The International Journal Of Engineering And Science (IJES)
|| Volume || 5 || Issue || 4 || Pages || PP -36-44|| 2016 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Hough Transform Consolidation of Back Propagation Network Line Estimates for Self-Navigation for a Mobile Robot

[1]Gideon Kanji Damaryam, [2]Adamu Sulaiman Usman
[1]*Department of Computer Science, Federal University Lokoja, Nigeria*
[2]*Department of Computer Science, Bingham University, Karu, Nigeria*

ABSTRACT
Work has previously been done to fully specify lines within a digital image. Here fully specify means determine the orientation of the line (for example in terms of angle to the vertical), determine the length in pixels from a reference point (for, example the center of the image), and specify the coordinates of the end-points of the line. This can be processor time intensive, and also too accurate for the purpose of vision based navigation, since visions systems typically just need a broad idea of the orientation of a line not its exact angle, and a rough idea of its position from important marker(s) in the image (not its exact distance). A strategy is presented in this paper, to use back-propagation network estimates of portions of different categories (based on orientation) of lines found in sub-images generated by breaking down the original image, as input for the Hough transform to consolidate the estimated lines into the longest lines possible, spanning multiple sub-images if necessary, and taking into account the range of orientations that are valid for sub-lines for each category. This is in an effort to replace parts of an analytical vision-system for a mobile robot, with artificial intelligence in the hope of saving some processing time.

Keywords: *artificial intelligence, artificial neural networks, back propagation networks, Hough transform, line detection, machine vision*

Date of Submission: 07 March 2016 | Date of Accepted: 07 April 2016

I. Introduction

[1] and [2] have presented steps to detect sub-lines from lines detected using the Hough transform from images of indoor rectilinear corridors for the purpose of self-navigation for a mobile robot. The steps from [1] yield the angles and distances of the lines from the center of the images, and [2] has presented steps to determine the endpoints of the lines.

Before steps in [1] are applied to an image, it is pre-processed to a thinned binary image using steps outlined in [3].

Fig.s 1, 2, 3 and 4 illustrate a sample image, a pre-processed version of it, lines detected from it using steps from [1] and sub-lines of those detected using steps from [2] respectively.

Figure 1: Typical captured image

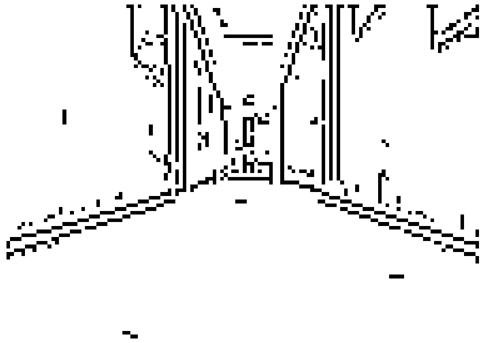


Figure 2: Typical thinned binary image after pre-processing

Figure 3: Lines detected in a typical image

Figure 4: Sub-lines found in a typical image

[4] and [5] have also presented steps to detect sub-lines from similar images by breaking them down into sub-images and detecting categories of lines from them. Table 1 summarisesdescriptions of the categories. The categories used in [4] and [5] are the same ones also used in [6] in detecting corridors and doors from the lines found using [1] and [2], and an estimate for the vanishing point determined using steps presented in [7].

Table 1: Lines Categorisation

Category ID	**Description**		**Minimum Θ**	**Maximum Θ**	**Range Size**
0	Vertical		-5 (or 175)	4	10
1	Vertical Backslash		5	24	20
2	Backlash		25	64	40

3	Horizontal Backslash		65	84	20
4	Horizontal		85	94	10
5	Horizontal slash		95	114	20
6	Slash		115	154	40
7	Vertical Slash		155	174 (or -6)	20

Fig. 5 shows a sample pre-processedimagebroken into sub-images.

