Door Detection Using

Edward Meyer, Kendyll Doi, Keone Hiraide
Department of Information and Computer Sciences, University of Hawaii at Manoa
2500 Campus Rd. Honolulu, HI 96822
emeyer@hawaii.edu
kendyll@hawaii.edu
hiraide@hawaii.edu

Abstract - The ability to identify doors with computer vision provide a greater ability for autonomous assistance. While computer vision provides different implementations for door detection, they are often limited by controlled factors of visibility and input size. In this paper, we present an attempt at adapting different image feature identification techniques to isolate potential door features (door seams, door corners, door frame, door knob, and/or hinges). Potential doors are detected by identifying edges and corners, confirming the geometric shape, and noting existing features which meet a standard criteria of a door. This method is tested over a wide variety of door types and colors and positions in various settings and in instances of doors occupying different percentages of the image.

Keywords - computer vision, door, image analysis, assistive aid, robot navigation

I. Introduction

A common structural element in any building is the element of a door. While this common structural element is a standard for entry and exit, there is a seemingly non-standard way of designing a door. The problem lies when trying to formalize or make decisions on a structure when the structure itself does not follow a clear structural guideline. Some doors afford pushing, pulling, turning, or even just mere presence (automated doors) to open them. On first inspection of our problem it seems rather simplistic, however, we as humans have learned to recognize doors over a lifetime of experiences along with expectations of what a door can and should be. Our project aims at creating a system that accounts for a subset of variants in door recognition such as scaling, viewing angle, and varying levels of illumination. The applications of our project are mainly situated in the field of robotics. Automated robots with door detection have a greater autonomy with the potential of increased mobility as they gain the ability to recognize doors and as a result, gain the potential to enter and exit rooms on their own. Assistive aids for those visually impaired may also find increased mobility with the capacity to freely move through a household.

II. Background

Given the perceived benefits in the previous section, there have been a variety of attempts at solving this problem. The paper written by [7] designed, evaluated, and implemented a method to recognize doors based on analyzing its edges and corners. Information from the extracted corners and edges was compared to several threshold values and rules such as the ratios between the height and width of the door, whether certain edges were parallel with each other, whether the door was perpendicular to its horizontal axis, etc. The authors also found a solution for distinguishing between “door-like” objects from actual doors such as cabinets and elevators by extracting additional corners in the image. However, after analyzing the author's methodologies, we suspect that there are several limitations to this approach. For example, if a door-like object does not have additional corners, we suspect this would result in a false-positive. Also, the authors did not make use of any methods related to feature detection. Neural Networks of extracted features of a training set of doors have also demonstrated to have promising results [2],[3]. Attempts at remedying image processing techniques such as non-connected edges, corners, or doors that do not encompass the whole image have been thresholded through fuzzy logic with significant results [6].

III. Approaches and Methods

A. Isolation of Image Features

Image Feature represent the main points of interest within an image. Differences in color and intensity are the main way in which a still image can capture instances of objects in an environment. To gather data about the image features, we can analyze the images to isolate data related directly to those image features. Figure 1 shows the output our approach, illustrating the features points extract from the image.

Our system uses the Canny's Edge Detection algorithm process to convert the images from their original form to one designed to directly represent changes in color and intensity in the image. We individually analyze images to find reasonable levels to run the algorithm. Gaussian blur is applied to remove minor or widespread features which are not ideal to highlighting potential doors. The Canny Algorithm is applied with varying thresholds to further highlight features of the image.

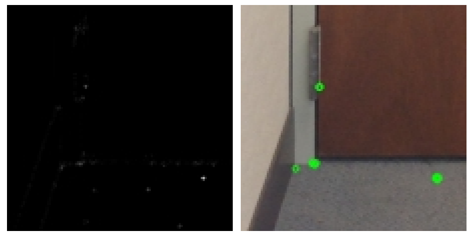
(a) (b)

Fig. 1. Processing images with Canny's Edge Detection: (a) The original image. (b) The image after being processed with Canny's Edge Detection and dilated. Dilation is done to make the image more recognizable to the human eye in this instance. Note that outstanding features are marked with white as the rest is made black.

B. Highlight Potential Corners

Corners points provide potential areas where the corners of doors exist as well as drastically changing feature points. An area mask of specific size is used to analyze the total intensity of an area and is then compared with the intensity of adjacent areas which overlap the area mask. Points which meet this criteria are marked at the origin of their mask. Areas of significant changes produce clusters of points which can be seen as areas of high features: corners or significant points of interest.

A version of the Harris Corner Detection algorithm is used to analyze a dilated version of Canny's Edge Detection image for corners for increased sensitivity. The image is checked pixel by pixel using the area immediately around a pixel. Image output features a map of points which meet the corner criteria as defined by the Harris Algorithm and marked with intensity based on how drastic the change in intensity is. Figure 2 shows the output of the Harris Corner Detection algorithm, as well as an overlay of corner clusters highlighted as circles over the original image for human analysis. The points provide another source of data which can be checked to find door shapes in a given image.



