# A Fully Geometry Based Model for Door Detection

Zoya Shafique
EMPLID: 2629
Computer Vision
Spring 2021

## Abstract

*Door detection is an essential task for indoor navigation aides. Here we present a fully geometric model based on edge detection for localizing doors and determining their orientation with respect to the viewer (e.g., if the door is open or closed). Our proposed network takes in RGB images of rooms and uses Canny edge detection, Hough transformations, and vanishing point geometry, to detect contours and calculate the orientation of the door. The final output is the input image with a bounding box drawn around the door and a 3D representation of the door with respect to the door frame. Our door detection model obtains accurate results for images with minimal background. However, as the background increases, the door localization worsens. Furthermore, the vanishing point detection accurately determines whether the door is opened or closed.*

## Introduction

In the United States, vision impairment (VI) is divided into four categories: partial sight, low vision, legally blind and totally blind [1], where partial sight refers to partial vision in either one or both eyes, low vision refers to severe visual impairment that cannot be corrected with glasses, and legally blind refers to a vision of 20/200 even with glasses. According to the Centers for Disease Control and Prevention, there are approximately 12 million people 40 years and older in the United States alone, who fall into one of the categories of vision impairment [2]. Furthermore, as of a 2019 report from the American Foundation for the Blind, approximately 55,249 children between 0 – 21 years old were classified as legally blind in 2018 [3]. To ensure day-to-day independence and comfort for those with a VI, navigation aids are a crucial and indispensable resource. However, despite many advances in computer vision and deep learning, the most used aids are still white canes and guide dogs [4]. Previously proposed computer vision aids come with bulky equipment or high costs [5], making them impractical for widespread use. Especially important for computer vision aids is the ability to recognize and detect people and important landmarks such as doors and furniture in indoor environments. In this paper, we focus on door detection and present a framework for a fully geometry-based door detection model. Our model has the advantage of not requiring high-cost equipment as it does not need a graphics processing unit (GPU).

## Related Work

With the advent and rapid growth of deep learning, much recent research has integrated deep learning into tasks based on conventional computer vision techniques. For instance, [6] uses a deep learning model to detect whether a door is open, partially open, or closed by calculating depth from a single input image. In conventional computer vision, depth cannot be calculated from a single image. Therefore, deep learning methods provide many more opportunities than conventional methods. They are also more robust. However, at the same time, employing a deep learning model requires the use of a GPU and can prove to be costly. Therefore, in this paper, we focus on geometry-based methods. In [7], the authors create a door localization model using purely geometric considerations. Specifically, the find the edges using Canny edge detection and corners of the door to create a more robust door detection compared to using features such as color and texture. The detected edges and corners are then constrained with respect to area and size to ensure that the model will detect a door and not any large, rectangular object in an image. Reference [8] also presents a robust door detection algorithm using Canny edge detection and the Hough transform. However, [8] employs fuzzy logic instead of geometrical considerations to determine if the detected regions conform to the characteristics of doors. In [9], the authors combine contour detection with constraints based on the shape, area, and perimeter of the contour to detect doors. Inspired by [7] and [9], we propose a fully geometric method for door detection.

## Algorithm

We present a fully geometric method for door detection based on Canny edge detection, Hough transform, and vanishing point geometry. Our model was deployed using Python 3.8.8 and OpenCV 4.5.4. All input images were taken by a Samsung Galaxy Note 10+.

