2014

Matlab Binocular Stereo System

WRITTEN REPORT

RICARDO SÁNCHEZ CASTILLO

UNIVERSITY OF NOTTINGHAM | G54VIS Computer Vision

Introduction

Zhang and Shan, *A Progressive Scheme for Stereo Algorithm* is a mix between Sparse and Dense algorithms, basically it starts taking a set of features and it labels the rest as UNKNOWN, then it will take pixels with a highly textured region and small uncertainty in their disparity and it will label them as MATCHED or NOMATCH.

The current project proposes to use Zhang and Shan's algorithm with some differences. It will start around the matched pixels in each iteration and it will use the SURF features as an initial point.

The purpose of this document is to describe and analyse the implementation of the proposed algorithm. First section contains a description of the implementation. Second section presents the results obtained and a comparison with the ground truth provided by the Middlebury dataset. Finally, an evaluation of the method is presented.

Description of algorithm

The steps used for the algorithm is:

1. Read the left and right images
2. Obtain up to the second image pyramid reduction.
3. Extract the SURF Features using *detectSURFFeatures* function in Matlab for each pyramid.
4. Match the features between the left and right images by using *matchFeatures* function in Matlab for each pyramid.
5. Delete all those matches which does not satisfy the epipolar constraint. I.e. those features matched in different heights.
6. Merge all matching points of each pyramid into a new lists called *LeftMatchedPoints* and *RightMatchedPoints*.
7. Create a matrix filled with zeros with the same size as the original image except for those points corresponding with the *LeftMatchedPoints* which are labelled as 1(MATCHED).
8. For each matched point get the neighbouring pixels within the radius defined.
9. Take the first pixel within the radius which has UNKNOWN label. This will be the pixel to match.
10. Calculate the candidate pixels related to the pixel to match. These pixels are those with disparity gradient lower than 1 taking as reference the current *LeftMatchedPoint*.
11. Compute the correlation coefficients between the pixel to match and each candidate pixel using *corr2* function in Matlab.
12. Detect peaks between the correlation coefficients using *findpeaks* function in Matlab.
13. If there is one peak higher than 0.7, label this pixel in the list of pixels as MATCHED and add it to *LeftMatchedPoints* and *RightMatchedPoints*
14. If there is no peaks, label this pixel in the list of pixels as NOMATCH and continue

15. Take the next pixel within the radius of the current *LeftMatchedPoint* and return to step 10.
16. When all pixels within the radius has been processed, return to the step 8.

The step 10 involves calculate the candidate pixels along the epipolar line. In order to not calculate the disparity gradient for each pixels in the line, the following procedure is applied:

The disparity gradient is defined as:

$$DG = \frac{|x_l - x_r|}{\sqrt{\frac{1}{4}(x_l + x_r)^2 + \left(a_y - b_y\right)^2}} \qquad (1)$$

If $DG \leq K$ is desired then,

$$|x_l - x_r| \leq K\left[\sqrt{\frac{1}{4}(x_l + x_r)^2 + (a_y - b_y)^2}\right] \qquad (2)$$

So, the candidate pixels are those in the range of a_{xr} which must be between:

$$a_{xr} \leq (x_l + b_{xr}) \pm \sqrt{\frac{K\left(a_y - b_y\right)^2}{1 - \frac{K^2}{4}}} \qquad (3)$$

Results

The first step is to extract the SURF features from the images, the features obtained for the second level in the pyramid are shown in Figure 1 while those for the first level in the pyramid are shown in Figure 2.

