International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Determination of Target Number of Peaks for Automatic Peak Detection Threshold Determination

Gideon Kanji Damaryam

Department of Computer Science, Federal University, Lokoja, Kogi State, Nigeria

Abstract: ***[1] describes a process to automatically detect peaks from the accumulator array of transform space after points from an image have been transformed using the straight line Hough transform. The process works by specifying a target for the number of peaks to be returned from the transform of each image, rather than specify a threshold for detection of peaks as is traditionally done. This was necessary because no threshold was found that efficiently yielded thehighest number of peaks that could be found, which matched lines manually judged to be navigationally important lines in the images. This paper presents how that target number of peaks was determined. A few typical images were processed to the point of determining gradients and intercepts of important straight lines from peaks found, and specifying their end-points, hence fully specifying the important lines, using various target numbers of peaks. Navigationally important lines in the images were also manually chosen. Optimal target numbers of peaks were chosen as the ones which matched the highest number of important lines manually chosen, while minimising unnecessary processing of peaks and use or storage space.***

Keywords: automatic threshold determination, Hough transform, peak detection

1. Introduction

[1] describesa process to detect straight lines from images using the Hough transform as part of a vision system for a self-navigating mobile robot. The Hough transform is used to reduce the search for lines in an original image to a search for points in a new transformed image, where curves representing lines from the original image intersect in the transform image [2]. It works by transforming lines in a given image to points in a new image while accumulating a measure of the likelihood that points in the new image correspond to lines from the original image. This likelihood is informed by the number of points in the original image that would lie on a line if it were drawn from one end of the original image to the other. When the transformation is complete, points in the new image can then be subjected to a predefined threshold so that points that are very likely to correspond to lines from the original image can be selected and the original lines identified by reversing the transform on the selected points. Those points likely to correspond to lines from the original image are called peaks, and their selection is called peak detection. The gradients and y-intercepts of important lines from the original image can be worked out from peaks, and so the lines are said to have been detected, hence the term line detection.

In [1], Peak detection is achieved by a scheme consisting of a number of steps:

1) Determination and application of the most appropriate threshold for the image under consideration
2) Application of the reduced butterfly filter described in[3] to entries above the threshold from step 1 above only
3) Determining which elements of the accumulator array selected from 2 above are local maxima within a 5 x 5 neighbourhood

In step 1 of that scheme, a threshold for detection of peaks is worked out for each image. The reason for this is that variations were found from image to image, and no single threshold was found that could be applied to satisfactorily detect peaks for all images.A variable threshold was determined real-time, from a fixed target number of peaks, T. This paper presents the method used to determine that T.

With Tdetermined, it can be used to find the right threshold to apply on an image by image basis. Application of the threshold, the second part of step 1, as well as steps 2 and 3 of the scheme have been detailed in [1].

[4]has pointed out that the Hough transform can yield false peaks in high proportions. [5] has pointed out that the transform yieldsmultiple peaks corresponding to only a single line to detect. This scheme is a measure taken by this work to minimise the emergence of false and multiple peaks, and their effects which include unnecessary additional processing, and distraction from finding important lines from the image ([1]).

Before the Hough transform is applied to any image as detailed in [1], the image is first pre-processed to reduce it to an optimal size, and convert it to a binary thinned edge-image showing outlines of important regions in the image. More details of the pre-processing scheme have been presented in [6].

In determining T as detailed shortly in this paper, the image is processed to the point of determining end-points of lines as detailed in [7]after their gradients and intercepts have been found. This completes the specification of important lines, and also helps to eliminate false finds.

In determining end-points, [7] tracks points contributing to the evidence that established each peak as a peak, and filters lines derived from peaks based on whether or not they contain at least 1 valid sub-line. A valid sub-line of a line found, is made up of a group of points on the line close enough to each other to be considered part of the same sub-line, and far enough away from any other point along the same line so that the other point cannot be considered to be on the same sub-line. [7] provides more details as to what

"close cnough" and "far enough away" mean for a particular line detection task.