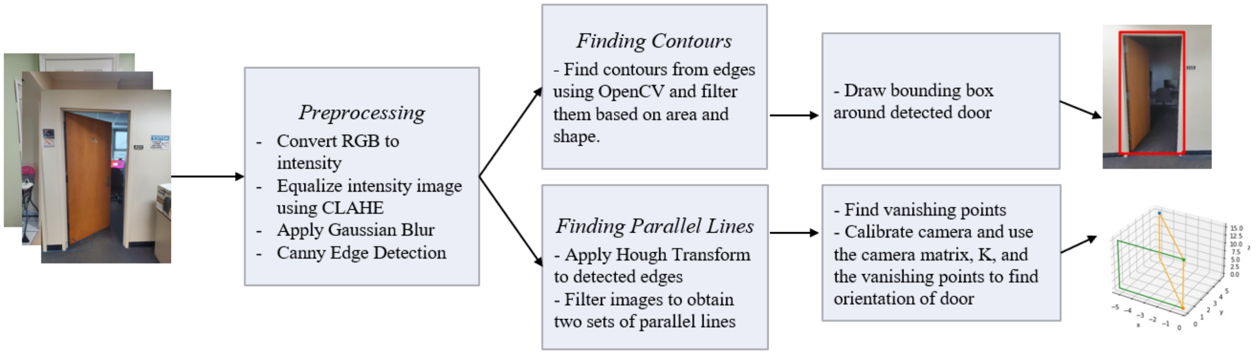


**Figure 1** Our proposed model. Our model takes in RGB images of an indoor space and after preprocessing, branches off into two separate streams. One stream (the top shown on the top in the figure) localizes the position of the door by using contours and drawing a bounding box around the detected door. The second steam uses Hough transforms to detect whether the door is open or closed.

The proposed model consists of two branches: door localization and orientation determination. Before being sent into the separate branches, the input RGB images were preprocessed. Preprocessing involved resizing the 4032 x 3024 pixels input to 672 x 507 pixels to decrease the runtime and number of computations, converting the RGB image into an intensity image, using contrast limited adaptive histogram equalization (CLAHE) and applying a 5 x 5 Gaussian blur filter with zero mean to smooth the image and decrease the effects of any noise in the subsequent edge detection. Edge detection was carried out using OpenCV's Canny edge detector with the threshold based on Otsu's threshold for each image. This edge map was then sent into both the door localization and orientation determination branches. An outline of our algorithm is shown in Figure 1.

## 3.1 Door Detection

Our goal for door detection was to localize the door frame in a given RGB image and draw a bounding box around it. To do so, we first applied morphological processing to the edge map. Specifically, we closed the gaps between any edges to form smooth lines without any gaps or holes as shown in Figure 2(b). This was done by using OpenCV. We then found all the contours in the image. A contour is formed by joining all the boundary pixels that have the same color and intensity REFERENCE. The detected contours are shown in Figure 2(c). As shown in Figure 2(c), contours were drawn around many objects in the background. Specifically, contours were drawn around the desk next to the door, c and around the mirror on the wall to the right of the door.

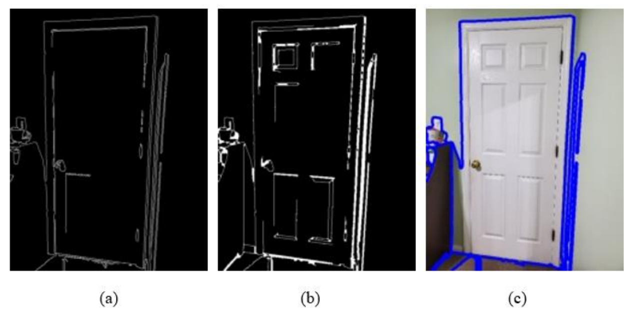


**Figure 2.** (a) Edge map detected by Canny edge detection. (b) We employed morphological closing to close holes and gaps in the edges and ensure a more accurate contour detection. (c) All the detected contours. From the general contour detection, undesired contours such as the ones on the desk, wall, and floor were also detected.

To ensure that the bounding box would only be drawn around the door frame, we filtered these contours based on their area and shape. Specifically, we assumed that if a door was present in an image, the contour around the door would have one of the, if not the, largest areas in the image. Therefore, we sorted the contours by area and chose the top 3 or 5 contours (depending on how many total contours were detected). This selection of contours was further refined by using built-in OpenCV functions such as approxPolyDP and arcLength to determine which contours formed rectangles and which of the rectangular contours had a perimeter between two given threshold values, respectively. The contour that fit the requirements of shape and arclength (i.e. perimeter) was outputted as the contour representing the door in the image. A bounding box was then drawn around this contour. An example of this fine tuning is shown in Figure 3.

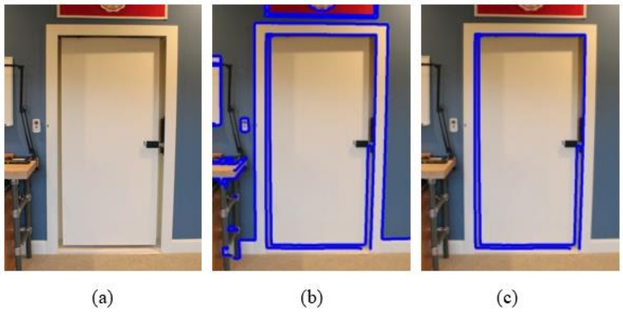

**Figure 3.** (a) Input image. (b) Contour detection. Contours were detected around the door frame, the desk to the left of the door, the light switch, and the board on the wall on top of the door. To filter the contours down so that only the contour around the door was detected, we placed constraints on the contour shape and size.

