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

Corridors and Doors Recognition from Detected Lines and Vanishing Point

Gideon Kanji Damaryam[1], Ovye John Abari[2]

[1, 2]Department of Computer Science, Federal University, Lokoja

Abstract: ***This paper presents algorithms that implement part of a vision-based mobile robot self-navigation system for navigation within rectilinear corridors, concerned with recognition of corridors and doors in corridors. The algorithms take information about lines and an estimate for the vanishing point already determined using other algorithms (not detailed in this paper) as input. Algorithms presented use parameters empirically manually determined for the types of images in this specific application, but they can be generalized, with likely modification to the parameters. The algorithms rely on a categorization scheme for lines found to be in the image, and determination of their distance and direction (displacement) from the vanishing point. The categorization scheme and algorithms to determine the displacements were both developed for this work and are first presented. These displacements from the vanishing point, along with the categories of the lines are used in the corridor and door recognition algorithms. Sample results are shown.***

Keywords: corridor recognition, door recognition, feature recognition,high-level feature, image processing, indoor navigation, robot navigation

1. Introduction

This paper presents methods for recognition of corridors and doors in corridors in images captured by a mobile robot within a rectilinear indoor environment.

In preparation for the methods presented, images would have already been pre-processed using a method presented in [1], the gradients and angles of important lines in the images would have been determined using a method presented in [2], endpoints for those important lines have been determined using a method presented in [3] and the vanishing pointsin the images would have been estimated using a method presented in [4].

The algorithms in [1] yield an optimally resized binary version of the original image showing thinned edges within the image. Figure 1 shows a typical image captured by the front-facing camera of the mobile robot used for this work after it is resized to a 128 pixel by 96 pixel size. Figure 2 shows the same image after edges have been detected using the Sobel edge detection operators described in [5] and thinned using a method introduced in [1]. [6] has established that for the vision system developed in this work, thinning yields higher quality line detection while ultimately saving processing time, despite the time taken to do thinning itself.

Figure 1: Typical captured image after resizing

Algorithms presented in [2] take a thinned binary image such as the one in figure 2 and establish parameters indicating the angle of each important line in the image to the vertical, and its distance to the centre of the image which is taken as the origin, i.e., the point (0,0). This is called line detection. [2] uses a process called the straight line Hough transform, along with other additional innovations to facilitate the detection of the most appropriate lines. In particular, [2] uses the polar form of the equation of a straight line

$$\rho = x\cos\theta + y\sin\theta \quad . \ . \ . \qquad (1)$$

for transformation as proposed in [7].

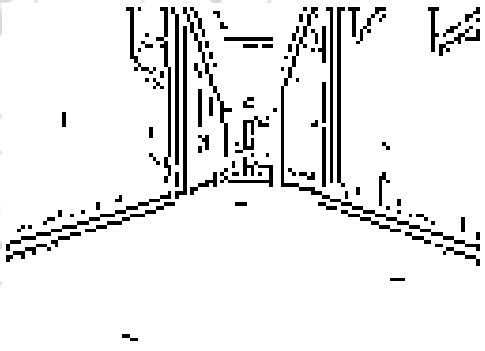

Figure 2: Typical thinned binary image after pre-processing

Using (1), a point (x, y) from the original image in the x-y plane is transformed to a curve in a new $\theta - \rho$ plane by keeping x and y constant and varying values for θ and ρ. Curves in the $\theta - \rho$ plane that are transforms of points on a straight line in the x-y plane, intersect on a point in the $\theta - \rho$ plane. The point they intersect at is the point (θ, ρ) where θ is the angle of the line to the vertical in the original image, and ρ is its distance from the centre of the image. Important lines in the original image become points in the transform image with relatively high number of intersecting curves. These are called peaks and finding them is called

peak detection. Peak detection can be accomplished by applying a threshold to the number of intersecting curves on points in the transform image. The lines which peaks found represent in the original image can then be found by rearranging (1) in the gradient-intercept form of the equation of a straight line

$$y = mx + c \quad . \ . \ . \quad (2)$$

giving

$$y = \rho \operatorname{cosec}\theta - x \cot\theta \quad . \ . \ . \ . \quad (3)$$

This gives

$$m = (-\cot\theta) \quad , \ . \ . \ . \quad (4)$$

and

$$c = \operatorname{cosec}\theta \quad . \ . \ . \quad (5)$$

where values for θ and ρ come from coordinate of the peaks in $\theta - \rho$ plane.