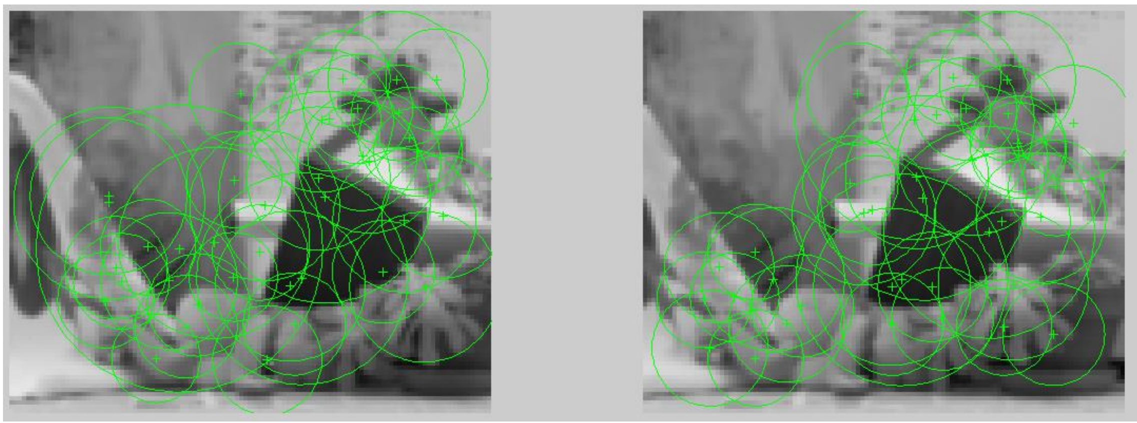


Figure 1 SURF Features found for second pyramid

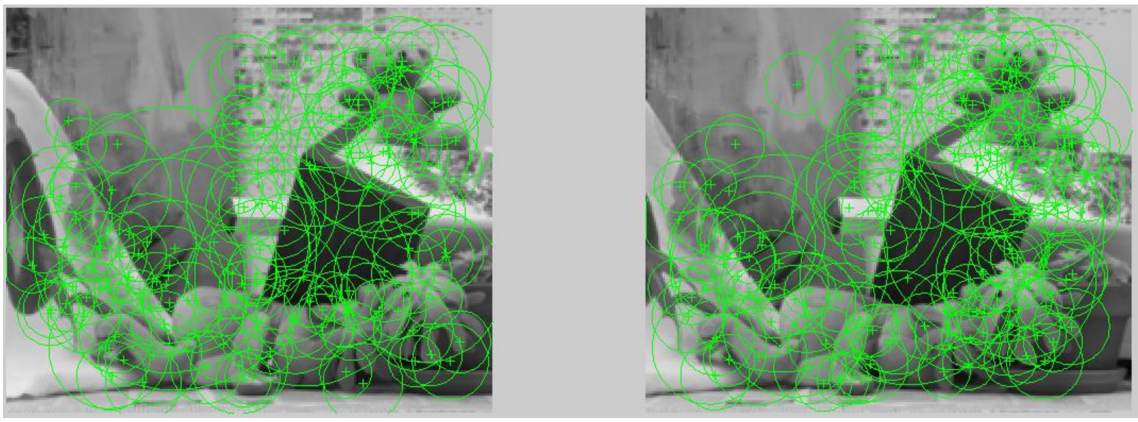


Figure 2 SURF Features found for first pyramid

In the second level 38 features were found while in the first level 130 were found. This indicates that second level features are stronger than those in the first level. Then the matched is performed for each pyramid, the results for the second and first level are shown in Figure 3.

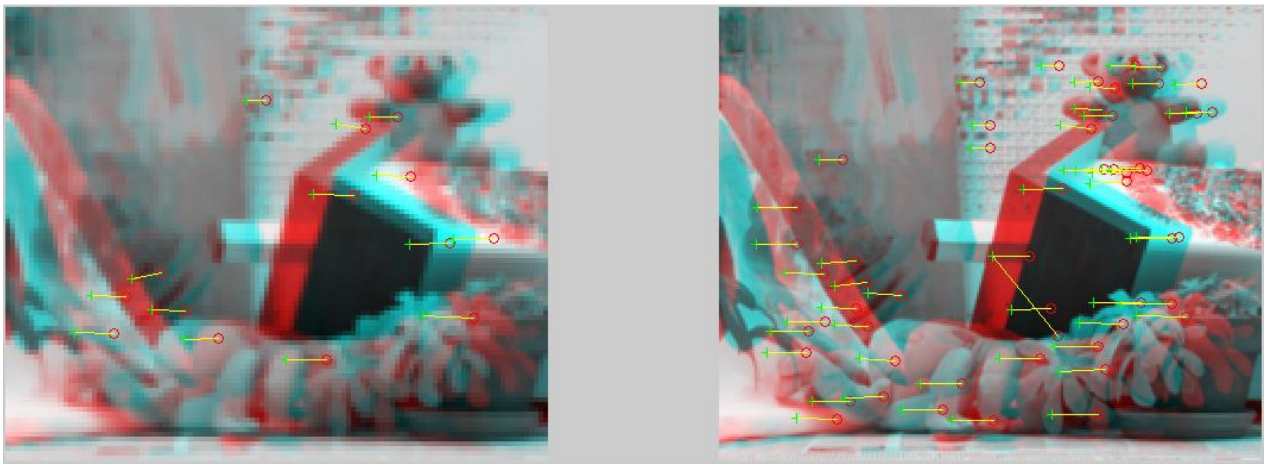


Figure 3 Left image shows matches for second pyramid, right image shows matches for first pyramid

Figure 3 shows that some matches does not correspond to the epipolar line, due to it is know that images are aligned, these matches are eliminated. The results are shown in Figure 4.