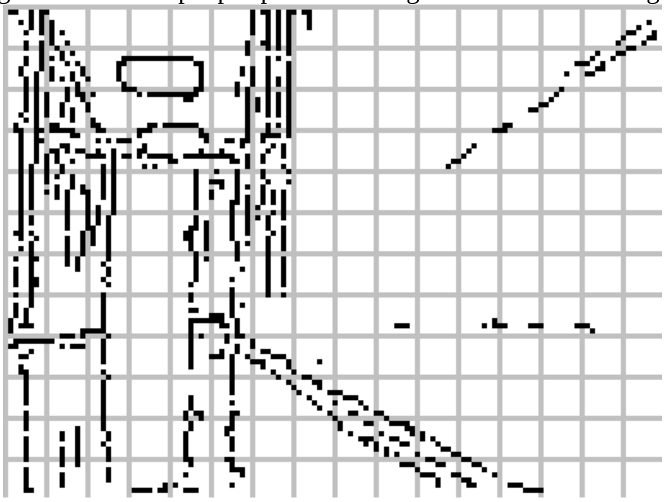


Figure 5: Sample pre-processed image broken into sub-images

Fig. 6 shows a sample result of sub-lines detected for the various category.

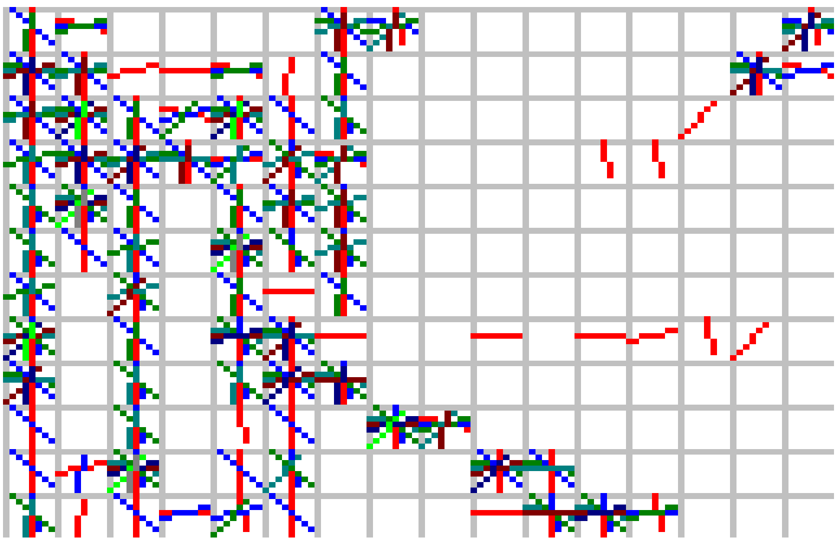


Figure 6: Results from line categories detection for image in fig. 5

Different colours are used to represent the different lines found in each sub-image. The first line found in each sub-image is coloured red, the second one is blue, the third one is green, etc. Table 2shows thecolours showing the orders in which the lines are found:

Table 2: Colours used to indicate different lines found in a sub-image

Line-Find Position	Colour	Sample Line
0	red	
1	blue	
2	green	
3	teal	
4	maroon	
5	navy	
6	lime	
7	dark gray	

It should be noted that the method presented in [4] and [5] cannot specify where in the sub-image a line was found, and all lines are shown centred in the sub-image in which they were found.
It is possible that some of the sub-lines found in sub-images are parts of lines that span beyond the sub-images the lines came from.
This paper presents the use of the Hough transform to stitch together lines from the same category along directions that are valid for the category.
Previous work on vision/navigation systems for mobile robots is well exemplified by the work of [8]. They have developed a method for vision based navigation using a THMR-III outdoor mobile robot. Their method enables the robot which consists of a van with two cameras, two computers and no human beings, to recognise and drive down roads using knowledge integrated into a fuzzy rule base for edge detection. Like the current work, [8] use a combination of Image Processing (edge-detection) and Artificial Intelligence techniques (fuzzy logic) to achieve self-navigation in a mobile robot based on visual information. The current work, looks at a combination of the Hough transform which is an image processing technique (which takes edge-detection a little further, to line detection), and Artificial Neural Networks (BPNs) which is an AI technique to achieve the same primary objective.

II. Combining Sub-Lines found with BPN with the Hough Transform

In this paper, combination of sub-lines from sub-images found using BPNs as presented in [4],are put together into full lines using the Hough transform. Results from line recognition using BPN provide input for the Hough transform. In [4], the result from processing a sub-image using BPN, is stored as an 8 column binary vector. Each of the 8 elements of the vector says whether or not a line in each of the 8 categories of lines was found. There are 192 such vectors for every image corresponding to the 192 sub-images of the image.
The Hough transform is applied to consolidate results for each line category one category at a time. Each sub-image, represented by one of the 8 elements of its results vector, is taken as 1 pixel. As the values of θ are increased from 5° to 185°, for every sub-image, the input value considered for the transform is the element of the results vector corresponding to the category in whose range the current value of θ lies.
For the vertical line category, vertical lines found by processing sub-images of the image in fig. 5using the method of [4] are shown by themselves in fig. 7 below.