Thresholds for peak detection are chosen for each image in real time using a method described in [2] requiring a target number of peaks worked out empirically for this application as discussed in [8]. Figure 3 shows a typical pre-processed image with the lines detected super-imposed on it.

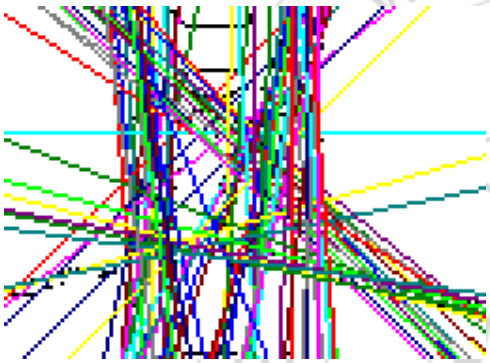

Figure 3: Lines detected in a typical image

With lines detected with the method described in [2], [3] haspresented a method to find all valid sublines of the lines detected, and specify the coordinates of their endpoints. Valid sub-lines have to meet criteria relating to minimum length and minimum separation from other sub-lines on the same line. Figure 4 shows sub-lines found for a typical image overlaid on the pre-processed version of the image. The pre-processed image is shown in black print, and other colours are used to show sub-lines found. Finding valid sub-lines further helps to reduce false and multiply detection of lines, in that lines without valid sub-lines are dismissed and not considered in further processing.

Figure 4: Sub-lines found in a typical image

Finally, [4] has presented a method for using lines found from [2] and [3] to estimate the vanishing point in the image by choosing candidate vanishing points from clusters of points of intersection of pairs of lines found, and evaluating them according to criteria chosen from the nature of vanishing points, and the images in this application. Figure 5 shows a sample result from the vanishing point estimation.

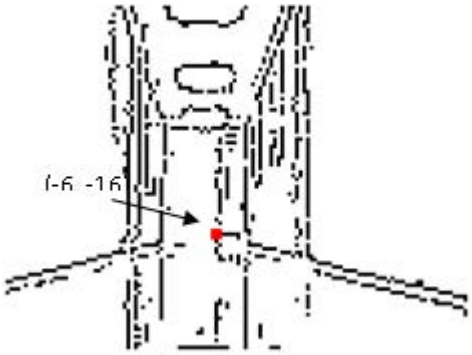

Figure 5: Vanishing point estimated in a typical image

This paper presents algorithms for recognition of corridors and doors in corridors from sublines and vanishing point. It starts out in *2. Category and Position Assignment to Lines*by grouping lines found into categories based on their angles (to the vertical), determining their distances from the vanishing point and assigning signs to the distances based on what side of the vanishing point they are on.*3. High Level Features Determination* then discusses how categories and displacements of lines from the vanishing point are used to recognise corridors and doors.

Others works have worked on detection of similar high-level features.

[9] also detect doors for indoor navigation of a robot. They use a convolutional neural network (CNN) to detect primary visual features such as angles of edges, corners and endpoints in a first stage, and combine these to recognise doors in later stages.

[10] use a Laser Range Finder (LRF) to find candidate lines for an elevator door detection system, and then use a camera image to select actual door edges. They use extrinsic calibration to fuse the LRF and the camera, and report that they get better results recognising doors, than using just LRF alone.