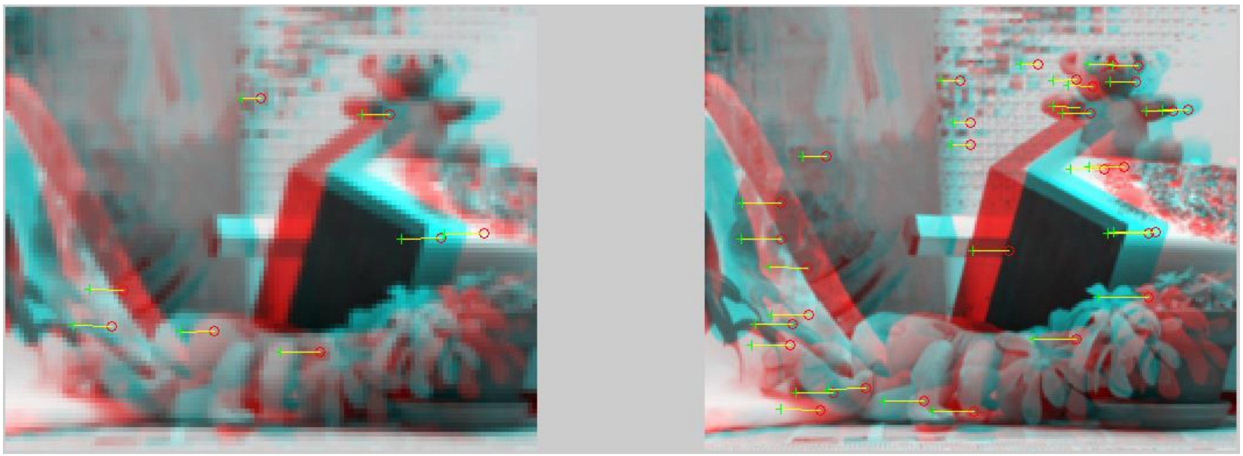


Figure 4 resulting matches after eliminate incoherencies

The resulting matched points are 8 for the second level and 32 for the first level. This points will be the seeds for starting the matching process in the rest of the pixels.

Then, the first left matched point (340, 139) will be taken, the first pixel (using a ratio 10) in the square is at [-10,-10], i.e. (330, 129). Applying (3) the candidate pixels are in (289:313, 129).

Now the correlation coefficients are calculated for each candidate pixel, the graph between correlation and pixel is shown in Figure 5.

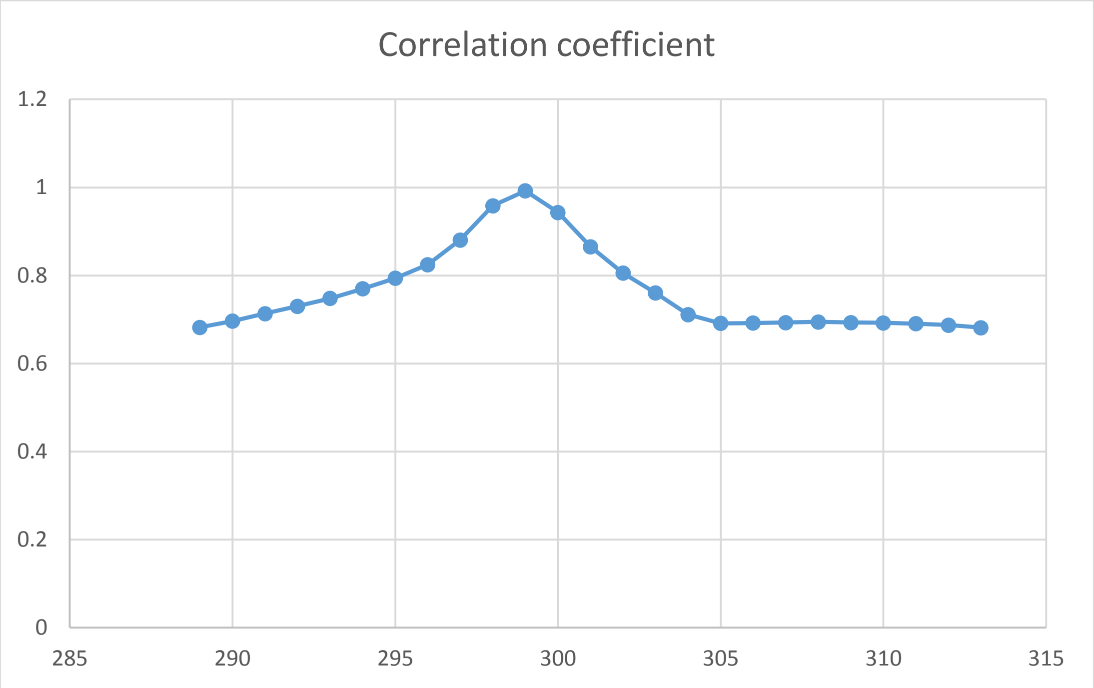


Figure 5 Graph of correlation coefficients

In order to match the pixel, there must be only one peak and it must be above 0.7. From Figure 5 it is clear that there is only one peak and the pixel (330, 129) in the left image will be matched to (299, 129) in the second image.

There are some cases where there are more than one peak as shown in Figure 6 with the pixel (345, 129) and candidate pixels in (304:328, 129). In this case, the pixel is not matched and will be processed again in the next iteration due to there is no enough information yet.