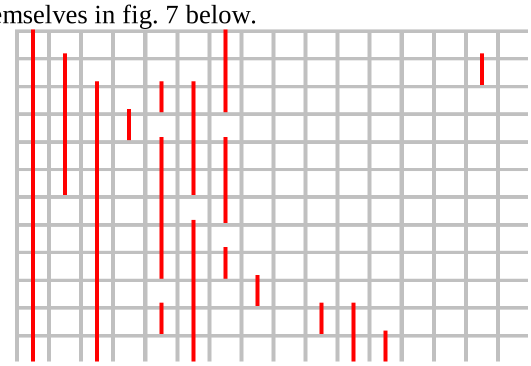


Figure 7: Vertical lines found in sub-images using BPNs

The Hough transform is applied for this category for θ between 175° and 184°. As discussed in Chapter 4, this range covers lines similar to lines in the range -5° to 4°. 175° to 184° is however, easier to deal with for setting up of the accumulator array (with all non-zero values) and processes such as application of the butterfly filter and selection of local maxima.
Significant lines are detected in a very similar way to detection of significant lines described in [1]. First peaks are detected by application of a threshold automatically determined from a target number of peaks found using [9]. This target found in this case is 6. The reduced butterfly filter is then applied. In [1], an idea of applying the filter only to peaks found from application of a threshold, to save processing time, was introduced. That idea is also used here. Local maxima are then selected from within 3 x 3 neighbourhoods in the butterfly filtered accumulator array. Note that this differs slightly from the case for processing Hough transform results from the complete image in [1] where local maxima were selected from within 5 x 5 neighbourhoods. This is because the 'pixels' in the input image in this case are actually 8 x 8 pixel sub-images and for that reason, 'close' results are actually not that close (actual lines could be up to 15 pixels away).
*3.1 Vertical Lines*which follows shortly shows the result for the vertical line category.

Other categories are processed in a similar fashion and results of them are summarized in *3.2 Other Categories Lines*.

III. Results

3.1 Vertical Lines

Lines resulting from all the processes described in section 2 above for the vertical line category are illustrated in fig. 8.

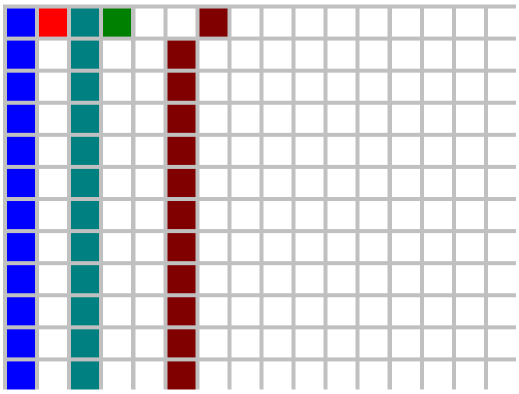


Figure 8: Vertical lines found in the 'full picture'

5 peaks were found in the example, differentiated in the figure by 5 different colours. Note that some of the lines overlap. Values of ρ and θ for the lines are shown in table 3.

Table 3: Parameter values for vertical lines found in the 'full picture'

Line ID	θ	ρ
0	175	8
1	180	8
2	175	6
3	180	6
4	175	3

The Hough transform finds lines but does not say where they begin and end. It also puts lines together which lie along the same infinite line. To determine endpoints of any possible actual lines on these infinite lines, further processing is done to determine which cells actually contributed to each line. This is done similar to what was described in [2]. However, the details of the criteria for lines in this case are different. The criteria for a line with this hybrid approach are:

1. It must be at least 3 pixels long
2. It must be at least 1 pixel away from any other line on the same infinite line

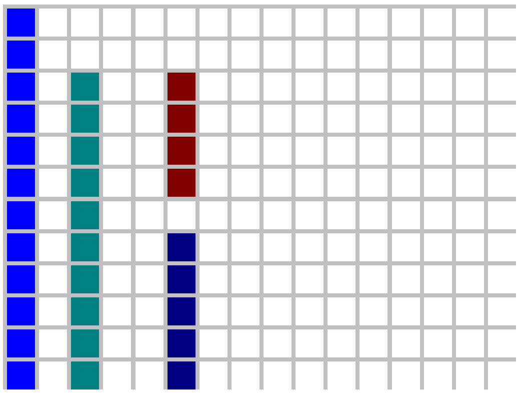


Figure 9: Vertical lines with valid sub-lines found in the 'full picture' with hybrid 5

Results are illustrated in fig. 9. Some lines do not appear due to overlap. Sub-images making up each line are shown in table 4.