2. Category and Position Assignment to Lines

In detecting corridors and doors from images of the inside of a corridor by a small mobile robot, this work uses the vanishing point and lines detected as described in *1. Introduction*. Each line is assigned two attributes – type and displacement from vanishing point. Displacement consists of the magnitude of the distance from the vanishing point, and its direction or sign, i.e. whether or not the distance is positive or negative. *2.1 Line Category*, *2.2 Magnitude of Distance for Line from Vanishing Point* and *2.3 Sign of Displacement of Lines from Vanishing Point* discuss these further.

2.1 Line Category

Possible types of lines, and the directions in which their distances from the vanishing point are measured are shown in table 1 which follows.

The column on the far right summarises directions of distance measurement for the various line types. A line passing through the vanishing point is at 0 position and other values are negative to positive along the direction of distance measurement shown. A vertical line on the left side of the image for example, would have a negative distance and a

Table 1: Lines Categorisation

Category ID	Description		Minimum Θ	Maximum Θ	Range Size	Direction of Distance Measurement
0	Vertical		-5 (or 175)	4	10	left to right
1	Vertical Backslash		5	24	20	bottom left to top right
2	Backlash		25	64	40	bottom left to top right
3	Horizontal Backslash		65	84	20	bottom left to top right
4	Horizontal		85	94	10	bottom to top
5	Horizontal slash		95	114	20	bottom right to top left
6	Slash		115	154	40	bottom right to top left
7	Vertical Slash		155	174 (or -6)	20	bottom right to top left

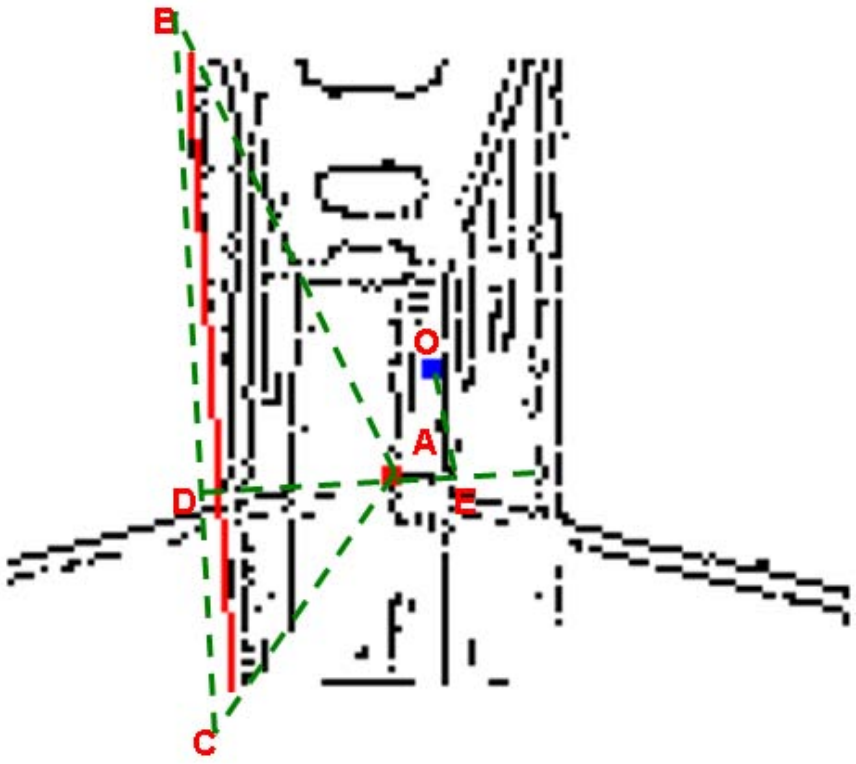

Figure 6: Determining magnitude of distance of line from vanishing point

horizontal slash on the top left side of the image would have a positive value.

2.2 Magnitude of Distance of Line from Vanishing Point

The magnitude of the distance of a line from the vanishing point is the perpendicular distance between the vanishing point and the line. This sub-section describes how it is determined.

