Workshop on Visual Localization for Mobile Platforms
(in association with CVPR)
Anchorage, Alaska, June 2008

Visual Detection of Lintel-Occluded Doors from a Single Image

Zhichao Chen Stanley T. Birchfield
Electrical and Computer Engineering Department
Clemson University, Clemson, SC 29634
{zhichac, stb}@clemson.edu

Abstract

Doors are important landmarks for indoor mobile robot navigation. Most existing algorithms for door detection use range sensors or work in limited environments because of restricted assumptions about color, pose, or lighting. We present a vision-based door detection algorithm that achieves robustness by utilizing a variety of features, including color, texture, and intensity edges. We introduce two novel geometric features that increase performance significantly: concavity and bottom-edge intensity profile. The features are combined using Adaboost to ensure optimal linear weighting. On a large database of images collected in a wide variety of conditions, the algorithm achieves more than 90% detection with a low false positive rate. Additional experiments demonstrate the suitability of the algorithm for real-time applications using a mobile robot equipped with an off-the-shelf camera and laptop.

1. Introduction

Indoor environments are highly structured places that are designed with specific constraints on the shape, size, relative location, and orientation of navigational components such as doors, walls, and corridors. Discovering and distinguishing these components are key for robust mobile robot navigation in such environments. Doors, in particular, are important landmarks for navigation because they provide the entrance and exit points of rooms, and because they provide stable and semantically meaningful structures for determining the location of the robot. As a result, the capability of robust real-time automatic door detection would benefit many applications, including courier robots, tour guides, and patrol robots.

Much of the previous work on door detection has relied upon 3D range information available from sonar, lasers, or stereo vision [7, 11, 13, 1]. We are interested, however, in using off-the-shelf cameras for detecting doors, primarily because of their low-cost, low-power, and passive sensing characteristics, in addition to the rich information they provide. Figure 1 illustrates our scenario, as well as the difficulties of solving this problem. The robot is equipped with two webcams, each one pointing at a different side of the hallway as the robot drives. Because there is no overlap between the cameras, stereo vision is not possible. Even more importantly, because the cameras are low to the ground, the top of the door (the *lintel*) — which otherwise would provide a powerful cue for aiding door detection — is often occluded by the top of the image. Pointing the cameras upward is not possible, because of the importance of being able to see the ground to avoid obstacles. Even with these constraints, our goal is to detect doors in a variety of environments, containing low-contrast edges, bright reflections, variable lighting conditions, textured and untextured floors, walls and doors with similar colors, and changing robot pose, as shown in the figure.

In this paper we present a solution to this problem based upon combining multiple geometric and photometric cues using the Adaboost algorithm. Our approach augments standard features such as color, texture, and vertical intensity edges with novel geometric features based on the concavity of the door and the gap below the bottom door edge. We demonstrate the performance of the approach on a large database of images from a variety of environments.

1.1. Related work

Two visual cues have been employed by previous researchers: intensity edges along the sides (*posts*) and top (*lintel*) of the door, and the average door color which is assumed to be different from that of the surrounding wall. For example, Stoeter *et al*. [13] extract vertical lines in the image using the Sobel edge detector followed by morphological noise removing. The resulting lines are then combined with range information from a ring of sonars to detect doors. In similar work, Kim *et al*. [7] extract both vertical and horizontal line segments, then analyze whether the segments meet minimum length and height restrictions. Door candidates are verified by a 3D trinocular stereo system. The recent system of Anguelov *et al*. [1] combines an omnidirectional camera and laser range finder in an expectation-

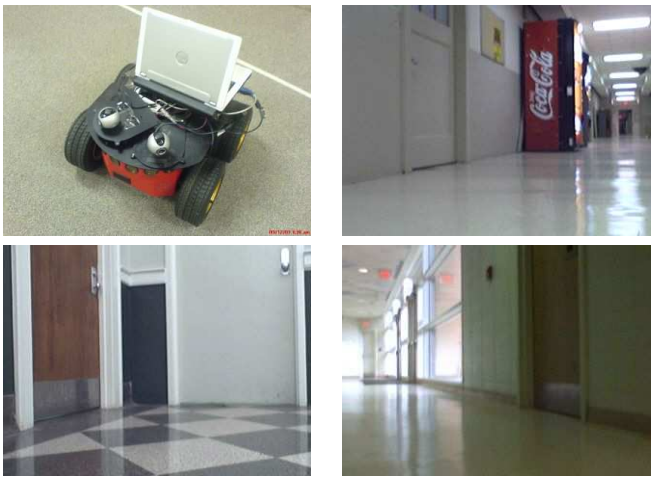


Figure 1. TOP-LEFT: Our robot is equipped with two non-overlapping off-the-shelf webcams, mounted on top (30 cm above the ground). TOP-RIGHT: An image taken by one of the cameras, showing a door whose color is the same as the surrounding wall and whose lintel is not visible. BOTTOM: Two additional examples, showing doors at drastically different poses and colors, along with a variety of floor patterns and lighting conditions. These challenges make vision-based door detection difficult.