Table 4: Sub-Images which made up valid vertical sub-lines

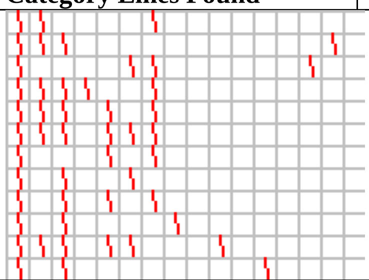
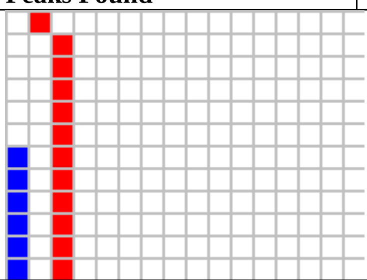
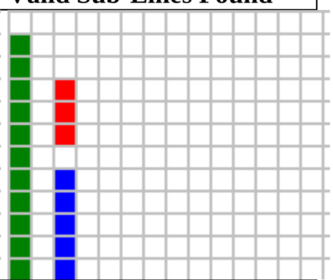
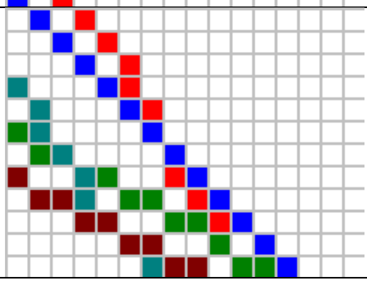
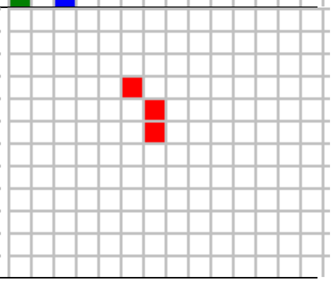
ID for Line with Valid Sub-Lines	Number of Sub-Lines	Number of Sub-Images in Sub-Line	IDs of Sub-Images in Sub-Line
0	1	12	0 16 32 48 64 80 96 112 128 144 160 176
1	1	12	0 16 32 48 64 80 96 112 128 144 160 176
2	1	10	34 50 66 82 98 114 130 146 162 178
3	1	10	34 50 66 82 98 114 130 146 162 178
4	2	4	37 53 69 85
		5	117 133 149 165 181

IDs are assigned to sub-images from left to right, and from top to bottom beginning from 0.

3.2 Other Categories Lines

The results for the other categories were obtained in a similar manner and are summarised in tables 5, 6 and 7.

Table 5: Examples of inputs to Hough Transform, Peaks Found and Sub-Lines Found

Category	Category Lines Found	Peaks Found	Valid Sub-Lines Found
Vertical Backslash			
Backslash			

