Design Solution:

The images were tested with a:

- Gamma correction
- Power-law transformation
- Bilateral filter
- Unsharp masking
- Edge-enhancement using a Canny edge detector

For the stereo imaging, the final values are:

- Gamma=1.5
- Power-law value=0.8
- Kernel size=11
- Maximum disparity=128
- Unsharp masking selected over bilateral filter
- Canny edge detection not used

The yolov3 dataset and neural network, by pjreddie, was used for object detection, with values of:

- Confidence threshold = 0.5
- Non-maximum suppression threshold = 0.4

The image is pre-processed with an unsharp mask at k=11.

SGBM was used for stereo ranging. Given the disparity map, and using the boxes found using the yolov3 object detection, disparities were extracted and associated with each detected object. In the case of overlapping objects, any overlap between the boxes have their disparities set to 0. When selecting a representative value, 0 values are ignored. In a situation where all the disparities become 0, we use the original disparities instead. If the disparities within the box are 0 regardless, we do not draw such object.

To select a representative value, multiple approaches were tested:

1. Select the modal disparity value within the box
2. Draw a histogram and select the minimum value from the range with the highest frequency
3. As above, but using the mean instead of the minimum

Results:

Images used for testing were selected based on:

- Number of scene objects
- Image lighting/exposure
- Object proximity
- Object Depth

Images are skewed towards higher-difficulty situations, hence some images will have many objects, poor exposure/lighting, densely clustered objects, many distant objects, or a mixture.

Image Preprocessing:

We have tested:

- The use of an unsharp mask versus a bilateral filter,
- The effect of kernel size on the above filters,
- Further edge enhancement with a Canny edge detector,
- And colour correction through gamma correction and power law transformation,

As seen in the figures below:

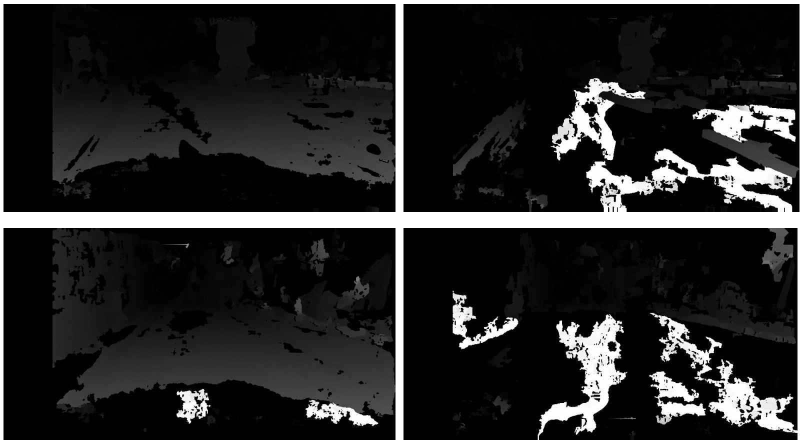


Fig.1 Disparity images of two scenes, using Unsharp Masking (left) and Bilateral Filter (right), k = 41

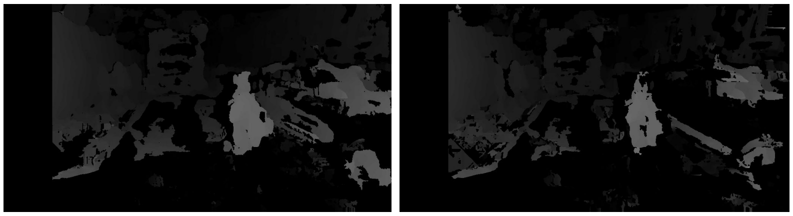


Fig.2 Disparity images of a scene, using Unsharp Masking (left) and Bilateral Filter (right), k = 11

At both low and high k, unsharp masking produces less noisy disparity maps than the bilateral filter.

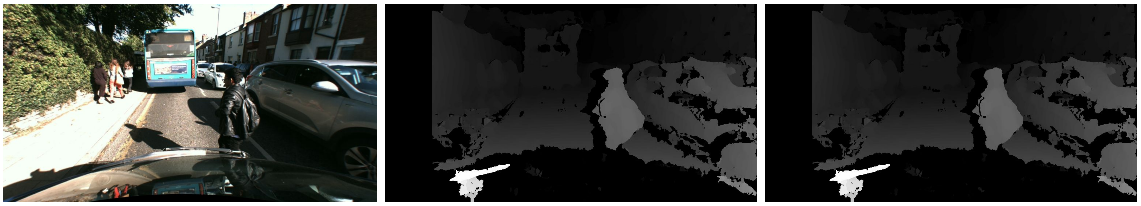


Fig.5 Scene image with disparity images, with Canny edge enhancement disabled on the left, and enabled on the right

Canny edge enhancement does not produce noticeable improvements.