(a) (b)

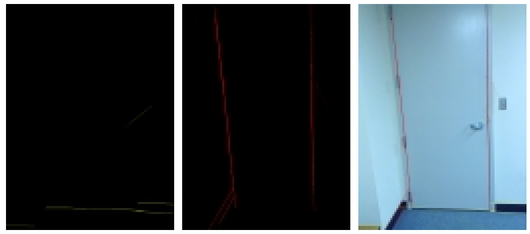
Fig. 2. An isolated sample of the processed images with Harris Corner Detection: (a) Corner points are displayed as white points (b) Points are filtered for high intensity points and overlaid onto an isolated portion of the original image using green circles for human visibility.

C. Highlight Potential Door Edges

Edges are made visible through the use of the Canny's Edge Detection algorithm. However, the edges given by the algorithm are not limited to those which could represent a door in the environment and are not transformed into an easily analyzed form of data. In order to compensate for these limitations, steps are taken in order to filter the data of lines present in the image and transform that filtered data into easily contained data points.

Both the filtering and transformation of the lines is done using Probabilistic Hough Transformation to detect curves (lines, circles, etc.) within the image. Probabilistic Hough Transformations is performed in our system using given thresholds rho and theta. Rho defines the minimal length of line in image to be considered for the transformation to identify the line. Theta defines the angle range from the the origin to the nearest point. For our algorithm we define rho as one unit and theta as one degree to maximum the number of lines output. These points are given as a series of start and end points.

To filter the lines, a significant length is calculated based on the longest point possible in a picture. Using pythagorean theorem, the significant length is stored as the value of the diagonal of the image. This value is then used to filter out any sizes which might be unsuitable for a door in an image, which is currently set to be 25% of the significant length. The lines are filtered by significant direction through angle calculation, by finding the arctan of the ratio of y points to x points. Horizontal lines are defined when their theta is within pi/4, and vertical lines defined as any point beyond that to pi/2. The storing of these values in two different sets (horizontal and vertical) allow comparisons to be limited to sets of horizontal and vertical lines, preventing the comparison of impossible sets. Figure 3 illustrates the Probability Hough Transformation applied to the output of Canny Edge Detection, separated into the horizontal and vertical sets. It should be noted that while lines appear to be solid, they are representing several potential edge lines overlapping each other.



(a) (b) (c)

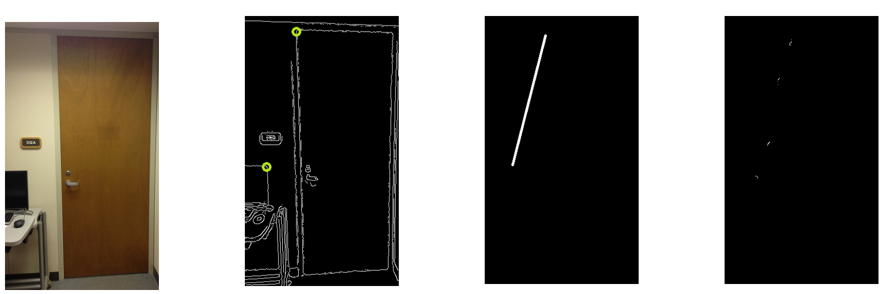
Fig. 3. The processed images based on Probabilistic Hough Transformation: (a) The output of horizontal lines (b) The output of vertical lines (c) The lines from Probabilistic Hough Transformation overlaid on top of the original image.

D. Fill-ratio Assertion for Edges

As mentioned in the previous sections, using the Harris corner detection algorithm, we were able to detect and output a set of features which represented the corners within our sample images. Although this set did contain the corners of the door that were of interest, it also contained several corners that were false-positives. Using our fill-ratio method, which was based off of [7]'s paper, we were able to minimize the false positives outputted by our Harris Corner Detector.

Our method differs from [7]'s paper's in that we use a "corner distance threshold". In our fill-ratio algorithm, the distance between each corner is compared with all other corners. Before the fill-ratio between each corner is calculated, the distance between these corners is compared by the set value of the corner distance threshold. This corner distance threshold is in place because we make the assumption that a door frame must have a certain width and height. Thus, if two corners have a distance that satisfies the corner distance threshold, then their fill-ratios are computed.

To compute the fill-ratio between any two corners, a straight-line is drawn between these corners. This straight line was created using a 3x3 mask. An output image containing the straight line between the desired corners is then created. An example of this image line is shown in Figure d To compute the amount of overlap, a logical AND between our canny's edge map and the image of the line was done. The fill-ratio was then computed by using the equation shown in Figure 4**.** If the fill-ratio between the corner points being analyzed has a high enough threshold, then we consider these corner points to be the corners of the door. The set of corners that satisfy our threshold are then outputted to an image; representing the corners of the door we are detecting.