As an illustration, figure 6 is a typical image showing the vanishing point as a red square labelled A (with coordinates (x_{VP}, y_{VP}), say) and one of the lines found, shown as a red line and labelled BC.

The distance between point A and the point labelled D is the perpendicular distance required. It is determined by first calculating θ_{per} and ρ_{per}, the parameters which define the perpendicular line between A and BC, that is the line AD. The angle of AD, θ_{per} is simply θ for the line under consideration, line BC, plus 90° i.e.,

$$\theta_{per} = \theta + 90° \quad \ldots (6)$$

ρ_{per}, the distance of the perpendicular line AD to the origin O (shown as a blue square labelled O), i.e. the length OE, can be obtained using

$$\rho_{per} = x_{VP}\cos\theta_{per} + y_{VP}\sin\theta_{per} \quad \ldots (7)$$

because it is known that the vanishing point (x_{VP}, y_{VP}) lies on the line AD. θ_{per} and ρ_{per} can then be plugged into (3) and (4) to obtain the gradient m_{per} and the intercept c_{per} of the line AD, so the equation of AD

$$y = m_{per}x + c_{per} \quad \ldots (8)$$

the point of intersection of line AD and line BC can then be determine using

$$x_{intersection} = \frac{-(c_{BC} - c_{per})}{m_{BC} - m_{per}} \quad \ldots (9)$$

and

$$y_{intersection} = m_{per}x_{intersection} + c_{per} \ldots \quad (10)$$

obtained by solving for $x_{intersection}$ and $y_{intersection}$ from the equation for line BC

$$y = m_{BC}x + c_{BC} \quad \ldots (11)$$

and (8).

The distance d between the intersection point ($x_{intersection}$, $y_{intersection}$) and the vanishing point (x_{VP}, y_{VP}) can then be worked out using

$$d = \sqrt{(x_{VP} - x_{intersection})^2 + (y_{VP} - y_{intersection})^2} \quad \ldots (12)$$

and that is the distance required. The distance between all lines found and the vanishing point, is evaluated in this way.

2.3 Sign of Displacement of Lines from Vanishing Point

To determine the sign of a line, the point in parameter space representing it is compared to the curve representing the vanishing point in parameter space. Where θ on the curve equals θ for the line, if the point lies on the negative side of the accumulator array relative to the curve, the sign of the line in the line categorisation scheme is negative and if it is on the positive side relative to the curve, the sign of the line is positive.

Take, for example, the lines found from a typical image in figure 7 after processing. Its pre-processed version, and the lines and vanishing point foundare shown in figures8and 9 respectively. An enlargement of the part of figure 9 enclosed by the red dashed rectangle,is showin numeric in figure 10.

