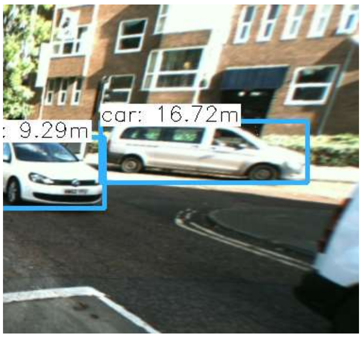
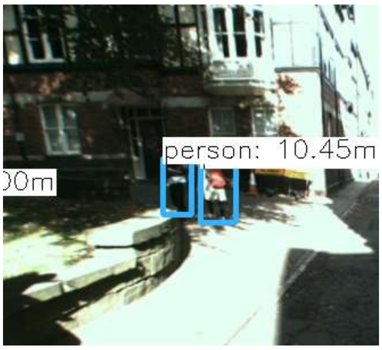


Figure 9: Dense estimation of the distance is 16.72m, estimate provided by Google Maps is 17.90m.



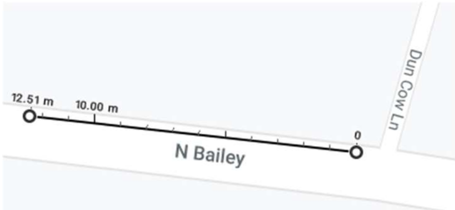


Figure 10: Dense estimation of the distance is 10.45m, estimate provided by Google Maps is 12.51m.

In these examples, the estimates generated by my dense stereo ranging implementation are, at worst, 20% different from the true value obtained from Google Maps.

2.2 Comparison of Dense and Sparse Stereo Ranging

Below are some examples of frames from the dataset with the distances as specified by both programs and the true value:

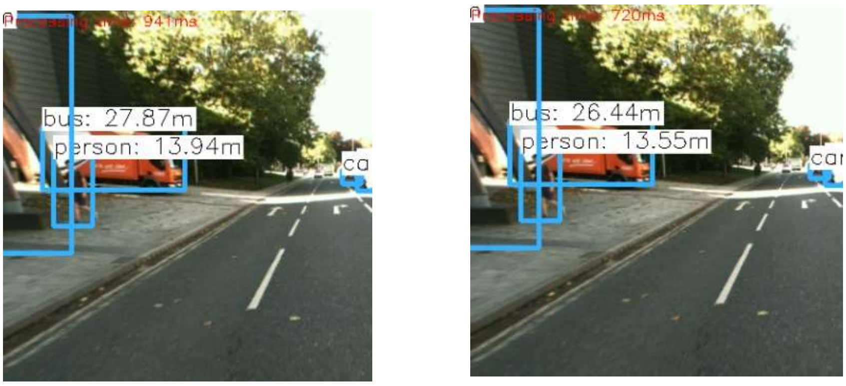


Figure 11: From left to right: Dense stereo estimation and sparse stereo estimation. The true value estimate was 24.93m.

Figure 12: From left to right: Dense stereo estimation and sparse stereo estimation. The true value estimate was 17.90m.

Figure 13: From left to right: Dense stereo estimation and sparse stereo estimation. The true value estimate was 12.51m.

In these examples, sparse produces results of a comparable quality, but in a much quicker timeframe. As this is an application for use in autonomous vehicles, sparse is a much better algorithm for object ranging as all processing must be done in real-time to allow for the car to react to sudden changes in the environment.

3 References

- Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Breckon, T. (2019). *tobybreckon/python-examples-cv/yolo.py*. [online] GitHub. Available at: https://github.com/tobybreckon/python-examples-cv/blob/master/yolo.py [Accessed 6 Dec. 2019].
- Yadav, G., Maheshwari, S. and Agarwal, A. (2014). Contrast limited adaptive histogram equalization based enhancement for real time video system. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.
- Breckon, T. (2017). *tobybreckon/stereo-disparity*. [online] GitHub. Available at: https://github.com/tobybreckon/stereo-disparity [Accessed 6 Dec. 2019].
- Min, D., Choi, S., Lu, J., Ham, B., Sohn, K. and Do, M. (2014). Fast Global Image Smoothing Based on Weighted Least Squares. In *IEEE Transactions on Image Processing*, 23(12), pp.5638-5653.
- Rublee, E., Rabaud, V., Konolige, K. and Bradski, G.R., 2011, November. ORB: An efficient alternative to SIFT or SURF. In *ICCV* (Vol. 11, No. 1, p. 2).