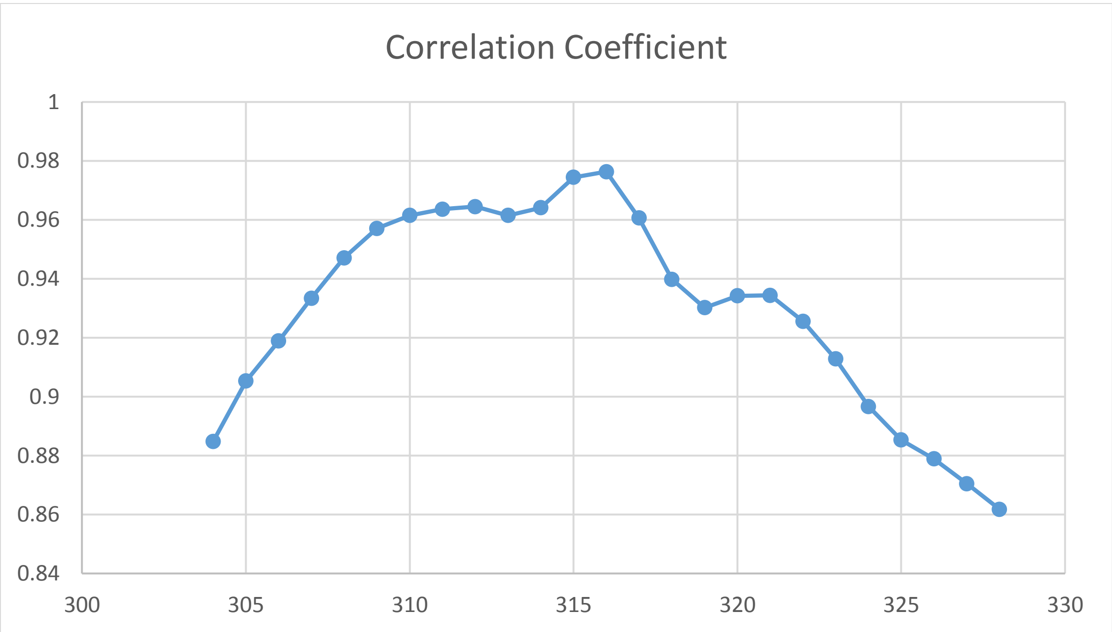


Figure 6 more than one peak in correlation coefficients

On the other hand, if there is no peaks in the correlation coefficients, it means that there is little likelihood of being matched later and the pixel will be labelled as NOMATCH as shown in Figure 7 with pixel (330, 141) and candidate pixels (304:328, 129).

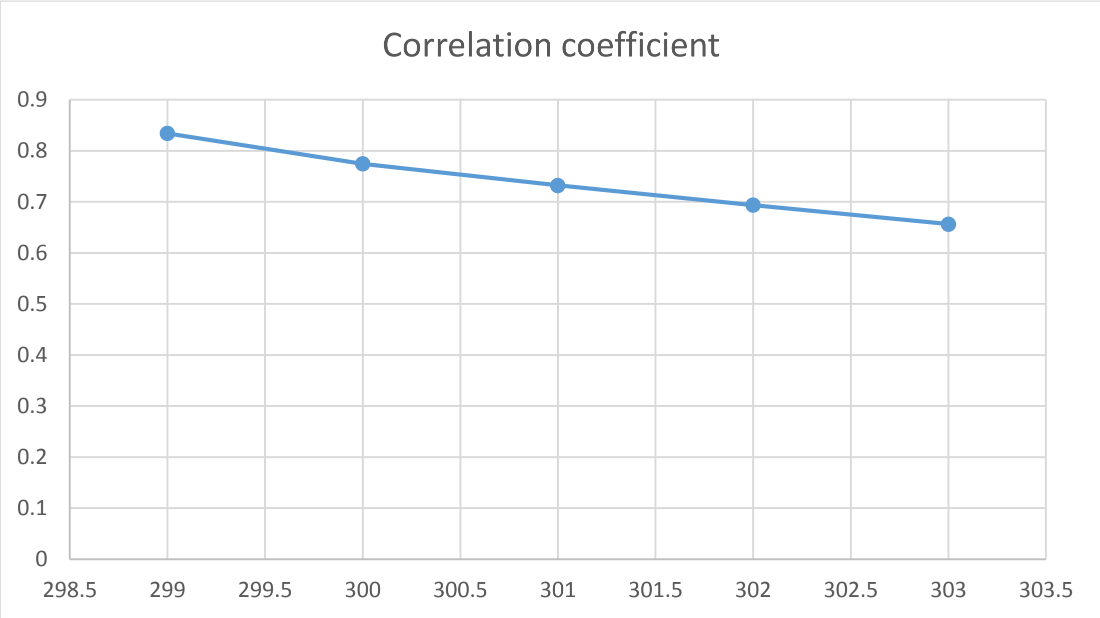


Figure 7 No peaks found in correlation coefficients

This process is repeated iteratively until no new matches are found, the results for "teddy" and "camera" images show that there is a relation between the radius and the amount of matched points as well as the time required to process as shown in Table 1 for the “teddy” images and Table 2 for the “camera” images.