maximization framework. Doors are detected by observing their motion over time (i.e., whether an open door later becomes closed, or vice versa), and the similarity of their color to a globally estimated mean door color. This approach assumes that the doors are all similarly colored, that the walls are similarly colored, and that the mean color of the doors and walls are significantly different from each other.

Other researchers have focused upon using visual data alone, without range information, which of course is a much harder problem. In an approach that assumes that the robot is facing the door at a certain distance, Monasterio *et al*. [9] detect edges corresponding to the posts and then classify the scene as a door if the column between the edges in wider than a certain number of pixels. Similarly, Munoz-Salinas *et al*. [10] apply fuzzy logic to establish the membership degree of an intensity pattern in a fuzzy set using horizontal and vertical line segments. Rous *et al*. [12] generate a convex polygonal grid based on extracted lines, and they define doors as two vertical edges that intersect the floor and extend above the center of the image. Their work employs color information to segment the floor, thus assuming that the floor is not textured. An alternate approach by Cicirelli *et al*. [4] analyzes every pixel in the image using two neural networks: one to detect the door corners, and one to detect the door frame, both of which are applied to the hue and saturation color channels of the image. While this system is able to detect doors under partial occlusion conditions and from different perspectives, its incurs a high computational load (approximately three seconds per image), making it unsuitable for a real-time system.

There remains a need for a vision-based door detection system that operates in real time and is capable of handling a variety of environments and the lintel-occlusion that often occurs when the camera is low to the ground and the door is nearby. It is our belief that the solution to this problem cannot be achieved by focusing only upon one piece of local evidence, such as lines or color. Rather, the integration of a variety of cues is needed to overcome the noise of sparse, local measurements. As a result, we approach the problem recognizing that recognition is inherently a global process.

2. Multiples cues for door detection

Our approach to door detection utilizes a single image. Let I be the image, and let D_ℓ be the predicate that is true if a door exists in the image at location ℓ. According to Bayes' decision rule, we declare a door if the *a posteriori* probability of the predicate is greater than that of its complement:

$$\frac{P(D_\ell|I)}{P(\neg D_\ell|I)} = \exp\left(\Psi(D_\ell|I)\right) > 1. \tag{1}$$

The energy $\Psi(D_\ell|I)$ has the form

$$\Psi(D_\ell|I) = \sum_{n=1}^{N} \alpha_n h_n(D_\ell|I), \tag{2}$$

where the weak classifier $h_n(D_\ell|I)$ tests whether the predicate is true given the data using some projection of the data, with positive values of h_n indicating truth and negative values indicating falsehood. The weights α_n govern the relative performance of the individual tests, which are assumed to be conditionally independent.

In this work we use five features described in the following subsections. Weak classifiers based on these features are then combined using the Adaboost algorithm to produce a strong classifier. Our approach makes the following assumptions:

- Both door posts are visible in the image,
- The camera is oriented so that these posts appear nearly vertical, and
- The door is at least a certain width.

Beyond these requirements, the algorithm is quite forgiving: The color of the door may be the same or different from that of the wall, the geometry of the door can be receded into the wall or flush with the wall, and the lintel may be visible or not. By utilizing multiple features in a boosting framework, the absence of one feature will not cause a door to be missed as long as there is sufficient evidence in the other features to overcome this loss.

2.1. Detecting pairs of vertical lines

The first weak classifier looks for pairs of vertical lines whose length and horizontal spacing between them is greater than a threshold, and whose top extends above the vanishing point. Intensity edges are detected throughout the image using the Canny edge detector [3]. To group the edges into straight line segments, many existing approaches apply the Hough transform, but in our experiments we found this approach to be quite sensitive to the window size: Small windows cause unwanted splitting of lines, while large windows cause unwanted merging of lines.