2. Determination of Target Number of Peaks

A study was done with three typical images labelled image1, image2 and image3, to determine a reasonably good target number of peaks, T, for the images captured in the mobile robot vision-based self-navigation system referred to in *1. Introduction*. Navigationally important lines (NILs) were identified manual in each image. Then, each image was processed with T of 100, 150, 200, 250 and 300. Processing is done as summarised in *1. Introduction*, and detailed in [6], [1] and [7]. The T s are then evaluated. A good T would eventually lead to the detection of high numbers of the NILs matching NILs found manually, while keeping processing time and memory use as low as possible. A discussion of results for each image follows.

3. Results and Discussion

Image1 is displayed in Fig.1a and the pre-processed version of it is shown in Fig.1b. In both figures, NILs are circled in red.

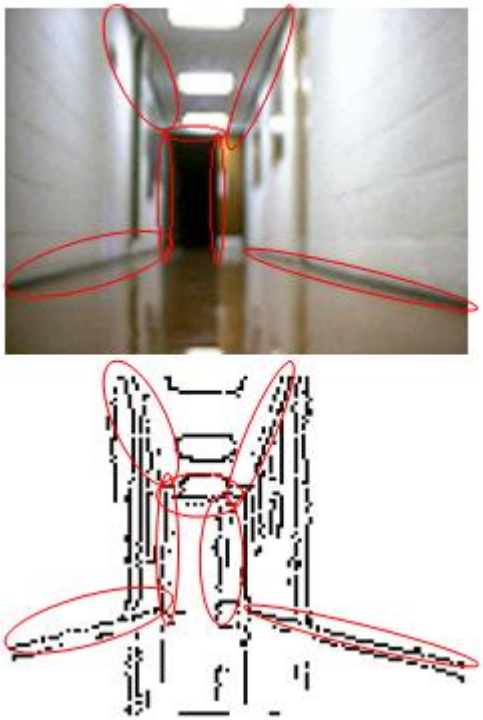

Figure 1: NILs in a typical image – image1
(a) A typical image (b) Thinned version of image1

Fig. 2 shows results with T set at 100. Fig. 2a shows the lines found from peak detection and subsequent application of the butterfly filter and Fig. 2b shows sub-lines found, in colours other than black.

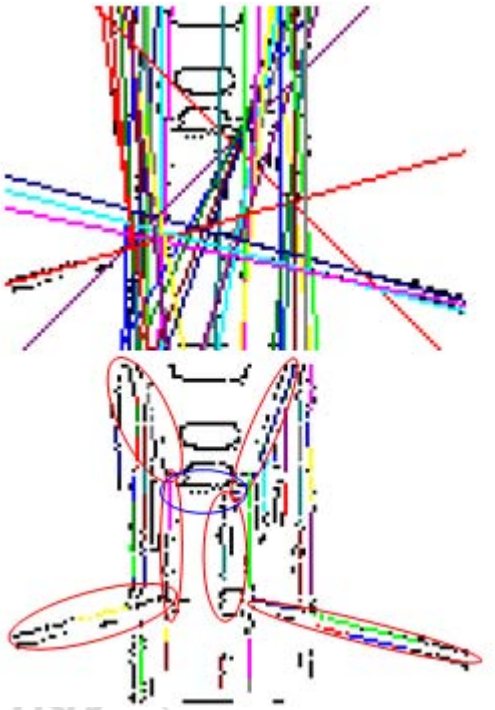

Figure 2: Lines and sub-lines found in image1 with T=100
(a) Lines found(b) Sub-lines found

Figure 3: Lines and sub-lines found in image1 with T=150
(a) Lines found (b) Sub-lines found

Figures 2a and 2b show that with a T of 100, 6 out of 7 NILs were found. Sub-lines found are circled in red. The sub-line corresponding to the top of the door circled in blue was not detected (it is still completely depicted by black pixels).