Radius	MATCHED	NOMATCH	UNKNWON	Time(seconds)
4	133,333	22,414	13,003	1,573.722
10	141,064	25,383	2,303	8,820.66
16	142,949	25,408	393	31,384.96

Figure 8 Number of pixels matched and time taken for Teddy picture

Radius	MATCHED	NOMATCH	UNKNWON	Time(seconds)
4	80,259	23,971	6,362	1,137.747
10	85,094	25,075	423	9,773.721
16	86,886	23,706	0	30,524.97

Figure 9 Number of pixels matched and time taken for Camera picture

The disparity maps obtained for “teddy” left image are shown in Figure 10 as well as the ground truth provided by Middlebury dataset.

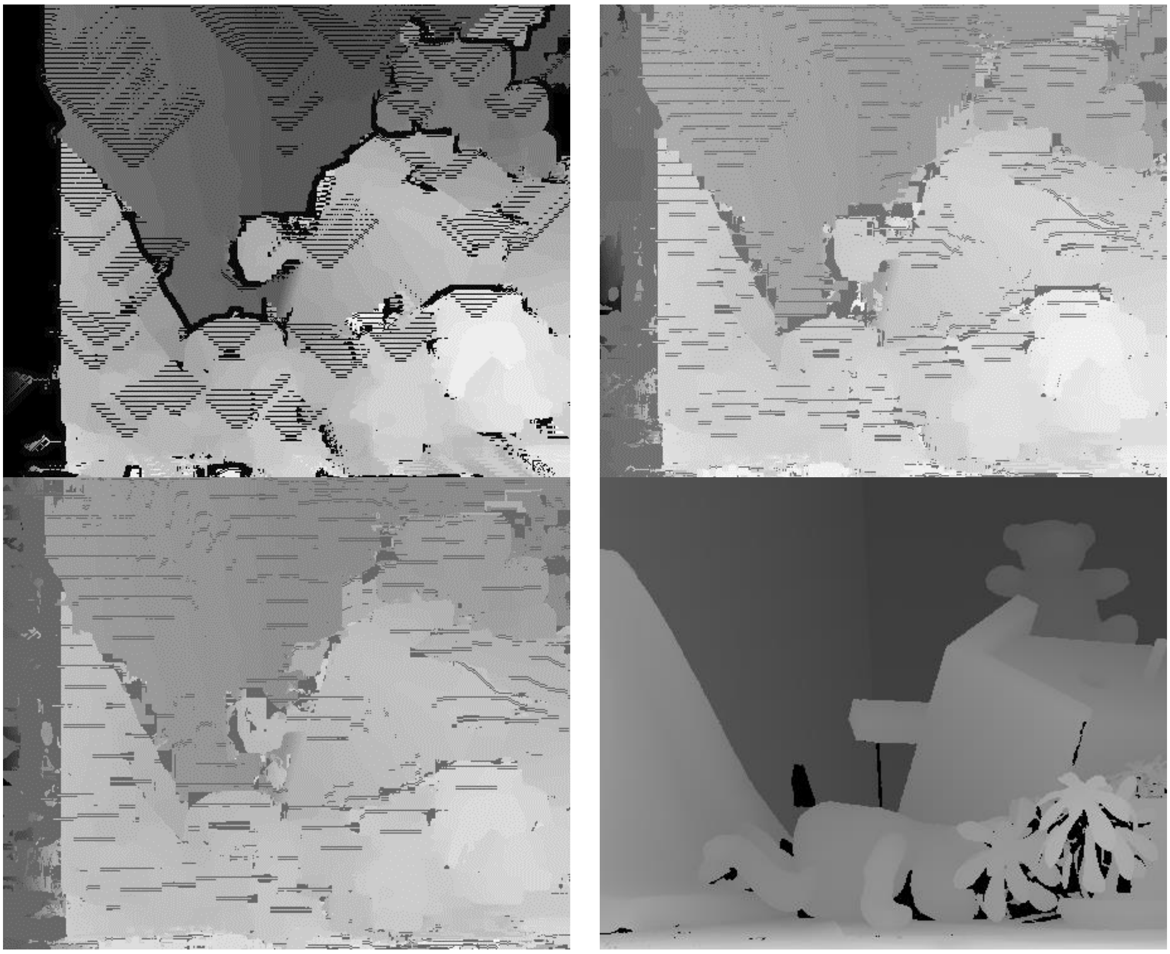


Figure 10 Disparity maps for teddy, top left rate 4, top right rate 10, bottom left rate 16, bottom right ground truth

The disparity map shows a decent estimate when the radius becomes bigger, however the shapes become wider, thus the algorithm was tested using a radius of 2, the time taken was of 928.227 seconds and results are shown in Figure 11.