Figure 7: Typical image

Figure 8: Pre-processed version of figure 7 image

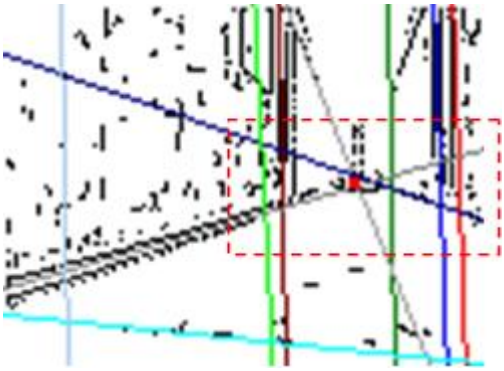

Figure 9: Lines found from figure 7 image

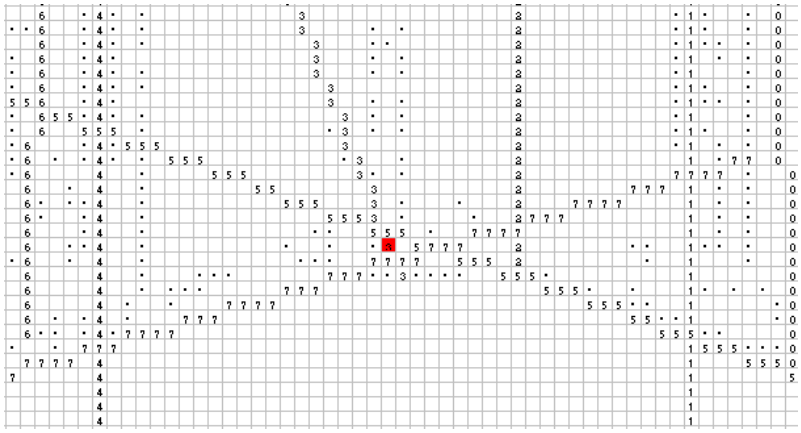

Figure 10: Area of figure 9 image enclosed in red dashed rectangle in numeric format

θ and ρ values for the lines found are shown in table 2.

Table 2: θ and ρ values for lines found in a typical image

Line id	θ	ρ
0	2	57
1	2	52
2	1	39
3	22	28
4	1	10
5	71	12
6	3	5
7	106	-8
8	84	-40
9	1	-48

The points representing lines 0, 2, 3, 4 and 6 are shown circled and labelled on an extract of the accumulator array in figure 11 as well as part of the curve representing the vanishing point (29, 1) found.

The column on the extreme left shows ρ values increasing from 5 to 52. Other columns correspond to the first 44 values of θ. In this figure any point lower than the curve has more positive ρ than the curve and so is positive relative to the curve. In our line classification scheme, the sign of such a line would be positive. That means lines 0, 2 and 3 would have positive displacements from the vanishing point while lines 4 and 6 would have negative displacements.
Mathematically, determination of the sign involves two steps:

1) determine the value of ρ on the curve for the vanishing point, ρ_{curve}, that corresponds to the value of θ, θ_{line} for the line under consideration
2) compare ρ_{curve} found from 1 with the value of ρ for the line under consideration, ρ_{line} and if this is less than ρ_{curve}, the sign is negative otherwise it is positive

Step 1 involves calculating

$$\rho_{curve} = x_{VP}\cos(\theta_{line}) + y_{VP}\sin(\theta_{line}) \quad ...(13)$$

where x_{VP} and y_{VP} are coordinates of the vanishing point, θ_{line} is θ for the line under consideration and ρ_{curve} is ρ on the curve at the point $\theta_{curve} = \theta_{line}$.

Step 2 will then involve the decision
if $(\rho_{line} < \rho_{curve})$ then assign a negative sign to the line's distance
else assign a positive sign to the line's distance

3. High Level Features Determination

3.1 Corridor(s) Recognition

Corridor recognition is critical to successful navigation in a corridor. In this work, this is achieved by searching for a left-corridor-edge and a right-corridor-edge. These refer to the intersection line between the wall of the corridor on the left and the floor, and the intersection line between the wall of the corridor on the right and the floor, respectively. A simple scheme was devised to achieve this.

To select a left-corridor-edge, points are assigned to lines according to whether or not they fit a certain criterion, and if so, how they score against three further criteria. The line which scores the highest is the left-corridor-edge provided its score is equal to or higher than the cut off score of 11. The criteria are as summarised in table 3 and are explained further below:

1) θ value in the range 90° to 180°: This is a necessary criterion ([11]) and lines are only assessed against further criteria if they meet this one.
2) SW to bottom-left-of-image distance: All lines which meet criterion 1 are ranked according to the distance of their most south-westerly point to the bottom left corner of the image. The one with the shortest distance receives a score of 5. The one with the next shortest distance receives 4, etc. Only the lines with the 5 shortest distances receive scores from this criterion.
3) Distance to the VP: All lines which meet criterion 1 are ranked according to their distance to the vanishing point. The one with the shortest distance receives a score of 5. The one with the second shortest distance receives 4, etc.