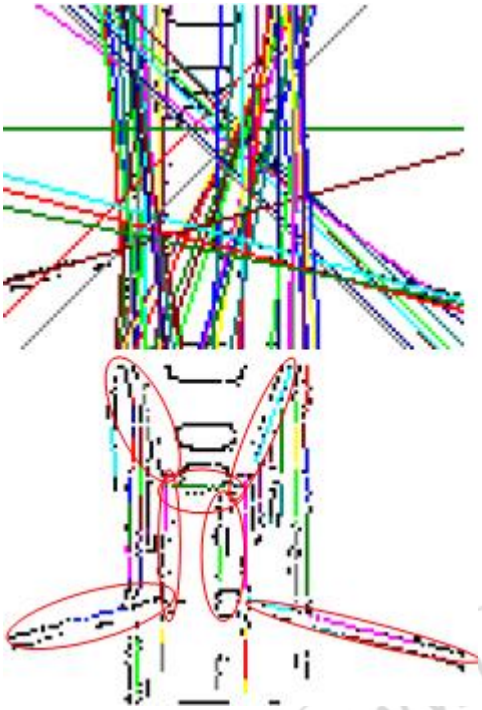

Figure 4: Lines and sub-lines found in image1 with T=200
(a) Lines found (b) Sub-lines found

Figure 5: Lines and sub-lines found in image1 with T=250
(a) Lines found (b) Sub-lines found

Similarly, Figures 3, 4, 5 and 6 show similar results for the cases with T set at 150, 200, 250 and 300 respectively. Table 1 shows further details for each of the images.

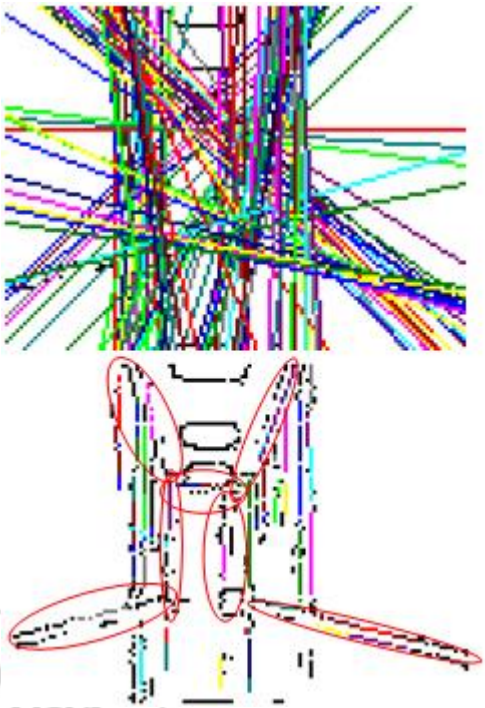

Figure 6: Lines and sub-lines found in image1 with T=300
(a) Lines found (b) Sub-lines found

Table 1: Processes information for image1 with various targets for number of edges

Target Number of Peaks	*100*	*150*	*200*	*250*	*300*
Hough Transform time (ms)	94	109	109	109	187
Peak Detection Time (ms)	31	31	31	31	47
Number of Peaks found	121	182	202	289	364
BF Application Time (ms)	16	15	16	16	15
Number of Peaks after Butterfly filtering	29	30	37	49	59
Number of lines with valid sub-lines	23	24	26	29	33
Number of sub-lines found	43	40	46	49	54
Time to find sub-lines (ms)	15	15	0	16	32
Number of NILs found	6 of 7	6 of 7	7 of 7	7 of 7	7 of 7
Total processing time recorded	141	155	156	156	249

The same number of NILs was found when T was increased to 150 as illustrated in figures 3a and 3b.

When T was set at 200, 250 and 300, all 7 out of 7 NILs were found as shown in Figures 4, 5 and 6.

From table 1, the number of peaks found with T =250, 289, is higher than the 202 found using T =200. Also, the number of peaks after butterfly filtering, and the number of actual sub-lines found is higher in the case for T = 250. T = 250 will therefore take up more memory than T = 200. This means there is an increase in storage requirement for T = 250, and no corresponding increase in useful information as far as the vision-for-navigation system is concerned. This also applies for the increase from T = 250 to T = 300.

Although there were variations to the times taken for peak detection and finding sub-lines, all the variations up to the highest variation of 16ms for both are not significant as

established in [8] (The same method was used to record time in [8] as the work described in this paper). There is in fact an unexpected time taken to find sub-lines of 0 for T = 200. It is safest to conclude that there is no significant conclusion to be drawn regarding processing times. This also applies to the results for images 2 and 3.So from the results with image1, 200 wouldbe the best value for T of the 4 values considered.