### 3.2 Camera Calibration

Camera calibration was carried out for the camera on a Samsung Galaxy Note 10+ using OpenCV. We obtained a checkerboard image from OpenCV, as shown in Figure 4 and taped it flat against a wall. Approximately 20 pictures were taken of the checkerboard, all from different angles and positions.

**Figure 4.** Examples of the pictures taken of the checkerboard for camera calibration.

These images were then uploaded to OpenCV's built in camera calibration functions to obtain the camera parameters. Specifically, we were interested in obtaining the intrinsic camera matrix, K, as given in Equation 1.

$$K = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

### 3.3 Vanishing Point Calculation

To find the vanishing points, we first used the Hough transform to find the lines in the input image and then filtered the lines down to two parallel sets of ones: one pair of (nearly) horizontal lines representing the top and bottom of the door and one pair of (nearly) vertical lines representing the left and right sides of the door. Note that here, the lines are tracing the door itself and not the door frame. The Hough transform was carried out by using the edge map and OpenCV and our results are shown in Figure 5.

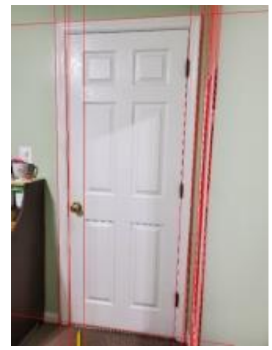

**Figure 5.** All lines detected by the Hough Transform in the input RGB image

The Hough transform detects lines by representing edges in the polar space, i.e., as ($\rho$, $\theta$) where $\rho$ is the perpendicular distance between the axis and the line and theta is the counterclockwise angle of the line with respect to the x-axis. Therefore, lines in the Hough space can be written as

$$\rho = x\cos\theta + y\sin\theta, \qquad (2)$$

where (x,y) is a point on the line in Cartesian coordinates. We can see from Equation 2 that, in the Hough space, vertical lines are represented by a theta of 0° or 180° while horizontal lines are represented by a theta of 90°. Based on this, we fine-tuned the final output of our line detection function based on whether we were looking for horizontal or vertical lines. This was done by applying a constraint on the value of theta. Specifically, to produce an image with one set of parallel horizontal lines, we set a constraint on theta such that only lines close to the desired theta were detected.

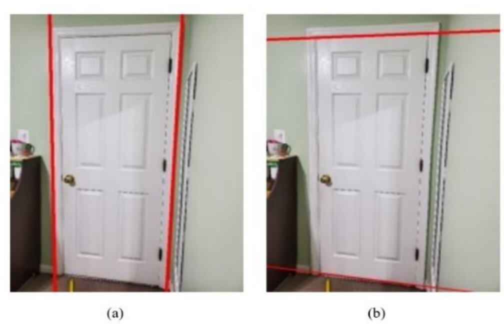

**Figure 6.** The detected lines were filtered depending on their theta to obtain two images: (a) a set of vertical parallel lines to represent the left and right sides of the door and (b) a set of horizontal lines to trace the top and bottom of the door.

A similar constraint was placed to detect solely vertical lines. The results of our fine-tuning are shown in Figure 6.

By definition, the vanishing point $(x_0,\ y_0)$ is the point in the image plane at which the projection of 3D parallel lines seem to intersect. Knowing this and that the representation of a line is Hough space is given by Equation 2, we formulated Equation 3 to find the vanishing point. We had to calculate the vanishing point for each set of parallel lines separately.

$$\rho_n \ = \ [x_0 \quad y_0] \begin{bmatrix} cos\theta \\ sin\theta \end{bmatrix} \tag{3}$$

Equation 3 was solved for $(x_0,\ y_0)$ using the least square fit method in NumPy.

### 3.4 Determining the Orientation

To determine whether the door was open or closed, we found the direction of the parallel set of lines by using the intrinsic camera matrix, K, and the vanishing point. We present the formulation of our method here.