Only the lines with the 5 shortest distances receive scores from this criterion.

Line Category: Lines are ranked according to their category as defined in table 1. Lines in the horizontal slash and slash categories earn 7 points. Lines in the horizontal category earn 5 points. Lines in the vertical slash category earn 3 points. Lines in the vertical category earn 2 points.

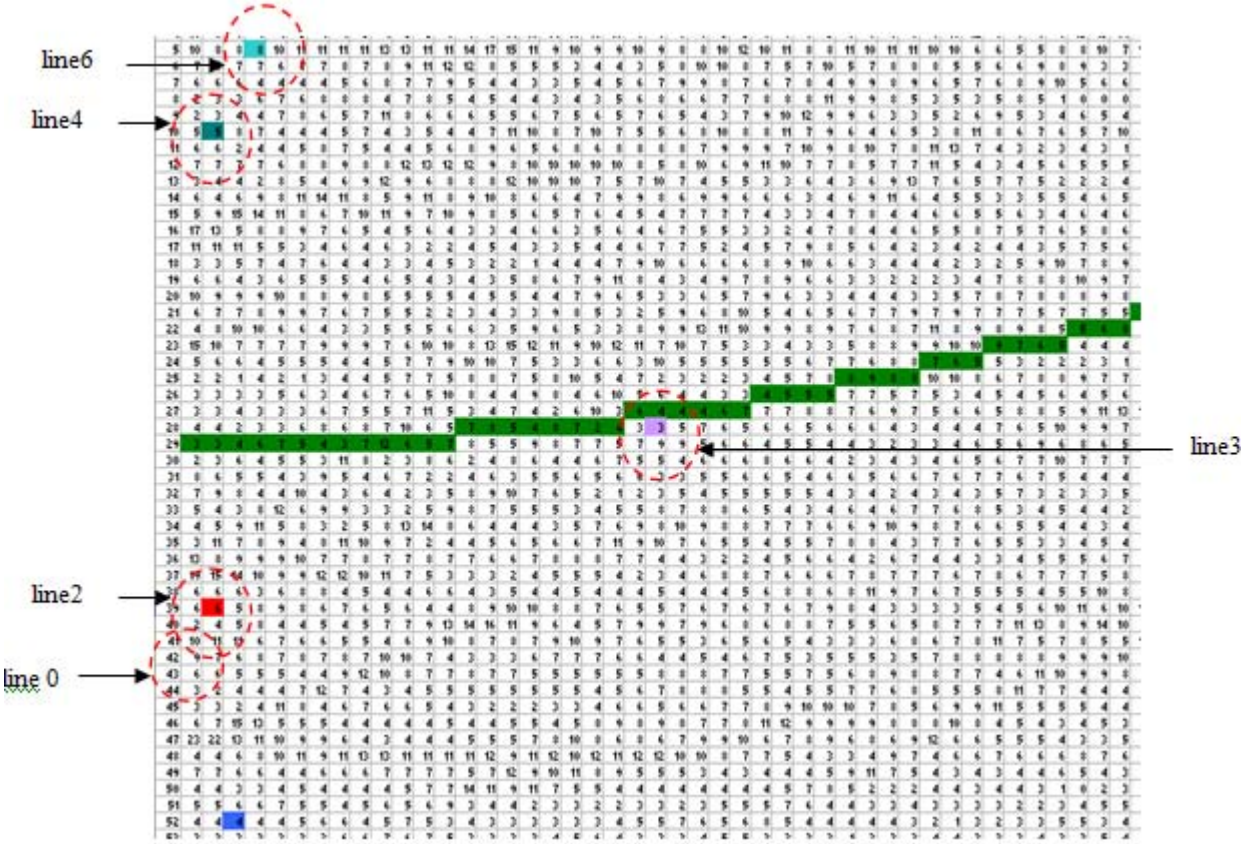

Figure 11: An extract of accumulator array showing some points found relative to the curve of accumulator array

Table 3 summarises the criteria for selection of left corridor edge.