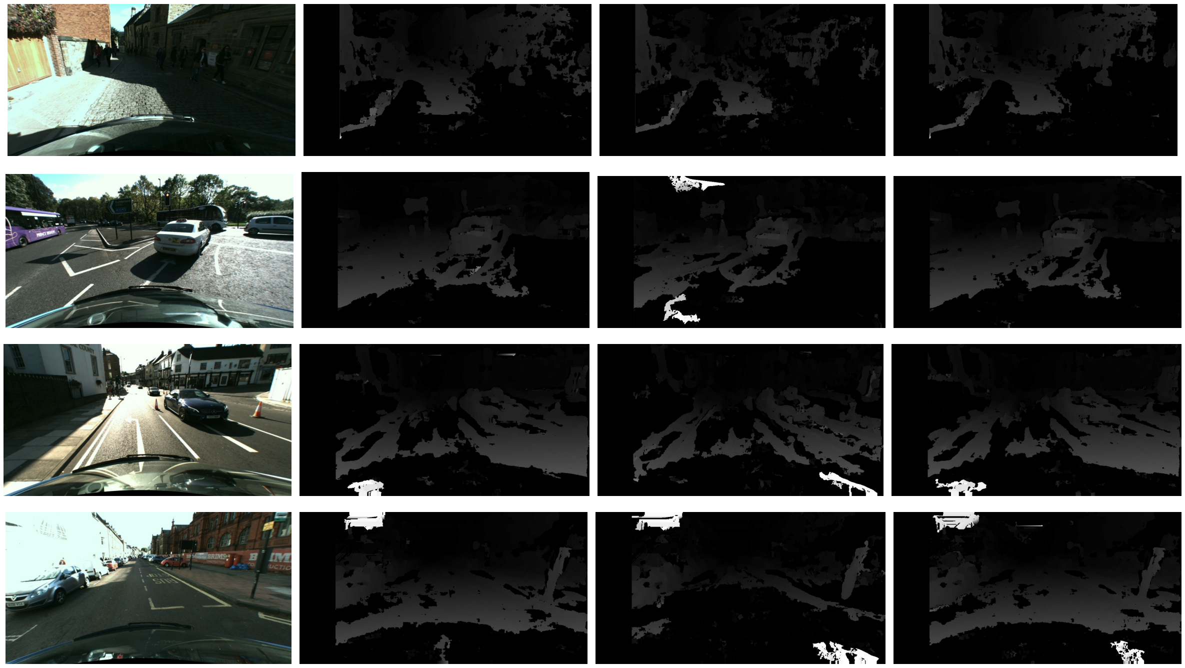


Fig.3 Scene disparities with changing kernel size. Original in col.1, k=11 in col.2, k=25 in col.3, k = 41 in col.4

Low and high kernel sizes have similar results, hence we choose a smaller kernel to reduce processing time.

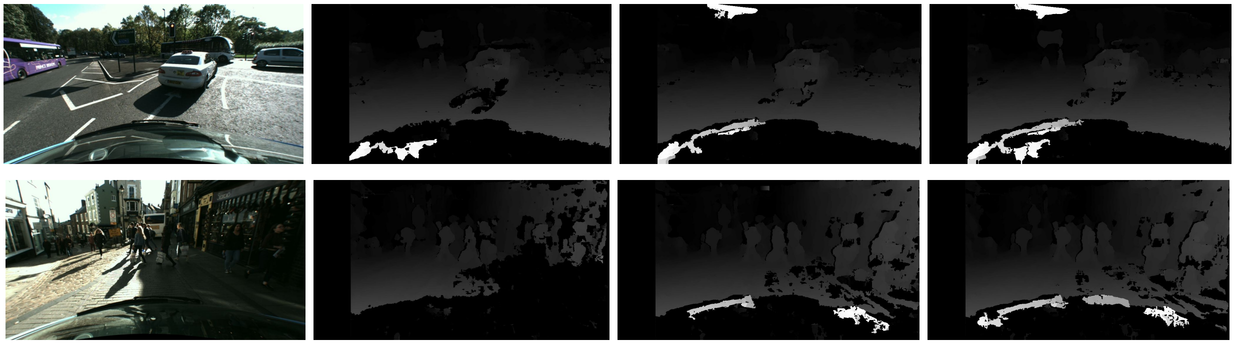


Fig.4 Scene images followed by disparity maps for two scenes, at g=0.5 (left), g=1.0 (middle), g=1.5 (right)

Increasing the gamma produces less noise, and more detail is kept.