Consider a three-dimensional line, *P(λ)*, represented by homogenous coordinates. We know that this line passes through some point, in this case the origin of our optical axis, o, and has some direction, d. Therefore, this line can be written as:

$$\mathrm{P}(\lambda) \ = \ \mathrm{o} \ + \ \lambda \mathrm{d}, \tag{4}$$

where $\lambda$ is a scaling factor. Note that Equation 4 represents a line in the camera coordinate system. To write this line in terms of the frame coordinates, we can multiply P($\lambda$) by the intrinsic camera matrix, K, as follows:

$$\mu \begin{bmatrix} \mathrm{p_x} \\ \mathrm{p_y} \\ 1 \end{bmatrix} \ = \ \mathrm{K} \cdot (\mathrm{o} \ + \ \lambda \mathrm{d}), \tag{5}$$

where μ is also a scaling factor for the homogenous coordinates $(\mathrm{p_x},\ \mathrm{p_y},\ 1)$. Since, by definition, the vanishing point is the projection of a line at infinity [10], the vanishing point can be written as

$$\mu \begin{bmatrix} \mathrm{v_x} \\ \mathrm{v_y} \\ 1 \end{bmatrix} \ = \ \mathrm{K} \cdot \mathrm{d}. \tag{6}$$

From Equation 6 we can say that the vanishing point uniquely determines the direction of the parallel lines. To find the direction, we can rewrite Equation 6 as

$$\mathrm{K}^{-1} * \mu \begin{bmatrix} \mathrm{v_x} \\ \mathrm{v_y} \\ 1 \end{bmatrix} \ = \ \mathrm{d}. \tag{7}$$

Equation 7 is used to determine the orientation of the door. For each set of parallel lines, Equation 7 outputs a point, d, to which a vector can be drawn from the door frame, as shown in Figure 7.

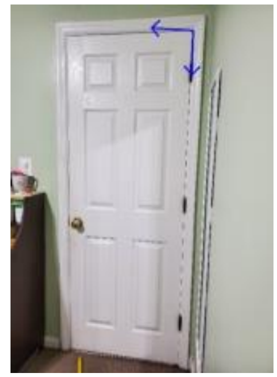

**Figure 7.** The blue arrows in the top right of the door frame represent the direction of the door with respect to the door frame. In this case, the door aligns with the door frame signifying that the door is closed.

## An Example

In this section, we breakdown our model and present the results after each step to detail an example of our model's functionality.

### 4.1 Preprocessing and Edge Detection

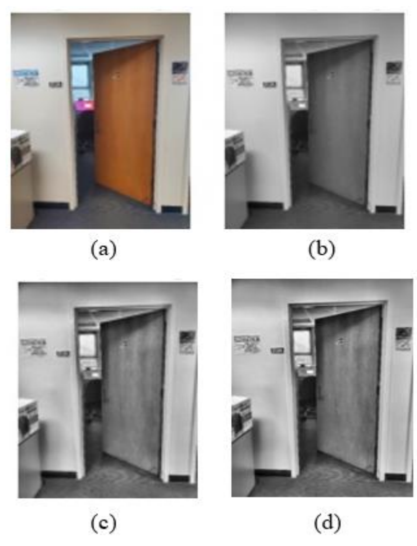

**Figure 8.** (a) Input RGB image (b) Intensity image (c) Image after CLAHE. (d) Image after Gaussian blur.

The results of the preprocessing are shown in Figure 8. Figure 8(a) depicts the input image, Figure 8(b) shows the image after conversion from RGB to grayscale, Figure 8(c) depicts the image after the application of CLAHE and lastly, Figure 8(d) depicts the image after application of

Gaussian blur. The Gaussian blurred and equalized intensity image was used to find the edge map as the last step in the preprocessing. The edge map is given in Figure 9.

The edge map was found using Canny edge detection. As shown in Figure 9, edges were detected for the door frame, the open door, and many of the objects in the background. To determine which of these edges belonged to the door, we used the edge map to draw contours.

**Figure 9.** Edge map created by Canny edge detection for the open door.

## 4.2 Contour and Line Detection

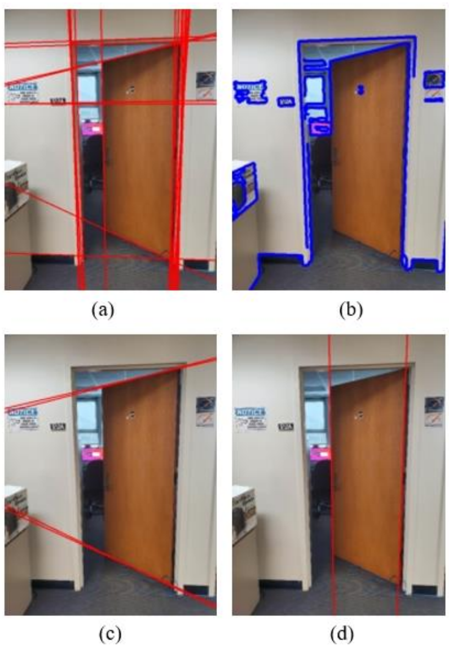


**Figure 10.** (a) All lines detected using the Hough transform. (b) All contours detected using OpenCV. (c) Filtered set of “horizontal” parallel lines to trace the top and bottom of the open door to the vanishing point. (d) Filtered set of vertical lines to trance the left and right of the door.