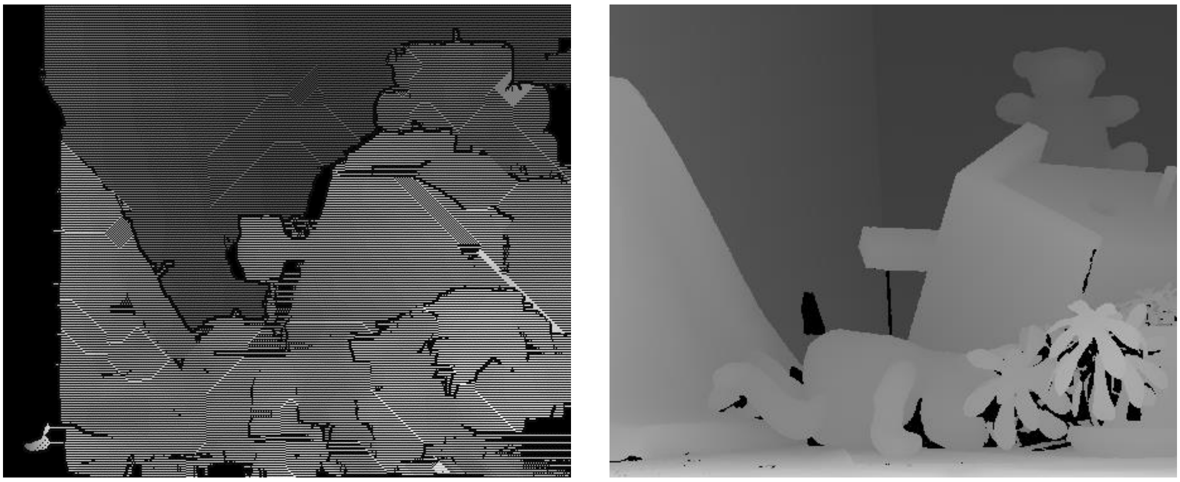
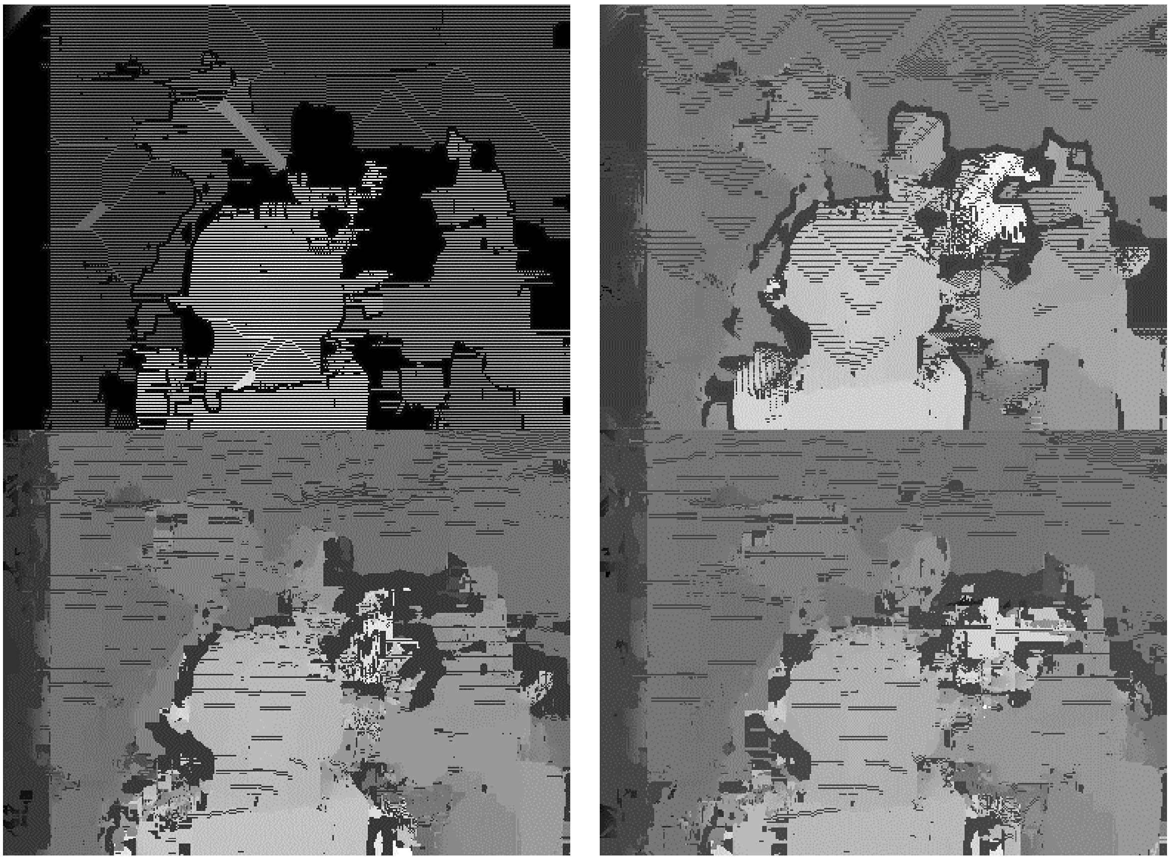


Figure 11 Results obtained with radius 2

The shapes are clearer and the disparity is a good estimate. However, there are some black lines in the edges, probably due to the small radius, some pixels does not find enough certainty in the correlation coefficients when there is a sharp change of disparity.

