ROBUST DOOR DETECTION

Christopher Juenemann, Anthony Corbin, Jian Li

Abstract – A method for identifying door features in images was implemented in MATLAB to extract door locations from a set of random still images. The method is based on Canny edge detection algorithm and the Hough transform, and uses fuzzy logic to determine the probability of existence of doors which fit a set of predefined rules. The algorithm has an 80% rate of detection for an input set of 196 images. Average decision time is 7 seconds for 750-by-550-pixel image.

Index Terms – door detection, Hough transform, Canny edge detection, fuzzy logic

I. INTRODUCTION

This paper details the implementation of an image processing algorithm to extract the location of doors in still images. The work was performed as a Class Project for the course EE368 "Digital Image Processing" under Professor Bernd Girod at Stanford.

With the ever increasing processing power of mobile handheld devices, automated navigation can be realized on such devices to help the handicapped. In the complete system, a cell phone would detect doors, stairs, passable areas in front of a person, walk signs at intersections and cars, and identify friends approaching. The scope of the project is limited to the implementation of a door detection algorithm in MATLAB.

The algorithm includes a Canny edge detector to extract edges in a still image, uses a Hough transform to extract lines segments that match a given criteria, and fuzzy logic that determines if doors are present.

The algorithm limits detectable doors to ones which fit entirely within the image and are least 80 pixels tall. Detected doors can be open or closed and at different viewing angles. The algorithm also detects multiple doors in the same image.

II. PRIOR AND RELATED WORK

Door detection is one of the popular topics in computer vision and robotics. Thresholding, edge detection, and line detection are some of the methods used in common solutions. Creative approaches such as neural network based pattern recognition and knowledge based interpretation have also been combined with basic thresholding and edge detection. [2] Others have used a combination of weak feature identifiers, e.g. the color of the wall, a door knob or door gap, to create strong classifiers. [3] We found some of the prior work useful and adapted it in designing our algorithm.

III. IMPLEMENTATION

The implementation of the algorithm is composed of a few core steps: region of interest (ROI) preprocessing, door line extraction, and door classification.

1) ROI PREPROCESSOR

The preprocessing steps locate regions of interest (ROI) that are likely to contain doors. Door corners are likely among the strongest corners in the image; however, regular corner detection not only detects the corners of the door, but also any other strong corners in the image.

To differentiate the cases, the preprocessing algorithm takes into account that the corners for doors are near the intersection of strong horizontal and vertical lines. This sacrifices some degree of rotational invariance, but significantly reduces the number of undesirable corners.

The image is converted to grayscale and cropped slightly to remove border effects. Canny edge detection is then performed, and the resulting image is dilated with a 3x3 cross structuring element to improve connections at the corners. Dilation with horizontal and vertical line structuring element is performed before corners are computed. Region counting is then performed and each line segment is assigned a weighting based on the region size, as shown in Figure 1. Notice that the strongest values in the image are the door corners.

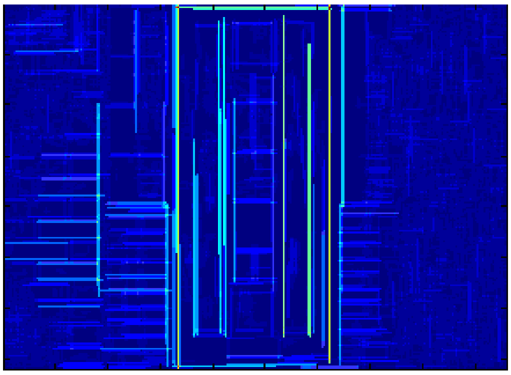

Figure 1 – ROI Preprocessor – Horizontal Vertical Line Counts after Dilation

Next the corners are determined by multiplying the squares of the horizontal and vertical segment images similar to as would be performed with a Harris corner detector with a sensitivity parameter equal to zero. Figure 2 shows the results and Figure 3 shows an overlay. Notice that the corners of the door are discernibly the strongest, especially near the top where the horizontal segment is clearer in the image.