As detailed in Section 3.1, contour detection was used to find all the contours in the input image to localize the door. The results of the contour detection are shown in Figure 10.

To determine whether the door was open or closed, the Hough Transform was applied to detect lines as described in Section 3.3. Results of the line detection for the input image in Figure 7(a) are shown in Figure 10.

## 4.3 Door Localization and Orientation Results

From the detected contours, a bounding box was drawn around the door. The final output image with the door localization is shown in Figure 11. As seen from the figure, the bounding box highlights the door frame accurately.

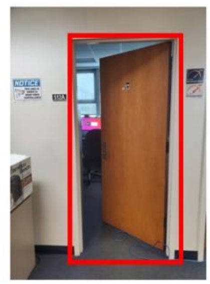

**Figure 11.** Output of the door detection stream. The bounding box localizes the door frame based on the detected contours and thus and thus the location of the door.

From the detected lines, the vanishing points were determined using the method detailed in Section 3.3. The results of the vanishing point calculation are given in Figure 12.

The vanishing points along with the calibrated camera matrix, K, were then used to determine whether the door was open or closed as detailed in Section 3.4. Two points representing the direction of each set of parallel lines were calculated. With these points known, the orientation of the

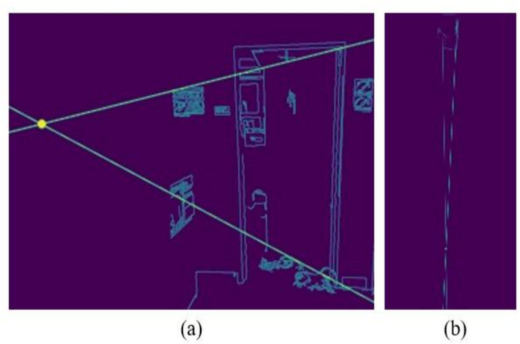


**Figure 12.** (a) Vanishing point calculation for the top and bottom sides of the open door. The lines along the top and bottom of the door are found from the Hough transform and intersect at the vanishing point.

door was determined by considering the top right corner of the door frame as the origin and drawing vectors to the calculated direction points from the origin.

## Analysis of Results

In this section we analyze the overall performance of our model with respect to door localization and the orientation calculation.

### 5.1 Door Detection Accuracy

Our door localization achieved varying accuracies depending on the input image as shown in Figure 13. When the input image had minimal background (e.g., there were very few objects in the image aside from the door, any other large object in the door was relatively far from the door), as the images in Figure 13(a), the door localization worked very well. However, as the images became more cluttered with other large objects and less centered on the door, the localization worsened (Figure 13(b) and 13(c)).

We believe that for the images in Figure 13(b), the contour detection algorithm could not detect large objects close to the door as separate objects and created one large contour to encompass both the door and the other large objects. Specially, the bushes, side of the desk, and flower vase shown in the images in Figure 13(b) were detected as part of the door rather than being assigned a contour of their own.

For the images in Figure 13(c), where the localization completely misses the mark, we believe that edge detection method was not able to identify strong edges and contours around the door because (1) in the first image, the door is the same color as the floor leading to an unclear boundary between the bottom of the door and the floor and (2) due to the door being off-center and at an angle with respect to the camera.


**Figure 13.** Door localization results of our model. (a) For images that were centered on the door, were taken with the camera pointing directly at the door with no angle and had minimal objects in the frame aside for the door, the localization worked very well. (b) For images taken at an angles and ones with other large objects near the door, the localization worsened but still encompassed the door. (c) For certain images, the localization was completely incorrect. We propose that this is due to issues in thresholding, edge detection, and contour detection.