Instead, we employ the algorithm `LineDetection`, which is a modified form of the divide-and-conquer algorithm developed by Kovesi [8]. In the first step, edge pixels are searched in order to label connected edges. Edges are divided by junction points, which are defined as edge pixels that are connected to more than two other pixels in their 8-neighborhood. Small isolated edges and spurs (short sequences of pixels jutting to the side of the main branch of the edge) are eliminated.

The second step divides the connected edges into sequences of straight line segments using a divide-and-conquer strategy. A straightness test is recursively applied to the edge, stopping when all segments pass the test. In Kovesi's algorithm, the threshold for the maximum allowed deviation $d_{allowed}$ is a constant, which mistakenly absorbs short line segments into long line segments and mistakenly divides long line segments into multiple short segments. Our modified version of the algorithm solves this problem by determining $d_{allowed}$ using a half-sigmoid function:

$$d_{allowed}(s) = \delta \left(\frac{\beta - e^{-|s|}}{\beta + e^{-|s|}} \right), \qquad (3)$$

where $|s|$ is the length of segment s, δ is a constant specifying the value of $d_{allowed}$ for long segments, and $\beta > 1$ is a constant to increase the slope of the half-sigmoid function. This function is shown in Figure 2, and the improvement from the modification is shown in Figure 3.

The third step merges small line segments if the angle deviation between them and the maximum distance between end points are below specified thresholds.

Figure 4a shows a typical result from this algorithm. Although straight line segments are found, several problems remain. For example, vertical lines corresponding to the door frame are often broken by door hinges or knobs. In addition, the reflection of the door creates spurious lines on the floor, often with a gap between the spurious lines and the true lines. To overcome these problems, we merge vertical lines separated by a small gap, discard lines that do not extend above the vanishing point, discard short lines, and retain only lines whose orientation is nearly vertical. The result of these tests is shown in Figure 4b.

Algorithm: `LineDetection`

Input: Intensity edges of an image
Output: A set of straight line segments

1. *Edge labeling:* For each unlabeled edge pixel
   (a) Track edge to find the rest of the connected edge points and label them, stopping if a junction point is encountered
   (b) Eliminate isolated edges and spurs that are below the minimum length
2. *Line segmenting:* For each labeled edge
   (a) Create a virtual straight line by connecting the start and end points of labeled edge
   (b) Calculate the deviation (perpendicular distance to the virtual line) of each point on the labeled edge
   (c) Divide the virtual line in half at the point of maximum deviation if the maximum deviation is greater than a threshold $d_{allowed}$
   (d) Repeat above process until the maximum deviation of all the line segments is less than $d_{allowed}$
3. *Line merging:* For each line segment
   (a) Merge with another line segment if the maximum tolerated angle deviation between them and the maximum distance between their end points are within a limit

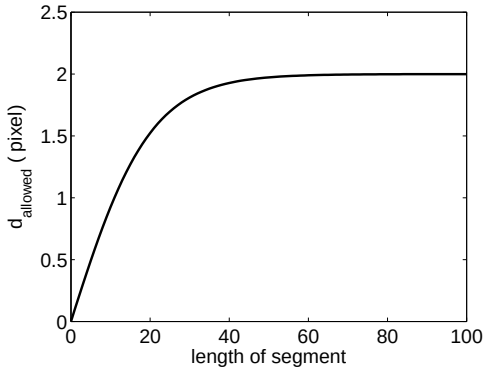


Figure 2. The half-sigmoid function of $d_{allowed}$, which makes the threshold dependent upon the length of the segment. $\delta = 2$ and $\beta = 10$.

2.2. Concavity

In many environments doors are receded into the wall, creating a concave shape for the doorway. A simplified con-

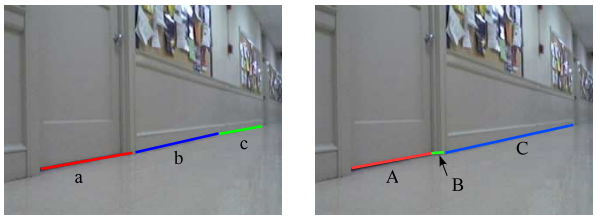
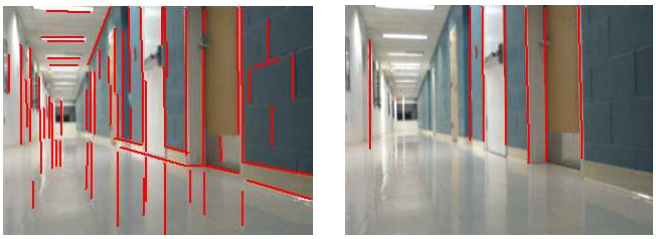


Figure 3. Modified Kovesi line segment detection algorithm. LEFT: The original algorithm mistakenly absorbs the door recession into segment a, and it mistakenly divides the single wall/floor line into b and c. RIGHT: With our modification, the recession is detected separately as segment B, and the segments b and c are correctly detected as a single segment C. Note that the recession segment B is important for the concavity test described in Section 2.2.