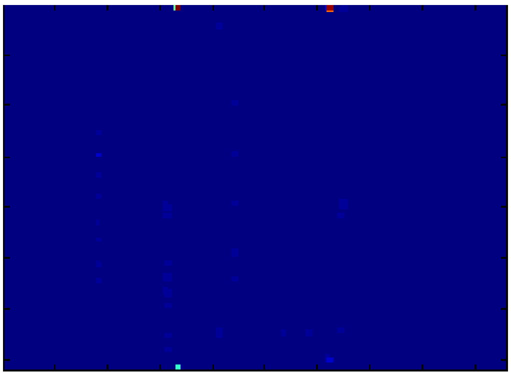

Figure 2 – ROI Preprocessor – Corner Detection

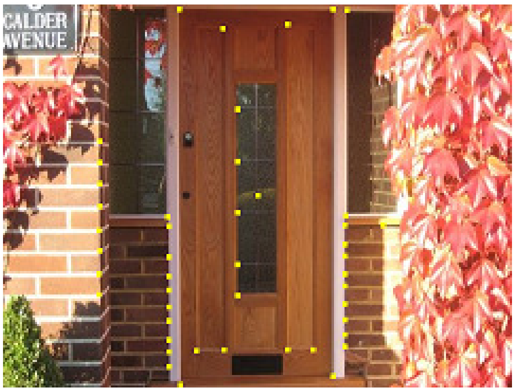

Figure 3 – ROI Preprocessor – Corner Detection Overlaid on Image

After corner detection is performed, multiple sets of three corners with an implicit fourth corner are used to generate quadrilateral regions where a door would likely be. During the initial search, the interior angles of the polygon must be within a tolerance comparable to those of a door viewed with a slightly rotated camera and have a height to width ratio which is acceptable. Next, the quadrilateral regions are each evaluated based on overall area and independently on a shape metric. Particular emphasis is given to quadrilaterals which could be described as rectangles and/or parallelograms and those which match the appropriate 2.2:1 standard for U.S. doors.

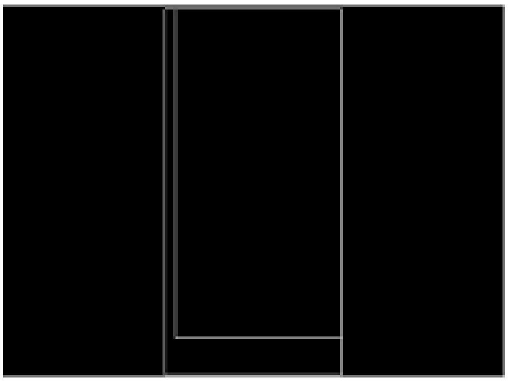

Figure 4 – ROI Preprocessor – Rectangular Region Detection

The quadrilaterals selected based on area are merged into one or more quadrilaterals of maximum extent. The quadrilaterals selected based on shape are evaluated for overall connection in the edge image. Each remaining quadrilateral region is converted to a rectangular region, shown in Figure 4, and each region is cropped from the original image, padded, and sent to the line and feature extraction algorithm. If the selected region is small, the image is upsampled before it is sent. Figure 5 shows an example output of the ROI preprocessing algorithm.

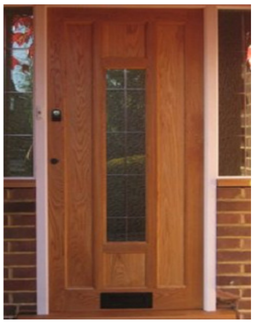

Figure 5 – ROI Preprocessor Sub-image

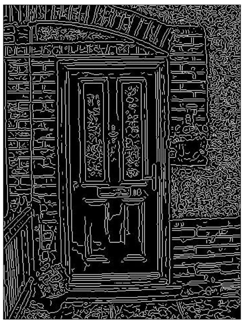

Figure 6 – Canny Edge Map

The ROI preprocessor increases the robustness of the door detection algorithm, and, depending on the image and parameters, can speed up detection. The detection rate increase is notable especially for images with small doors.

2) Door Line and Feature Extraction

With the ROI preprocessor removing most of the context around the door, the remaining image is grayscale and processed to extract dominant line features which are put into a line database.

The database extraction is a multistep process. First, the input image is converted into an edge map using Canny edge detection, with a very low fixed threshold. The output of the Canny edge detector is shown in Figure 6.

Morphological erosion with 3x1 pixel line structuring elements is used to extract horizontal and vertical lines from the edges. The resulting horizontal and vertical edge maps are shown in Figures 7 and 8.

Region counting is applied to remove small vertical and horizontal lines from the respective images. Only regions of sufficient size are retained. A Hough transform is applied to each horizontal and vertical line. The theta and rho values are stored, along with the line segment locations, in a database. The final line database consists of horizontal and vertical lines that meet a length criterion and an angle criterion.