For image2, there is no improvement on the navigationally useful information beyond T = 250 as figures 7 to 12 show.There is, however, a considerable increase to the number of lines found after application of the butterfly filter, and the actual sub-lines found from T = 250 to T = 300. This means more memory is used when T is increased to T =300.It may be concluded therefore, that for the purpose of this work, 250 is the best value of T for image2.

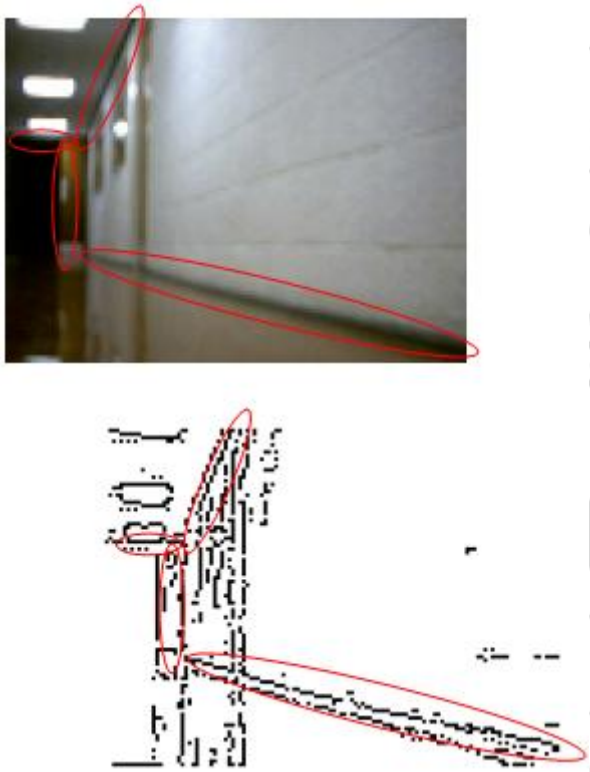

Figure 7: NILs in a typical image – image2
(a) A typical image (b) Thinned version of image2

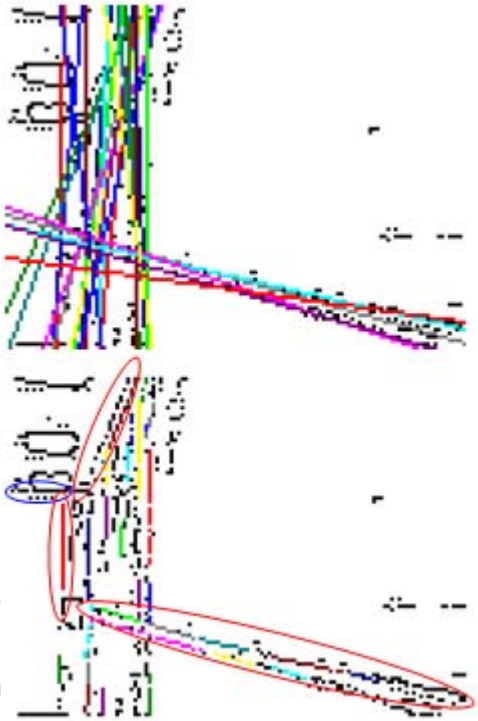

Figure 8: Lines and sub-lines found in image2 with T=100
(a) Lines found (b) Sub-lines found

Figure 9: Lines and sub-lines found in image2 with T=150
(a) Lines found (b) Sub-lines found