The configuration we have settled on is:

- Unsharp Mask, k = 11
- Gamma Correction, g = 1.5
- Power Law Transformation, p = 0.8

Object Detection:

Image	Objects Detected			
	None	Gamma Correction	Unsharp Masking	Canny Edges
1506942513.475180	4	5	4	3
1506942765.475656	7	6	6	6
1506942962.483417	7	6	7	6[1]
1506943009.480358	10	11[2]	12	10
1506943033.477067	12	11	12	11[1]
1506943461.478259	8	8	7[3]	7[3]
1506943538.486899	9	7	8	12
1506943556.482920	8	4	7	9
1506943575.484109	12	11	12	11
1506943595.482292	5	6	6	5
1506943686.478904	9	9	9	8
1506943927.380365	0	1	0	0

[1] Bus not detected [2] Reflection detected as person [4] False positive removed

Fig.6 Table of objects detected in selected images, separated by filter

Image processing techniques have some effect, with no consistent improvements shown. Gamma correction often results in fewer detection. Unsharp making helps with detecting people to a small degree. Canny edges is the most inconsistent of all the filters - at times it would help detect significantly more person objects, and at times less, but often reduces the number of vehicles detected. In testing, we found that the improvements from these filters were inconsistent. In Fig.7, the confidence for a person object decreased when using Canny edges, yet we were able to detect an extra person as well.

Upon combining the filters, we found that some confidences improve significantly, whilst others decrease significantly.

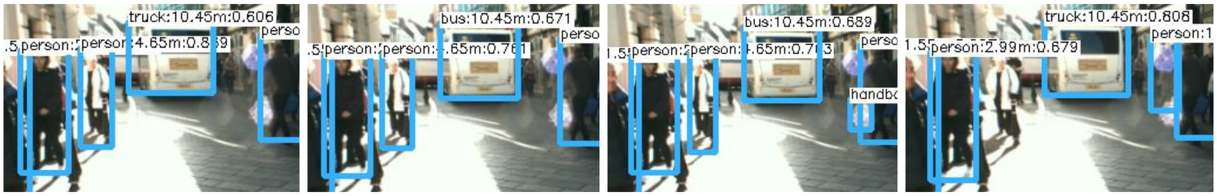


Fig.7 Final output images with confidence values, using no filters (col.1), gamma correction (col.2), unsharp masking (col.3) and Canny edges (col.4)

Stereo Ranging Performance:

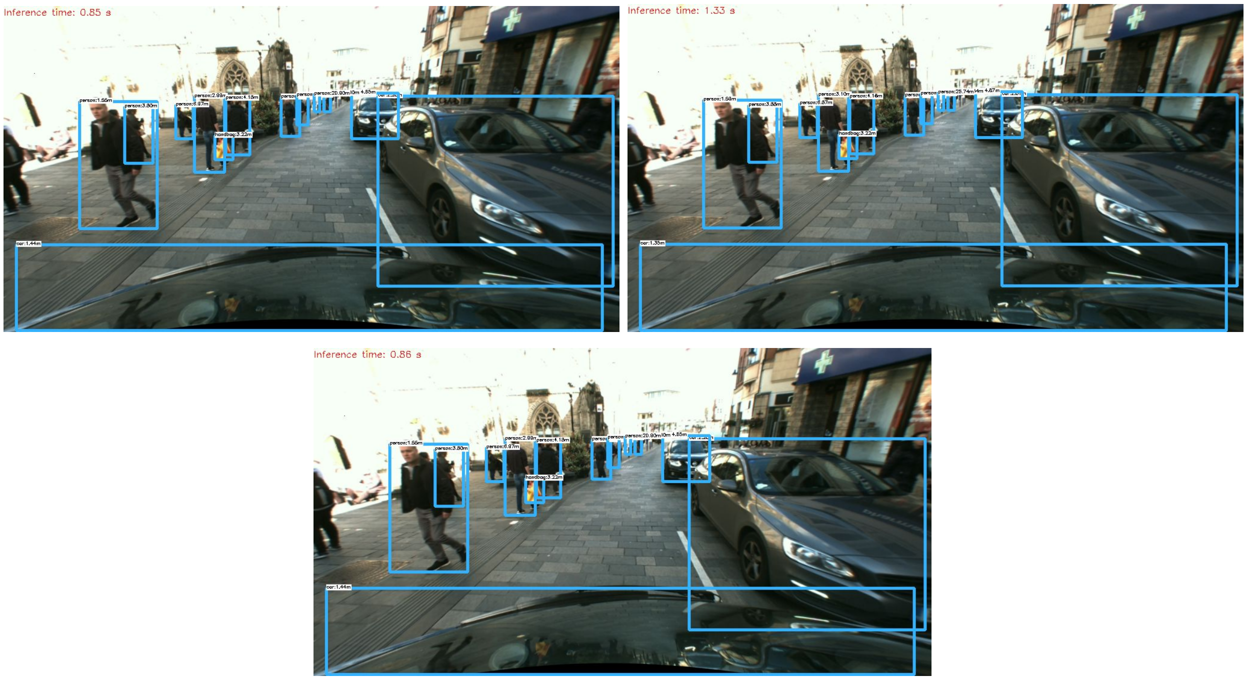


Fig.8 Output images with detected objects and their distances, calculated using histograms (top) or the mode (bottom)

The distances measured with method 1 are often identical to that measured with method 2. With method 3, distances are more varied, but are generally estimated to be further away. In a safety critical situation, reporting the shortest distance is safer, but results in many objects having the same distance, despite obviously being at varying distances apart.

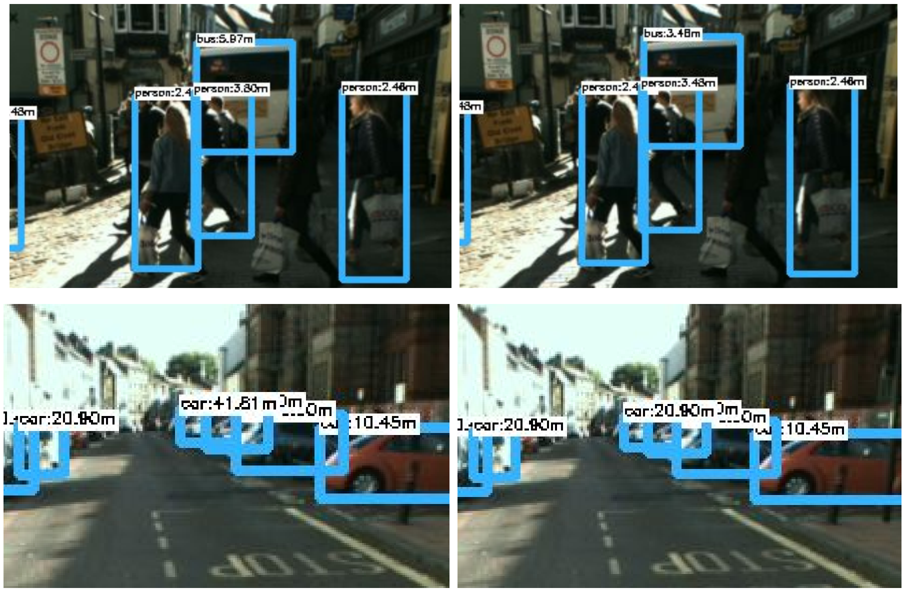


Fig.9 Output images with detected objects and their distances, with overlap correction (left) and without (right)

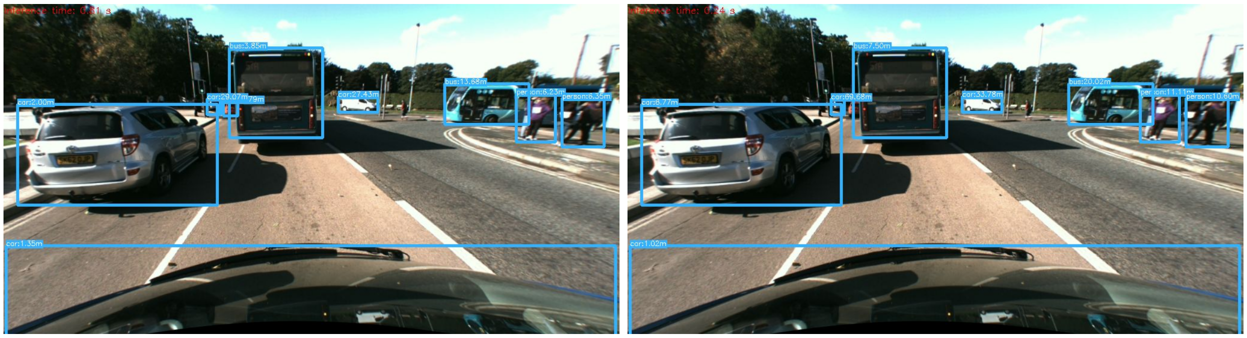
The overlap correction algorithm proves to be effective - the top row improves on the distance from the bus, and the bottom row fixes the distance of the furthest car.