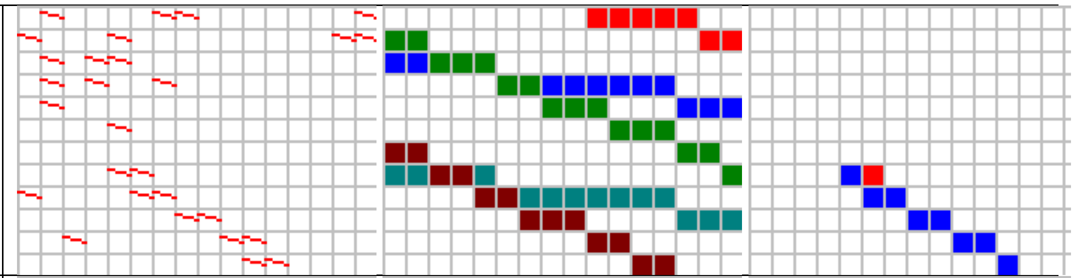
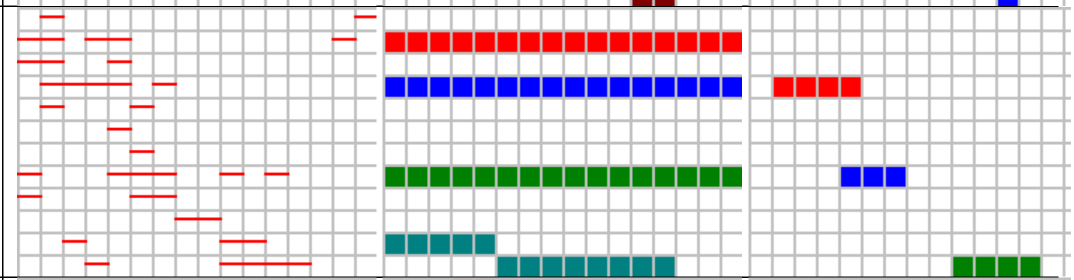
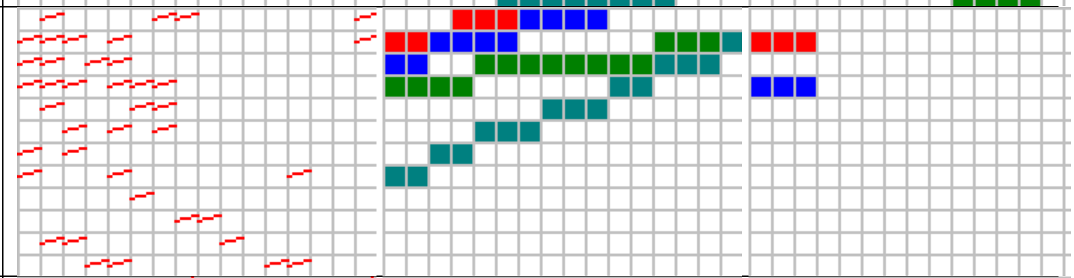
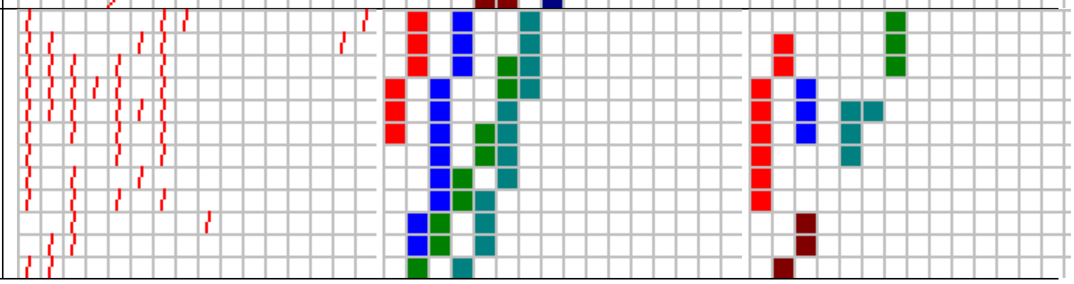
Horizontal Backslash	
Horizontal	
Horizontal Slash	
Slash	
Vertical Slash	

Table 6: Details of Peaks found Hough transform

Category	θ Range for which Hough Transform was Performed		Number of Peaks Found	Parameters of Peaks Found	
	Lower Bound	Upper Bound		θ	ρ
Vertical Backslash	5	24	2	5	-6
				5	-8
Backslash	25	64	5	30	-1
				45	-1
				60	-3
				35	-5
				60	-5
Horizontal Backslash	65	84	5	80	6
				80	3

				70	2
				80	-2
				65	-3
Horizontal	85	94	4	90	5
				90	3
				90	-1
				85	-5
Horizontal Slash	95	114	4	100	6
				105	6
				95	4
				110	2
Slash	115	154		130	6
				130	4
				115	2
				135	-1
				125	-2
				145	-2
Vertical Slash	155	174		170	8
				170	6
				155	4
				165	3

Table 7: Details of Sub-lines

Category	Line ID with Valid Sub-Lines	Number of Sub-Lines	Number of Sub-Images in Sub-Line	IDs of Sub-Images in Sub-Line
Vertical Backslash	0	2	3	50 66 82
			5	114 130 146 162 178
	1	1	11	34 50 66 82 98 114 130 146 162 178
Backslash	0	1	3	53 70 86
Horizontal Backslash	0	1	4	116 117 133 134
	1	1	8	116 133 134 151 152 169 170 187
Horizontal	0	1	4	49 50 51 52
	1	1	3	116 117 118
	2	1	4	185 186 187 188
Horizontal Slash		1	3	16 17 18
		1	3	48 49 50
Slash	0	1	3	15 30 45
	1	1	3	15 30 45
	2	1	3	15 30 45
Vertical Slash	0	1	8	17 33 48 64 80 96 112 128
	1	1	3	50 66 82
	2	3	3	6 22 38
			4	68 69 84 100
			3	146 162 177

3.3 Combined Results

The full results of the Hough transform and subsequent post processing for all the line categories are presented together in fig. 10.