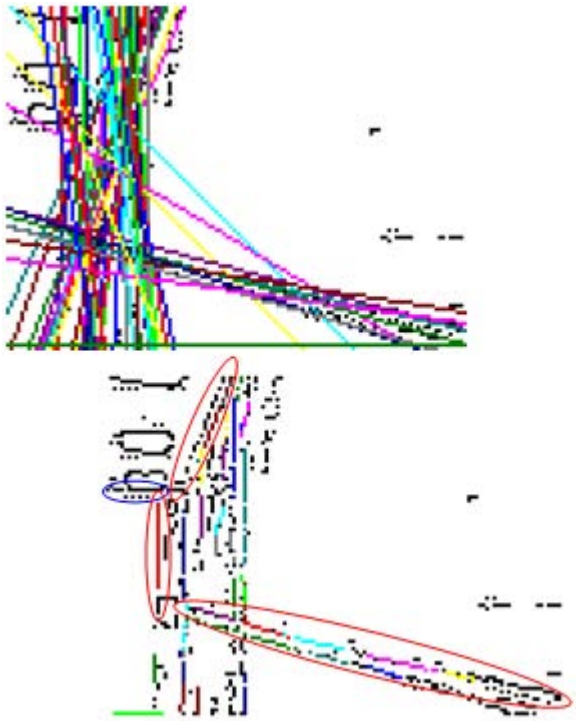

Figure 10: Lines and sub-lines found in image2 with T=200
(a) Lines found (b) Sub-lines found

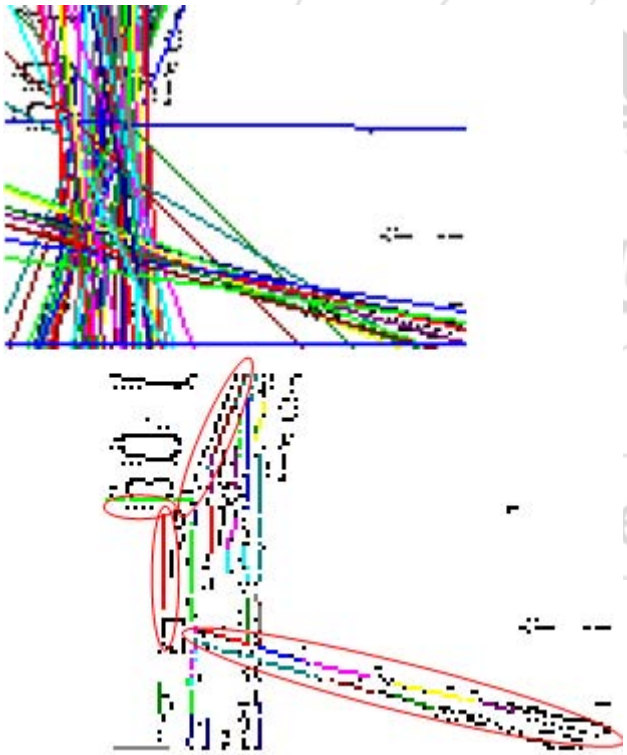

Figure 11: Lines and sub-lines found in image2 with T=250
(a) Lines found (b) Sub-lines found

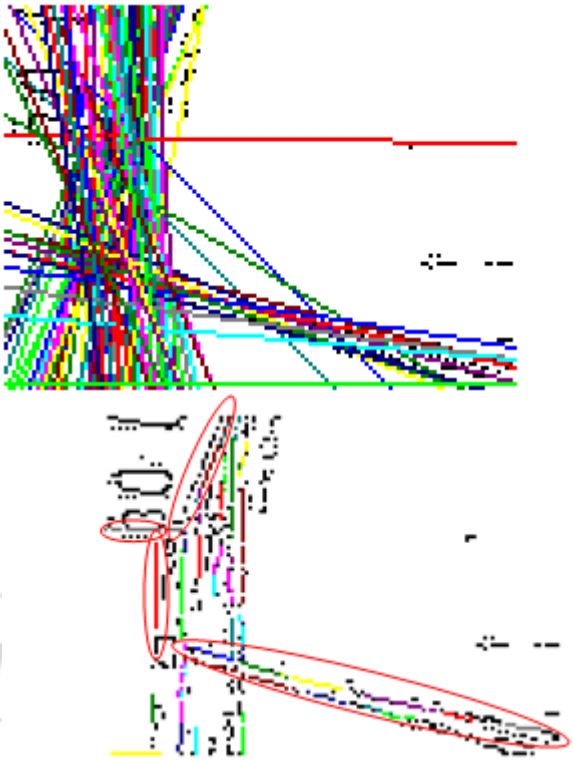

Figure 12: Lines and sub-lines found in image2 with T=300
(a) Lines found (b) Sub-lines found

Table 2: Processes information for image2 with various targets for number of edges