Table 3: Criteria for selection of left corridor edge

	Criteria	*Maximum Points*
1	In the range 90 to 180	Necessary
2	*SW* to bottom-left-of-image distance	5
3	Distance from vanishing point	5
4	Line category	7

Similarly, the criteria for selection of a right-corridor-edge is summarised in table 4.

Table 4: Criteria for selection of right corridor edge

	Criteria	Maximum Points
1	In the range 0 to 90	Necessary
2	*SE* to bottom-left-of-image distance	5
3	Distance from vanishing point	5
4	Line category	7

The criteria for selection of a right-corridor-edge differ from those for a left-corridor-edge in a few ways. In criterion 1 for selection of a right-corridor-edge, the range angles of lines must fall in, is 0 to 90. In criterion 2, the distance considered is that of the most south-westerly point on a sub-line to the bottom right of corner of the image. Criterion 3 is exactly the same for both. In criteria 4, lines in the horizontal backslash and backslash categories earn 7 points, lines in the horizontal category earn 5 points, lines in the vertical backslash category earn 3 points and lines in the vertical category earn 2 points.

Sample results are shown in figure 12. Sub-lines found as corridor edges are shown in red.

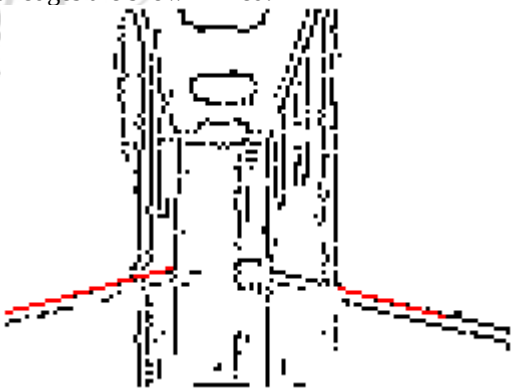

Figure 12: Sample result of corridor detection scheme

3.2 Door(s) Recognition

With the vanishing point estimated, and lines recognized and categorised, and corridor edges recognised, the following algorithms can be implemented to detect doors. Broadly, a door can be a corridor-door or a wall-door. A corridor-door

is a door at the end of a corridor. A wall-door is a door on one of the walls on the side of the corridor. A wall-door can be on the left, a wall-door-left, or on the right, a wall-door-right.

3.2.1 Corridor-Door Recognition

A corridor-door is found by searching for the left, top and right edges of the door.

The left edge is found by checking sub-lines against 2 necessary criteria, and where they meet these criteria, scoring them against 2 further criteria. The necessary criteria are:

1) it's parent line must belong to the vertical category as defined in *2.1 Line Category*
2) it must be on the left side of the vanishing point, determined from the sign of its parent line as determined in *2.3 Sign of Displacement of Lines from Vanishing Point*.

The two further criteria are – the distance of the candidate lines (the lines they are sub-lines of actually) from the vanishing point, and the difference between the y–coordinate of their midpoint and the y–coordinate of the vanishing point.

The first criterion was introduced because from observing several images, the door edge is usually on one of the first vertical lines encountered scanning from the vanishing point leftwards. The distance is as discussed in *2.2 Magnitude of Distance for Line from Vanishing Point*. Sub-lines on the closest line score 5 points, sub-lines on the next closest line score 4 points, etc.

The second criterion is necessary to improve scores for sub-lines which are at the right vertical level in the image to be door edges. The sub-line with the smallest difference scores 5, the one with the second smallest distance scores 4, etc.

The right and top edges of the door are determined in a similar manner with minor adjustments. For the right edge, only sub-lines to the right of the vanishing point are considered. For the top edge, horizontal sub-lines above the vanishing point are considered.

Figure 13 shows a sample result from running this algorithm.