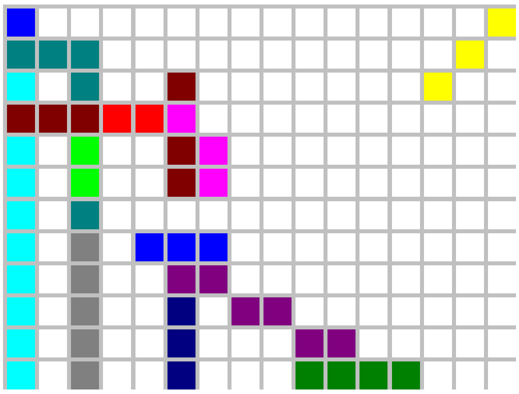


Figure 10: All valid sub-lines found in the 'full picture'

IV. Conclusion

This paper has presented consolidation of sub-lines detected with BPNs generated by a processes described in [4],to detect their full parent lines, by apply the Hough transform on the sub images as though they were pixels, and taking into account the constraints imposed by the orientation categories of the sub-lines.

The detection of lines as presented in this paper is not as exact as using [1] and [2], say, but provides a good estimation. While [1] provides exact angles (correct to the nearest degree) and exact distances (correct to the nearest pixel) from the center of the images, [4] only provides information about which category a line belongs to, where a category spans about 10 degrees, and which sub-image the sub-line came from. A sub-image is 8 x 8 pixels in size. Using an artificial neural network such as in [3] and then stitching them together with the Hough transform along valid directions, however, is potentially faster than using the Hough transform to detect lines from pixels, and then having to go on to detect sub-lines as in [2]. The Hough transform can be very time consuming [6, Dempsey]. Time is important in a real-time system such as a system for control of a moving mobile robot. Moreover, although using [1] gives exact angles and distances, the ultimate use of the lines in determining high-level features in [6] does not require that much accuracy about the orientation of the line or its displacement. It just requires knowledge of the broad category of the sub-lines and their positions relative to the vanishing point. Information about category is obtained from results from [4].

Going the route of [4], however, there is no information about the vanishing point, although there is information about relative position from the center of the image. Estimation of vanishing point for results from this paper will be the subject of a future work.

Future work will also look into generation of navigation instructions for a mobile robot from the results of the work presented in this paper.

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]. 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

[3]. G. K. Damaryam and H. A. Mani, "A Pre-processing Scheme for Line Detection with the Hough Transform for Mobile Robot Self-Navigation", *International Organisation for Scientific Research – Journal of Computer Engineering, 18*(1), 2016

[4]. G. K. Damaryam, Detection of Various Categories of Sub-Lines Using the Back-Propagation Network, In Press, *International Organisation for Scientific Research*.

[5]. G. K. Damaryam, A Back-Propagation Neural Network Detection of Vertical Sub-Lines, International Journal of Engineering and Science, 5(1), pp 22-29, 2016.

[6]. G. K. Damaryam and O. J. Abari, Corridor and Doors Recognition from Detected Lines and Estimated Vanishing Point, *International Journal of Science and Research, 5*(3), pp.204-211, 2016.

[7]. G. K. Damaryam, One Point Perspective Vanishing Point Estimation for Mobile Robot Vision Based Navigation System, *International Journal of Research and Science, 5*(2), pp 930-934, 2016.

[8]. W. Li, X. Jiang and Y. Wang, Recognition for Vision Navigation of an Autonomous Vehicle by Fuzzy Reasoning. *Science Direct Fuzzy Sets and Systems, 93*(3), p275-280, 1998.

[9]. G. K. Damaryam, Determination of Target Number of Peaks for Automatic Peak Detection Threshold Determination, *International Journal of Science and Research, 5*(2), pp 189 -196, 2016.