Target number of Peaks	**100**	**150**	**200**	**250**	**300**
Hough Transform time (ms)	94	78	94	94	94
Peak Detection Time (ms)	31	31	31	31	31
Number of Peaks found	104	158	220	262	332
BF Application Time (ms)	16	16	16	160	32
Number of Peaks after Butterfly filtering	21	29	33	37	45
Number of lines with valid sub-lines	21	28	28	29	31
Number of sub-lines found	36	43	43	44	47
Time to find sub-lines (ms)	15	16	16	0	16
Number of NILs found	3 of 4	3 of 4	3 of 4	4 of 4	4 of 4
Total processing time recorded	141	125	141	141	157

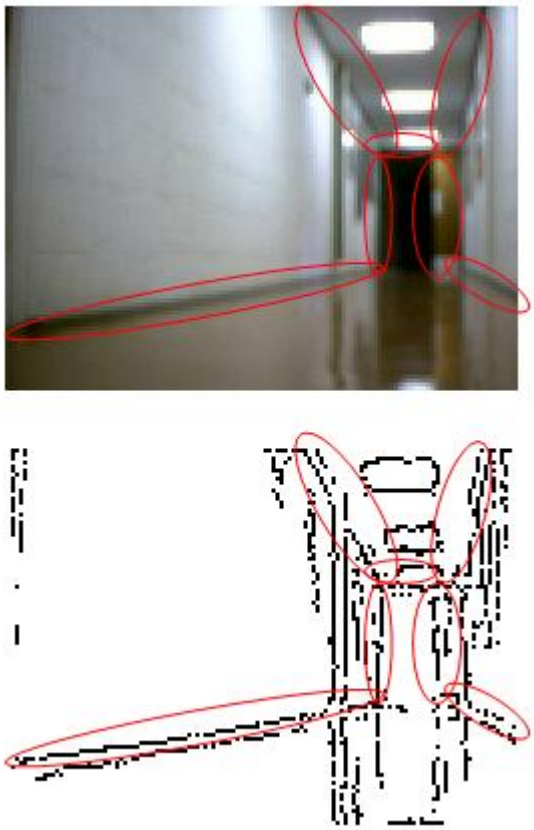

Figure 13:NILs in a typical image – image3
(a) A typical image (b) Thinned version of image3

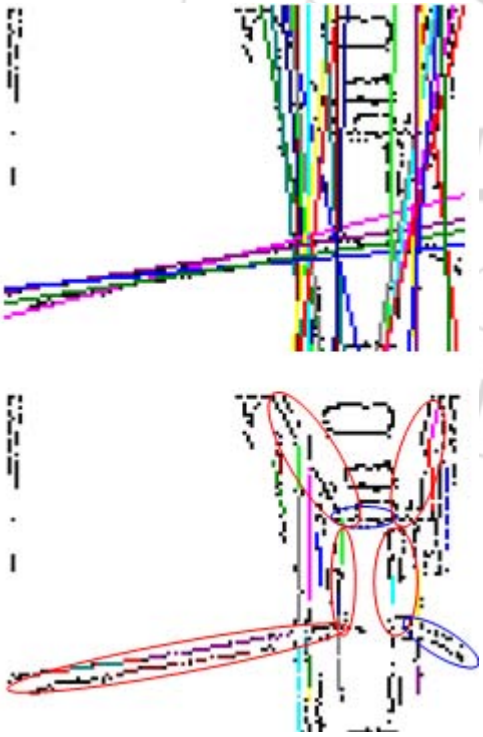

Figure 14: Lines and sub-lines found in image3 with T=100
(a) Lines found (b) Sub-lines found

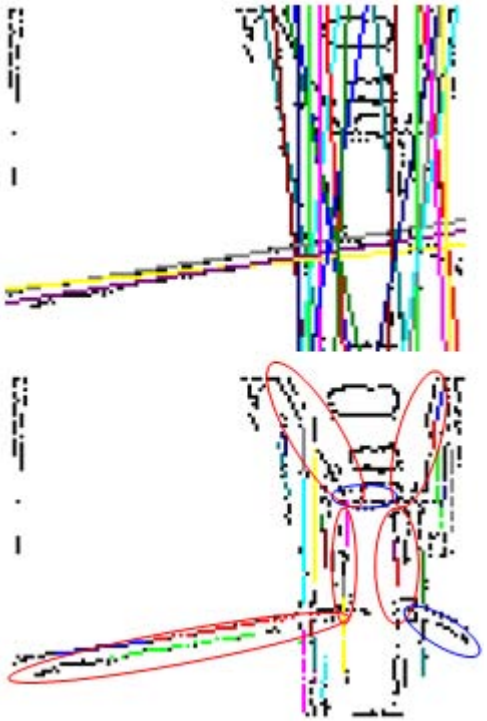

Figure 15: Lines and sub-lines found in image3 with T=150
(a) Lines found (b) Sub-lines found