### 5.2 Orientation Calculation Accuracy

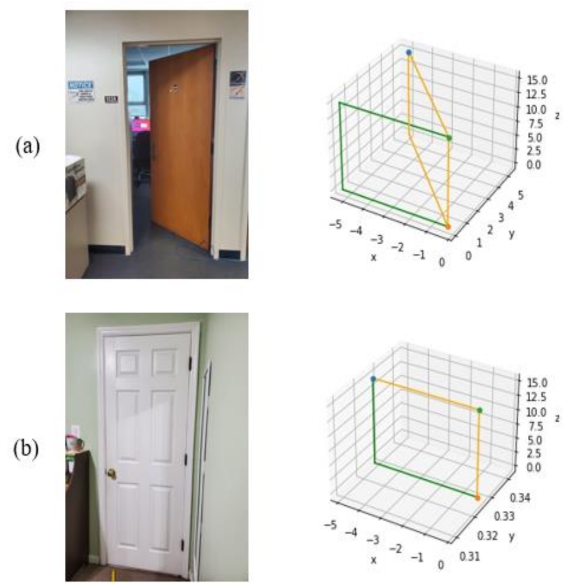


**Figure 14.** 3D visualizations of the door in the input image based on the calculated direction. The yellow lines represent the door, and the green lines represent the door frame. The blue and orange dots represent the calculated direction, and the green dot represents the top right corner of the door frame. (a) The calculated direction shows that the door is open. (b) The calculated direction shows that the door is closed.

In general, the direction points calculated by our method accurately aligned with the ground-truth value of whether the door was open or closed. Figure 14 depicts our calculation results for two doors. The yellow lines represent the door while the green lines represent the door frame. The 3D visualizations tell us whether the two overlap (i.e., if the door is closed) or if the yellow rectangle is at some angle with respect to the green one. Although we can determine whether the door is open or closed from these visualizations, we cannot distinguish between "open" and "partially-open". We also cannot obtain any other information about the door aside from the qualitative “open” or “closed”.

## Conclusion

In summary, we proposed a fully geometric model for localizing doors in RGB images and determining whether they are open or closed. The proposed model is a two-stream network which takes in a binary edge map and uses it to (1) uses contours to draw a bounding box around the door frame and (2) finds parallel lines in the image to determine the orientation. As part of the second stream which determines the orientation of the door, we calibrate our camera using built-in OpenCV functions to obtain the intrinsic camera matrix.

Although our model accurately localizes the door in images where there is minimal background, it is not robust and does not generalize well to images that are more cluttered. Specifically, for an image where the door is off-center or is close to other large objects such as a desk, our method fails to accurately detect the door. Furthermore, the second stream for determining the orientation only provides us with qualitative information on whether the door is open or closed and cannot differentiate between “open” and “partially-open”.

Further research can be done to make the proposed model more robust to improve door localization. Additionally, the proposed model only localizes the door frame. In future research, work can be done to localize the door itself rather than the door frame. Furthermore, the orientation calculation can be enhanced to include classification of doors as “partially-open”.

## References


[1] Industries for the blind and Visually Impaired. https://ibvi.org/blog/blind-vs-visually-impaired-whats-the-difference/

[2] “Fast Facts About Vision Loss”. Centers for Disease Control and Prevention . https://www.cdc.gov/visionhealth/basics/ced/fastfacts.htm

[3] Statistics About Children and Youth with Vision Loss. American Foundation for the Blind. https://www.afb.org/research-and-initiatives/statistics/statistics-blind-children#population18

[4] H. Zhang and C. Ye, "An indoor navigation aid for the visually impaired," *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2016, pp. 467-47

[5] R. Oktem, E. Aydin and N. E. Cagiltay, "An indoor navigation aid designed for visually impaired people," *2008 34th Annual Conference of IEEE Industrial Electronics*, 2008, pp. 2982-2987, doi: 10.1109/IECON.2008.4758435.

[6] Ramôa, J.G., Lopes, V., Alexandre, L.A. et al. Real-time 2D–3D door detection and state classification on a low-power device. SN Appl. Sci. 3, 590 (2021). https://doi.org/10.1007/s42452-021-04588-3

[7] Yang, X., & Tian, Y. (2010). Robust door detection in unfamiliar environments by combining edge and corner features. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 57-64.

[8] Juenemann C., Corbin A., Li J. Robust Door Detection. Stanford. https://stacks.stanford.edu/file/druid:rz261ds9725/Corbin_Li_Juenemann_RobustDoorDetection.pdf

[9] Sewell, Derek & Bellotto, Nicola. [2014] Door Detection for the Visually Impaired. University of Lincoln. https://thederek.github.io/door-detection/

[10] “Lecture 2: Vanishing points and horizons, Applications of Projective Transformations”. University of Adelaide. https://cs.adelaide.edu.au/~ianr/Teaching/CompGeom/lec2.pdf