In the "camera" image with radius 2 the time taken was of 454.593 seconds and results are shown in Figure 12.



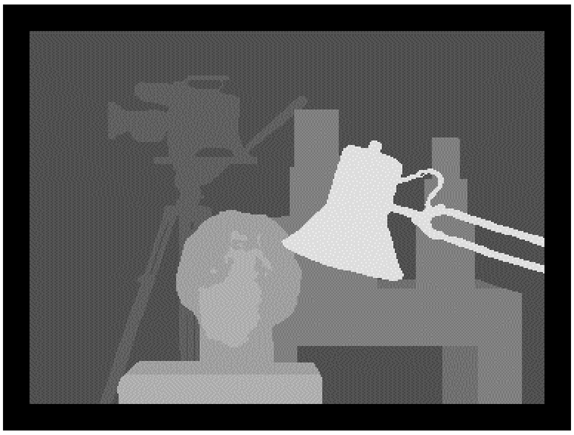
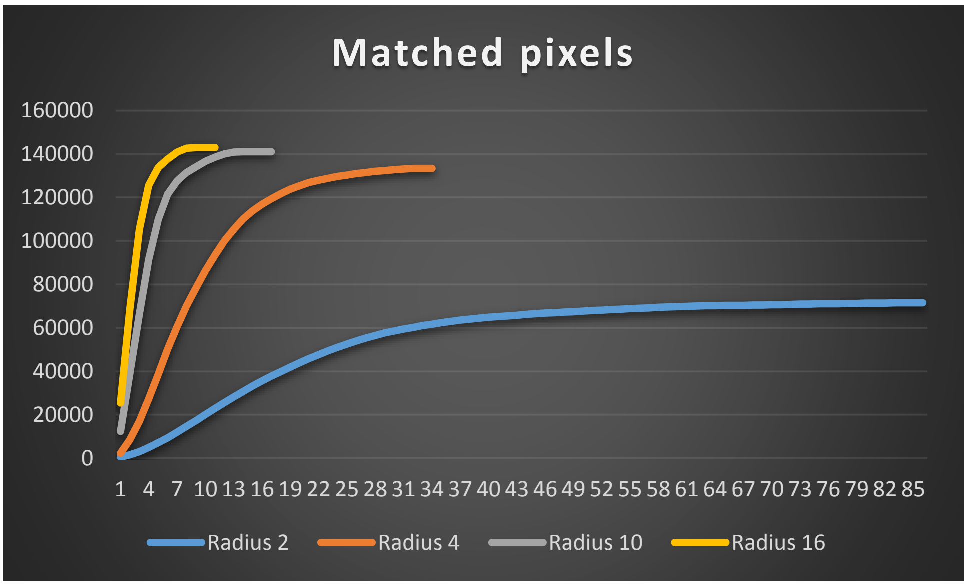


Figure 12 Image results for "camera", top left: radius 2, top right: radius 4, middle left: radius 10, middle right: radius 16, bottom: ground truth

Finally, there is a relation between the number of matched, unknown and no match pixels in each iteration and radius used. The graph between each iteration and number of pixels found are shown in Figure 13. From this graphs it can be said that as the radius increases, the number of matched pixels so will, on the other hand, as the radius decreases the number of no matched pixels increases.