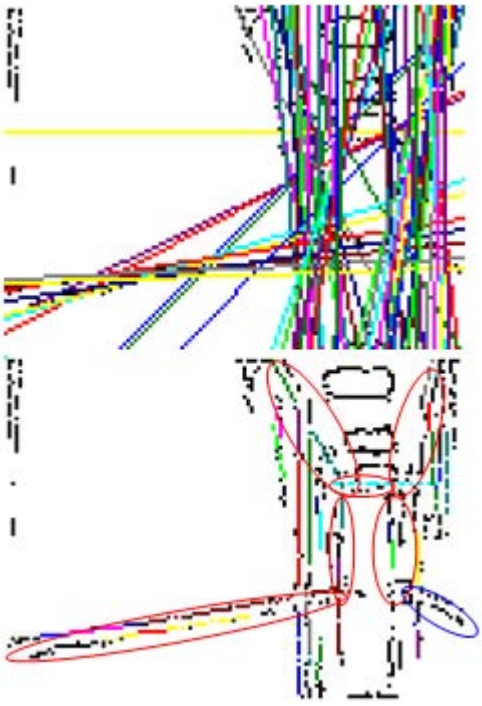

Figure 16: Lines and sub-lines found in image3 with T=200
(a) Lines found (b) Sub-lines found

For image3, results are shown in figures 13 to 18. Again there was no improvement beyond 200, even though there are considerably more peaks and sub-lines as the values of T increased as table 3 shows. For this image, 200 is the best value to choose.

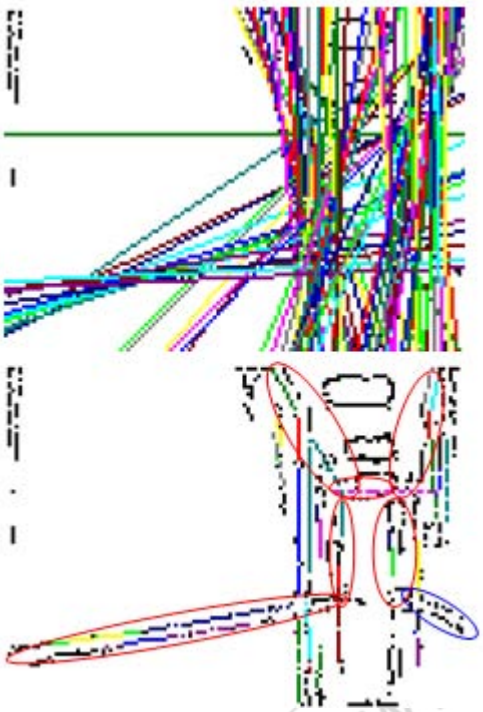

Figure 17: Lines and sub-lines found in image3 with T=250
(a) Lines (b) Sub-lines found

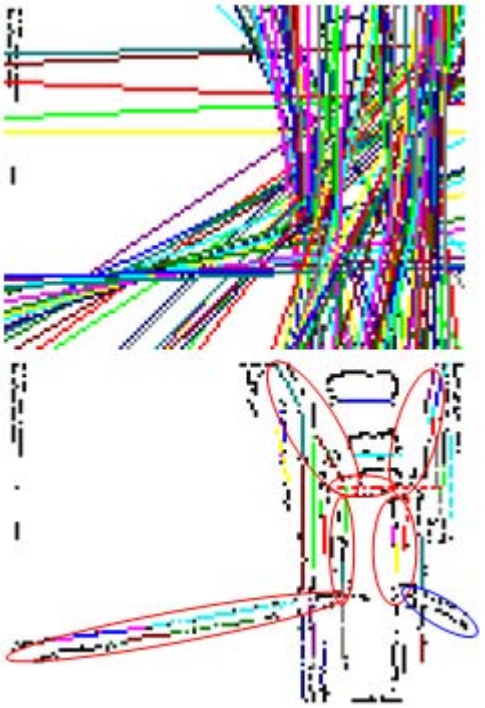

Figure 18: Lines and sub-lines found in image3 with T=300
(a) Lines found (b) Sub-lines found