(d) (e) (f) (g)

(h) (i) (j)

E. Criteria and Threshold

From the set of corners detected using our fill-ratio method mentioned in the previous sections, we then used several equations mentioned in [7]'s paper in order to determine whether the image being analyzed contained a door. From our set of corners, we can then measure the width and height of the potential door within the image. Figure 5a. calculates the ratio between the height or width of the door frame, and its diagonal. Figure 5b. is used to calculate the direction of either the door's vertical or horizontal lines.

$$Siz_{ij} = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{DI}$$

Fig. 5a. Siz_{ij} is used to calculate the ratio of between the width or height of a door frame and its diagonal.

$$Dir_{ij} = tan^{-1} \frac{|x_i - x_j|}{|y_i - y_j|} \times \frac{180}{\pi}$$

Fig. 5b. Dir_{ij} is used to calculate the direction of the door's door frame length.

Using Sizij and Dirij allows us to

F. Implementation

Using Java as our coding language, we implemented our door detection module using a computer vision framework called JavaCV; the Java interface based off of OpenCV (Open Source Computer Vision Library). JavaCV provided us with optimized versions of Canny's Edge Detection algorithm, Harris' Corner Detection algorithm, and several morphology operation methods such as dilation and erosion.

Although the majority of the functionality provided in OpenCV for C, C++, and Python was a direct port to Java, much of the documentation and examples provided were not directly applicable to the Java platform.

Therefore introductory development time was needed to get acclimated to the Java implementation of OpenCV. This in turn caused milestones to be pushed back and ultimately some milestones could not be met.

IV. Results

We were able to detect doors in their most simplest cases. That is, a frontal view with an image of a door. If an image of a door contained texture such as rust, contained occlusion, or had noise, our Harris Corner detection algorithm produced an excess amount of detected corners. Thus, with these excess corners produced by Harris, it was difficult to isolate the real corners of the door, and therefore, difficult to detect the door frame of the door.

V. Discussion

The current implementation of the system obtains several points of data which can be used to identify features of doors. Edges, corners, and feature points can be used together and analyzed for existing connections, and used to establish shapes which can be compared to doors. As of this current iteration the code fails to automate the process completely, requiring additional manual settings for thresholds and to properly find the edges, and corners of the image. In addition this current iteration fails to continue complete analysis of the images beyond the acquisition of data.

VI. Conclusions and Future Work

In our attempt to create a method for door detection, we have implemented various techniques to help accomplish our goal. In order to isolate image features of a door, we first use Canny edge detector to create an edge map with minimal noise. Because of the varying quality and settings of images: indoor, outdoor, occlusion, and perspective, each image was tuned specifically to provide an optimal edge map.

Next, probabilistic Hough Lines were used to extract horizontal and vertical line segments, this is used to detecting the outer frames of the door. Using the Hough Lines, Harris corner detection was used to create an optimal corner map to find the corners of the door image. From there, fill-ratio was used to create complete line segments.

Due to time constraints we could not create a fully functional module that could detect doors per say. Resulting module can detect basic features of doors such as frame and corner. Using probabilistic Hough lines resulted in vectors that visually looked complete, however, it was an overlay of continuous, separate line vectors that made the appearance of a single, continuous line. We attempted to remedy this issue using fill-ratio, however, the proportions of an American standard door, ~2.2 are not currently calculated.

VIII. References

[1] Anguelov, D., Koller, D., Parker, E., & Thrun, S. (2004, April). Detecting and modeling doors with mobile robots. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on* (Vol. 4, pp. 3777-3784). IEEE.

[2] Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

[3] E. Cokal and A. Erden, "Development of an image processing system for a special purpose mobile robot navigation," in The Fourth Annual Conference on Mechatronics and Machine Vision in Practice 1997, pp.246-252.

[4] Juenemann, C., Corbin, A., & Li, J. (2010). Robust Door Detection. *From Stanford Electrical Engineering Dept. Course EE368*.
[5] Murillo, A. C., Košecká, J., Guerrero, J. J., & Sagüés, C. (2008). Visual door detection integrating appearance and shape cues. *Robotics and Autonomous Systems*, *56*(6), 512-521.
[6] NOZ-SALINAS, R. M., Aguirre, E. U. G. E. N. I. O., Garcia-Silvente, M. I. G. U. E. L., & ALEZ, A. G. (2004). Door-detection using computer vision and fuzzy logic.
[7] Tian, Y., Yang, X., & Arditi, A. (2010). Computer vision-based door detection for accessibility of unfamiliar environments to blind persons. In *Computers Helping People with Special Needs* (pp. 263-270). Springer Berlin Heidelberg.