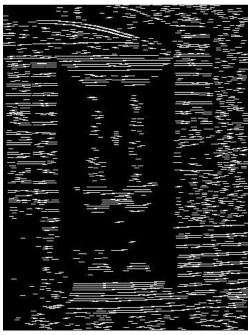

Figure 7 – Horizontal Edge Map

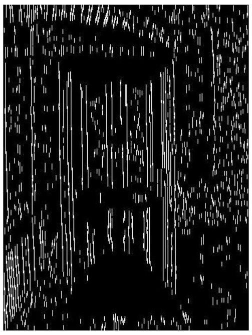

Figure 8 – Vertical Edge Map

8-level Otsu thresholding is applied to the original image passed by the ROI preprocessor. Note that the Otsu thresholding function was leveraged from [4]. Each Otsu level is turned into a binary image with all levels below the current level set to 0 and all levels equal to or larger than the current level set to 1. Each binary image is region counted looking for features that are the correct size for hinges or door knobs.

Figure 9 – Lines, Door Knob, and Hinges Overlay

Figure 10 – Final Door Detection Overlay

The hinge locations are stored in a database, while the door knobs are stored in a separate database. The three databases are shown overlaid in Figure 9. The result is then passed onto an algorithm utilizing fuzzy logic to calculate confidence.

3) Fuzzy Logic to Extract Doors

A heuristic method is used to extract parallel lines from the line database. The parallel lines are used to find the four corners of the door.

The heuristic method employs three metrics to determine the likelihood of line segments being part of a door. One metric is assigned to vertical line pairs, the second metric is assigned to the top horizontal line, and the third metric is assigned to the bottom horizontal line. High metric values indicate more confidence that the line combination constitutes a door. The steps used to calculate vertical line metrics are as follows:

Calculate the mean length of each pair of vertical line segments. If the distance between the pair of lines is greater than 1/3rd and less than 2/3rd of the mean length, the pair is considered further.

The difference between the two vertical lines is subtracted from the metric. If the distance between the vertical lines multiplied by 2.2 is close to the length of the vertical line pair, the metric is increased. Vertical line pairs with a metric less than 50% of the highest vertical line metric are removed from consideration.

Once the vertical line pairs are scored, the fuzzy logic scores the horizontal lines with respect to each pair of vertical lines still under consideration. The average top and bottom Y coordinates of vertical lines are calculated. Each horizontal line has a metric calculated based on how close the average Y coordinate of the horizontal line matches the top or bottom average Y values of the vertical line pair. The lines with the highest metric are stored as the top and bottom that match with the vertical line pair. Horizontal lines that do not span the entire distance between the two vertical lines score a lower horizontal metric.

Once all three base metrics are calculated, the four lines constituting the door are passed to the door feature detection section of the fuzzy logic. The door feature detection section uses the location of the door as described by the four lines to search for features which improve the confidence score. The door feature code works as follows. A door handle near either vertical line in the second quadrant from the bottom of the door adds half the difference between the vertical line score and the top and bottom horizontal line scores independently. Hinges near either vertical line add a sixth to the difference between the vertical line score and top and bottom horizontal line scores independently. The door feature detection helps significantly with close door detection, which is defined as doors closer than 6 ft. This is because the top or bottom of the door is commonly missing from these images. However, door handles and hinges become more easily identifiable with thresholding as these features are larger in close door pictures than other picture types.

The feature metric values are summed with the original vertical and horizontal door metrics to create the final door confidence metric. These final values are sorted from highest to lowest. The last section of door confidence is determining if the metrics constitute a door, and how many. The highest ranked combined metric, is indicative of the existence of a door, if its horizontal metrics are at least 1/4th of the vertical line metric. In addition, all lower confidence doors are thrown away if their area overlaps 50% or more of the higher scored rectangle. If the top few candidates have close metrics and their areas do not overlap, it is indicative of multiple doors in the image.

An example door region result is highlighted in Figure 10.

IV. Experimental Results

The code was tested against several different door types, in different scenes, and at different camera angles. The original proposal only had simple doors, doors in sidewalls, open doors, and doors in scenery as its objectives. It was quickly determined that doors in sidewalls are extremely difficult as the door aspect ratio, which is critical to robust door detection, is grossly skewed in this application. However, since the remaining applications were solved quickly in the time allotted, the goals were expanded to include doors with angles up to 30° from straight on, finding multiple doors in the same picture, finding doors within 6 feet of the camera, finding

French doors, and finding doors where the frame has obstructions in front of it.