The bottom edge of the door will usually not result in a line except when the door is closed.

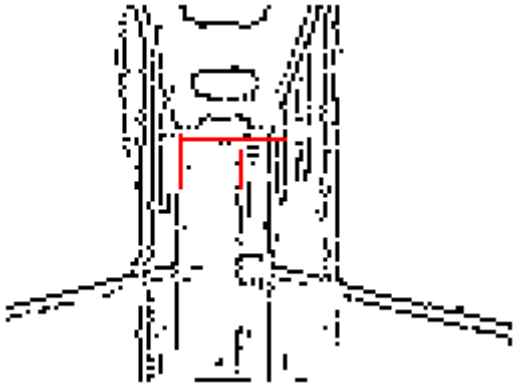

Figure 13: Sample door edges found

3.2.2 Wall-Door Recognition

To determine that there is a wall-door-right, a slash, or a vertical slash line is sought that does not coincide with the ceiling-wall intersection on the right. To do that, first the slash (or vertical slash or horizontal slash) sub-line corresponding to the ceiling-wall intersection is found, and then all other slash and vertical slash lines just below it are found. They are checked to see if any of them has a sub-line which has two vertical lines approximately at its ends which satisfy:

1) have a sub-line whose top edge approximately intersects with the top edge sub-line under consideration
2) have a sub-line which approximately intersects with the right-corridor-edge

Figure 14 shows a sample result. The top of the door is drawn in green. The two sub-lines of both sides which are edges of the door found are shown in red and blue for the first and second sub-lines found respectively.

Figure 14: Sample wall-door-right detected

The nature of the images encountered is such that the slash line corresponding to the top of a wall-door is not easily detected, so if a wall-door is not found, the image is checked for vertical sub-lines extending from the top of the wall where the door is to be found, to the bottom of it. Where such vertical sub-lines are found, it is possible that a wall-door exists on the wall, and another attempt should be made to find it again after the robot has moved a little.

A wall-door-left is found in a similar way except that a backslash and vertical backslash are the target categories of lines rather than slash and vertical slash respectively, and other similar adjustments are made.

4. Conclusions

Algorithms have been presented in this paper for recognition of corridors, doors at the end of corridors, and doors on the left and right walls on the sides of corridors, from within images captured by a front facing camera of a mobile robot for the purpose of self-navigating along a corridor. The algorithms rely for input, on other algorithms which pre-process the images and find lines and vanishing points in them. Results from limited tests conducted have been good.

The next stage of this work is to take the corridors and doors found at this stage, and put them to use in generating navigation instructions for the mobile robot.

The algorithms presented rely on scoring, thresholds and other parameters which were developed empirically specifically for this application, and so cannot be ported "as is" to another application. New parameters would need to be developed. The general idea can be ported to similar applications.

References

[1] 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
[2] 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
[3] 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
[4] 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.
[5] V. F. Leavers, "Shape Detection in Computer Vision Using the Hough Transform" (London: Springer-Verlag, 1992).
[6] G. K. Damaryam, E. Ogbuju and H. A. Mani, "An Investigation of the Costs and Benefits of Thinning on the Straight Line Hough Transform", International Journal of Advanced Computer and Communications Engineering, 5(1), pp 238-243, 2016.
[7] Duda and Hart. 1973. "Pattern Classification and Scene Analysis" (New York: Joh Wiley and Sons, 1973).
[8] 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.
[9] W. Chan et. al., "Door Recognition and Deep Learning Algorithm for Visual Based Robot Navigation", In Proceedings of the IEEE International Conference on Robotics and Biomimetics, 2014.
[10] M. Seung-Wan et. al., "Robust Elevator Door Recognition using LRF and Camera", Journal of Institute of Control, 18 (6), 2012
[11] A. O. Djekoune and K. Achour, "Visual Guidance Control based on the Hough Transform", IEEE Intelligent Vehicles Symposium, 2000.