Processing Efficiency:

For speed optimisation, we compared an opencv (CPU) neural net approach to a pytorch (CUDA/GPU) approach. The opencv network ran at 0.84s per frame, and the pytorch approach ran at 0.55s per frame. However, the pytorch approach was very inconsistent for SGBM.

Comparison to Sparse Stereo Ranging:

We've implemented a sparse approach using ORB feature matching. In conjunction with the pytorch network, this ran at 0.22s per frame. All image preprocessing was kept the same.



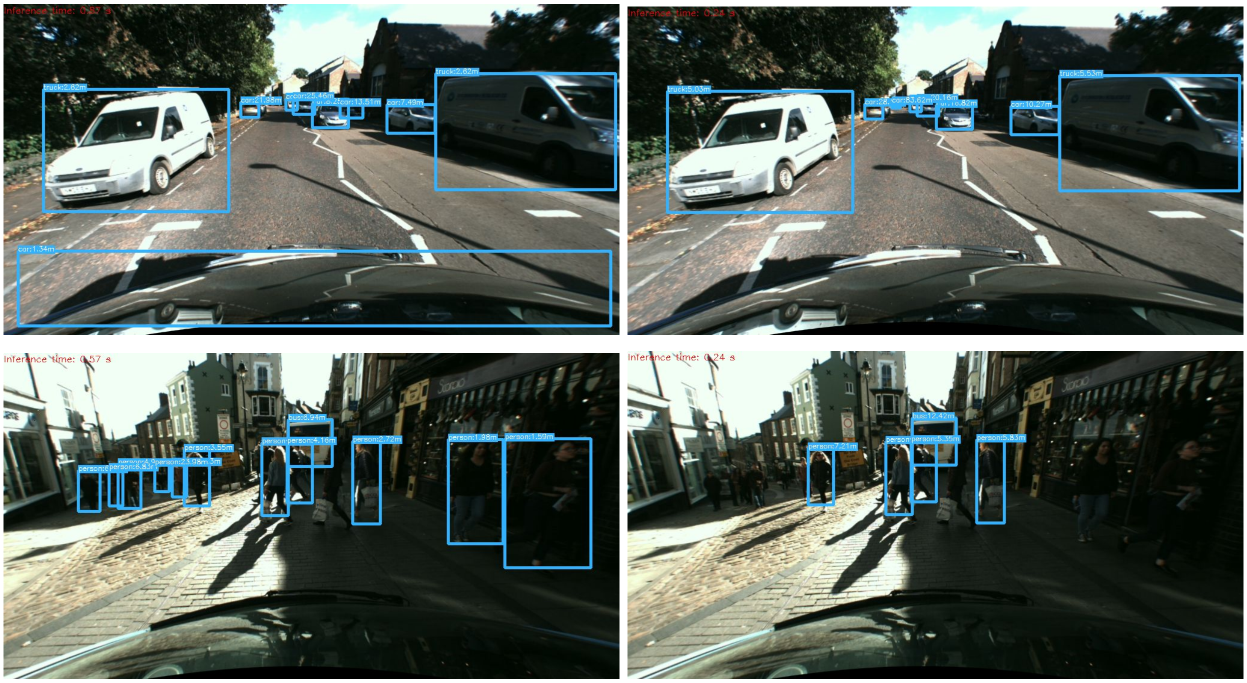


Fig. 10. SGBM Output Images (left) in comparison to Sparse Approach Output Images (right)

In fig.10, it is often the case that SGBM detects more objects than the sparse approach. This is due to the sparse approach not matching enough points throughout the image, hence certain objects detected by yolov3 do not have any matched feature points within them. This results in an assumed 0 distance, hence the box is not drawn. However, using method 3 to calculate distance, we find that the distances seem a lot more realistic as opposed to in SGBM. In order to truly measure this, ground-truth depth maps would be needed for comparison.