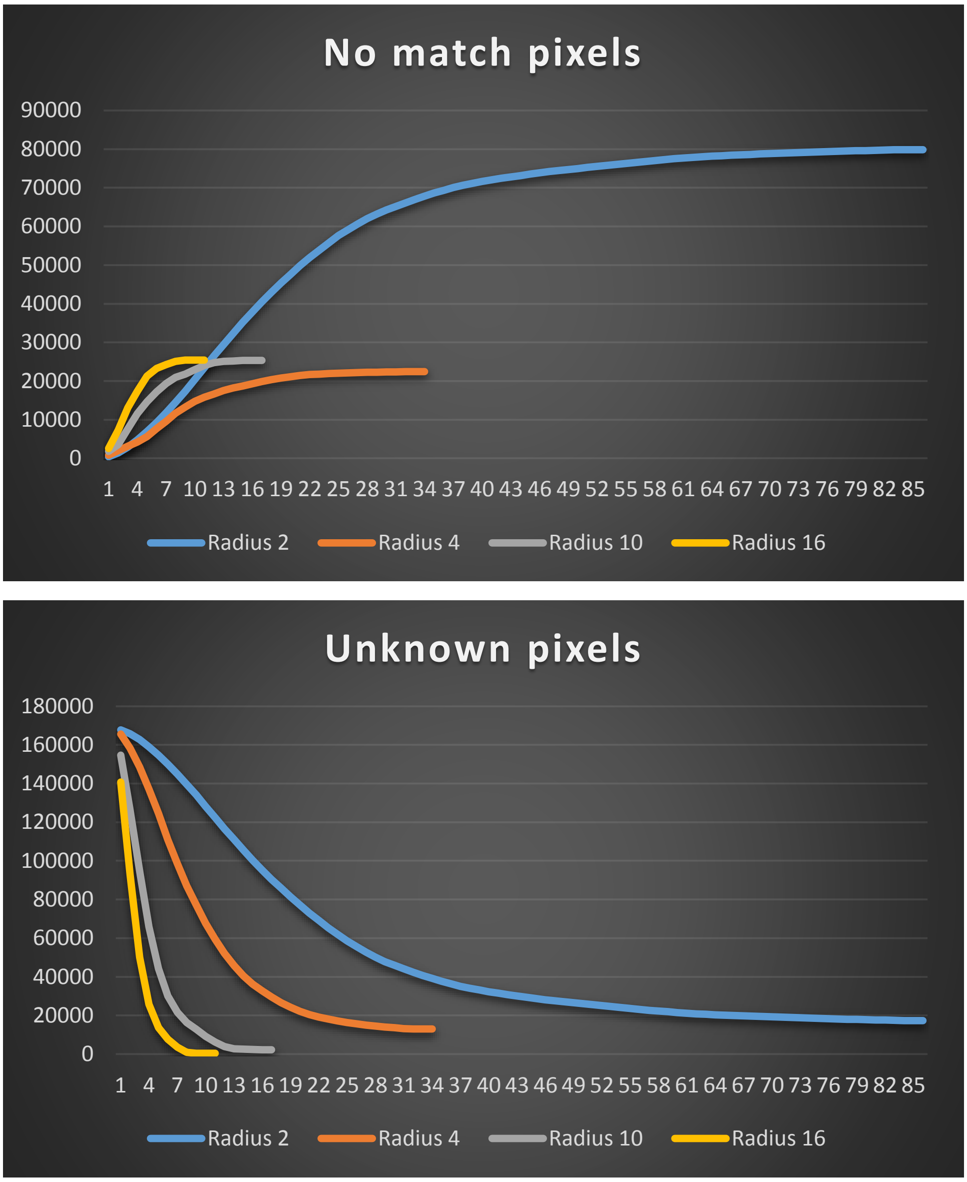


Figure 13 Relation between match, no match and unknown pixels and the radius

Evaluation

According to the results obtained it can be said that the algorithm seems to be an acceptable approach even though the method used does not completely follow the method described in Zhang and Shan’s paper. For instance, the paper made use of statistic methods for calculating the disparity between each pixel and normal distributions of all matched points, and it multiplies them with the correlation coefficients, however, for this project I only used the correlation coefficient for labelling pixels, based on that the candidate pixels already satisfies the disparity gradient constrain and we only have to find the maximum correlation.

As strengths of the algorithm I would note that Zhang and Shan's algorithm uses only one every four pixels instead of each pixels due to memory limitation, in my case, the algorithm takes every pixel and memory used (8 GB RAM) never exceeded 40%.

In contrast, one weakness of the algorithm is the time taken for processing although the original paper does not mention any time for processing, this is bigger than expected. On the other hand, even though the disparities are similar to the ground truth, the shapes in the disparity map are not clear, even more, we can find a lot of noisy horizontal lines through the image.

Prior to the results, I expected to get a disparity map with the objects well defined because the algorithm starts with strong SURF features and maybe the background or non-textured areas could be either noisy or unclear. However, the results show that the objects are wider as the radius is increased, but when the radius is decreased we find some no matched points around the edges which are shown as black lines around the objects. Another non expected result are those horizontal lines in the image, these lines are more evident as the radius is decreased and sometimes we find triangles in the image as shown in Figure 14 which was taken from "Teddy" image with radius 4. When the radius is 2 we find horizontal black lines in odd rows. I presume this is because the radius is so small that not all pixels are evaluated and because the lowest peak allowed is 0.7, the correlation is not enough to be labelled.

Figure 14Triangles formed in resulting images

As improvements it could be added parallel processes to reduce the processing time or use dynamic programming to avoid processing a pixel more than once in each iteration. Furthermore, currently the minimum height of the peaks is 0.7 which could be reduced in order to get more reliable results but the time of processing could increase due to more unknown pixels would be found.

In addition, when the SURF matched features for each pyramid are mixed in step 6, the features in the second pyramid are concatenated to those in the first pyramid, but if we did the opposite (i.e. put the features in the second pyramid first and then those in the first pyramid) would process the strongest features first obtaining more reliable results.

Finally, the main objective of looking for the peaks in the correlation coefficients of the candidate pixels is taken from a Bayesian decision in the original algorithm, in my case as I am only taking the correlation to take a decision some changes could be made, for instance I found that sometimes the correlation graph has a decrease curve starting with values between 0.9 and 1. In these cases the pixel may be matched with the first candidate pixel causing more pixels are matched and those horizontal lines could disappear.

References

Zhengyou Zhang and Ying Shan. *A Progressive Scheme for Stereo Matching*. 2000.

D. Scharstein and R. Szeliski. *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*.
International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002.

D. Scharstein and R. Szeliski. *High-accuracy stereo depth maps using structured light*.
In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), volume 1, pages 195-202, Madison, WI, June 2003.