Table 3: Processes information for image3 with various targets for number of edges

Target number of Peaks	**100**	**150**	**200**	**250**	**300**
Hough Transform time (ms)	94	109	110	109	187
Peak Detection Time (ms)	47	15	310	310	47
Number of Peaks found	101	163	239	289	299
BF Application Time (ms)	16	15	16	16	15
Number of Peaks after Butterfly filtering	18	25	38	49	49
Number of lines with valid sub-lines	17	23	29	29	30
Number of sub-lines found	26	32	40	49	40
Time to find sub-lines (ms)	0	15	0	16	0
Number of NILs found	5 of 7	5 of 7	6 of 7	6 of 7	6 of 7
Total processing time recorded	157	154	157	172	249

4. Conclusion

The following observations can be made from the results recorded:

1) The higher the target number of peaks, T is, the higher the number of important lines, NILs found. Every time T was increased, at least as many NILs were found as were found when a lower value of T was found.
2) The lowest T for which all NILs that could be found were found for the images under investigation is 250
3) There is no significant, consistent increase in processing time taken from the peak detection stage and beyond which would be the stages affected by application of thresholds based on T provided.
4) There is consistent increase in number of peaks foundas T is increased. For the images under investigation, there is on average about 60 additional peaks for every 50 increase in T. Each peak is a data structure of about 37 integers and 3 floating point numbers on average. For a system where an integer is 2 bytes and a float is 4 bytes, this would be about 86 bytes per peak. So an increase in T of about 60 peaks takes about 86 x 60, or about 5k bytes of storage more.
5) There is a general but not always consistent increase in number of peaks after butterfly filter, number of lines with valid sub-sublines and number of valid sub-sublines found. The increase for these is not as significant as the increase for peaks, but would also result in an increase in memory use.

In summary, from studying the results from typical images, it is concluded that 250 is about the best value of T to use as it enabled detection of all the NILs that should reasonable be expected without taking up memory space unnecessarily.

5. Acknowledgement

This paper discusses work that was funded by the School of Engineering of the Robert Gordon University, Aberdeen in the United Kingdom, and was done in their lab using their robot and their building.

References

[1] G. K. Damaryam,"A Hough Transform Implementation for Line Detection for a Mobile Robot Self-Navigation System", International Organisation for Scientific Research – Journal of Computer Engineering, 17(6), 2015
[2] A. O. Djekouneand K. Achour,"Visual Guidance Control based on the Hough Transform", IEEE Intelligent Vehicles Symposium, 2000.
[3] J. F. Boyce, G. A. Jones and V. F. Leavers, An implementation of the Hough transform for line and circle location. Proc, SPIE Inverse Problems in Optics, The Hague, Netherlands, 1987.
[4] W. Grimson, W. L. Eric and D. P. Huttenlocher. "On the Sensitivity of the Hough Transform for Object

Recognition". IEEE Transactions on Pattern Analysis and Machine Vision, 12(3), 1990, 255-274.
[5] V. F. Leavers,"Shape Detection in Computer Vision Using the Hough Transform"(London: Springer-Verlag, 1992).
[6] G. K. Damaryam and H. A. Mani, "A Pre-processing Scheme for Line Detection with the Hough Transform for Mobile Robot Self-Navigation", In Press, International Organisation for Scientific Research – Journal of Computer Engineering, 18(1), 2016
[7] G. K. Damaryam, "A Method to Determine End-Points of Straight Lines Detected using the Hough Transform", International Journal of Engineering Research and Applications, 6(1), 2016
[8] G. K. Damaryam, E Ogbuju and H. A. Mani, "An Investigation of the Costs and Benefits of Thinning on the Straight Line Hough Transform", In Press, International Journal of Advanced Research in Computer and Communications Engineering"