The training set was 30 random pictures from the test set of 196 doors. There was at least one door of each type in the training set. The output of the program was a quadrilateral that covered the area of the door in the input image. The output was considered correct if the colored area constituted more than 90% of the door area and did not exceed 110% of the door plus door frame area. The output was considered mostly correct if the selected door area was in the correct location of the image and was greater than 50% of the door area but not exceeding 130% of the door plus frame area. In the case of multiple doors and French doors, the output was considered mostly correct if it detected more than half of the door(s) with the correct size. As far as determining correct percentage, mostly correct findings were weighted 0.5, correct doors were weighted 1.0, and missed doors were weighted 0.0. The results are shown in Table 1.

Even though the program was run against still pictures, the hope of running this on a mobile device against real time video was a guide post for tempering the algorithm. The detection rate was higher than 80% in two previous versions, but the execution time of those algorithms seemed excessive, therefore simpler less time consuming methods were developed and then optimized to improve the door detection rate. This constant struggle of trading off processing time versus detection rate led to three major revisions during the coding process. The group is very satisfied with the average processing time of the current algorithm being 7 seconds for a 750 by 550 pixel image.

The program averages 2 seconds per image of this size because the door is usually only a small part of the input image. In these cases, the ROI preprocessor dissects the image and gives the longer door detection algorithm smaller images to work on, which speeds its processing immensely. The algorithm could be improved to increase detection rate, but the group is confident that the current algorithm's speed of detection has been fully optimized. The program uses feature detection, canny edge detection, morphological operators, thresholding, and region counting to robustly detect doors in images. Each operation has been simultaneously optimized to reduce execution time and improve detection rate.

Table 1 - Experimental Results

Door Type	Missed	Mostly Correct	Correct	Correct %
Rectangle	2	4	36	**90%**
Open	2	3	15	83%
Scenery	6	19	55	81%
French	1	3	0	38%
Closed	1	1	3	70%
Multiple	2	9	8	66%
Angled	3	5	18	79%
Obstacle	1	1	5	79%
Total	**18**	**45**	**140**	**80%**

The average execution times given in this report were taken on a midrange x86-based machine running MATLAB. Actual implementations compiled and optimized for a mobile platform would likely see significantly improved performance. Additionally, the scope of the mobile application could be further limited allowing a more practical application of the algorithm.

V. Conclusions

The team has successfully developed and implemented an algorithm to detect doors in still images based on a heuristic approach to detect lines and door features such as door knobs and hinges using various image processing techniques. The method achieves an average detection rate of 80% with an average delay of 7 seconds per 750-by-550-pixel image. We have found pre-processing the image to determine region of interest significantly speeds up the algorithm. The detection rate is most likely limited by the small number of training images.

Improvements can be made to the algorithm by coming up with classifiers in addition to the existing heuristic classifiers and by optimizing the threshold values using a larger set of training images.

VI. BIBLIOGRAPHY

[1] Shi, et. al. "Investigating the Performance of Corridor and Door Detection Algorithms in Different Environments," International Conference on Information and Automation, p. 206-211, 2006

[2] Cokal, et. Al. "Development of an Image Processing System for a Special Purpose Robot Navigation," Middle East Technical University, Turkey. http://mechatronics.atilim.edu.tr/aerden/publications/m2vip97.pdf

[3] Hensler, et. Al. "Real-time Door Detection based on AdaBoost learning algorithm," University of Applied Sciences Konstanz, Germany. Laboratory for Mobile Robots. http://www-home.fg-konstanz.de/~bittel/Publikationen/DoorDetection_Eurobot2009.pdf

[4] Damien Garcia, Assistant Professor, Department of Radiology, University of Montreal Hospital, Montreal, QC, Canada http://www.biomecardio.com/matlab/otsu.html

VII. APPENDIX

Team Work Distribution

Chris	Created line and feature database methods. Created fuzzy logic door confidence metric system. Explored and used the advantages of Otsu thresholding. Created and edited portions of the report.
Tony	Created region of interest extraction algorithm and documented it in the report. Investigated line extraction techniques and applications. Evaluated multiple edge detection filters and structuring elements. Created poster.
Jian	Worked on line extraction algorithm and investigated correlation between image corner and extracted lines. Create, edit and proof portions of the report. Collect test images and build test image database.