(a) (b)

Figure 4. (a) Line segments detected by Canny edge detector followed by line detection. (b) Candidate door segments retained after applying multiple tests to the segments.

cave door structure is illustrated in Figure 5, leading to two observations regarding the intensity edges:

- A slim "U" exists consisting of two vertical lines (the door edge and jamb) and one horizontal line (between the door frame and floor); and
- The bottom edge of the door is slightly recessed from the wall/floor edge.

Let (x, y) be a pixel on the line formed by extending the wall/floor boundary in front of the door (the dashed line in the figure), and let (x, y_b) be the pixel on the bottom of the door in the same column of the image. We test the recession of the door as follows:

$$H_{rec}(\omega) = (\tau_{\min} < y - y_b < \tau_{\max}), \qquad (4)$$

where $\tau_{\min} = 2$ and $\tau_{\max} = 10$ pixels, and $\omega \in \{L, R\}$ indicates whether the wall/floor boundary was extended from the left or the right of the door. The predicate H_U tests the presence of the slim "U" by looking for a vertical line adjacent to the door and an adjoining horizontal line near the bottom.

It may be the case that both left and right sides of the wall/floor boundary are not visible due to the location of the camera or an occluding object (e.g., a cabinet); or the slim

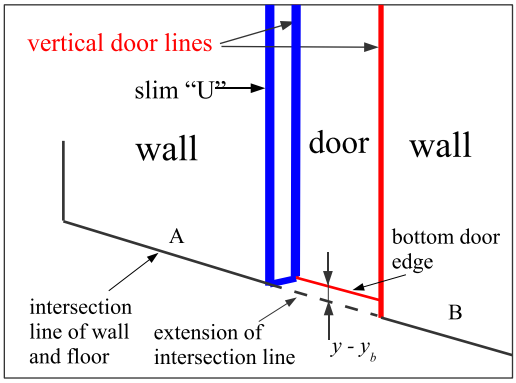


Figure 5. A concave door exhibits a slim "U" to its side, as well as a recession of the bottom edge. This geometry yields a concavity test which is an important cue for detecting doors.

"U" itself may be undetectable due to the oblique viewing angle. To handle this complication, the bottom edge of the door is determined as the line connecting the endpoints of two vertical lines separated by a minimum distance. Then the recession test is applied with respect to all non-vertical line segments below the vanishing point, and the slim "U" test is applied to the side of the edge. A door is declared if at least two of the three tests ($H_{rec}(L)$, $H_{rec}(R)$, and H_U) succeed. Note that in practice there is not enough information to use this concavity procedure if the robot directly faces the door, a situation which is detected by measuring the angle of the wall/floor boundary.

2.3. Gap below the door

Almost without exception, doors are constructed with a gap below them to avoid unnecessary friction with the floor as they open or close. As a result, when the light in the room is on, the area under the door tends to be brighter than the immediate surroundings, whereas if the light is off, then the area tends to be darker. In either case this piece of evidence, which is often just a few pixels in height, provides a surprisingly vital cue to the presence of the door, which is illustrated in Figure 6. In particular, this phenomenon is important for disambiguating the bottom door edge from the wall/floor edge. For each pixel along the bottom door edge, we compute the minimum and maximum intensities in the surrounding vertical profile. If one of these extrema is above a threshold and located near the door edge, then the pixel votes for the presence of a door. A majority vote among the pixels is taken.

2.4. Color

In many environments, doors and the surrounding walls have different colors. Rather than store a single mean color of the walls, we store a color histogram to enable the capturing of richly textured surfaces. The presence of a door is

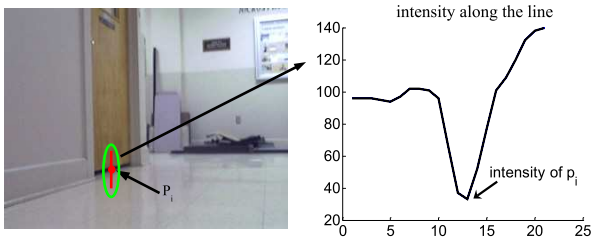


Figure 6. LEFT: An image of a door. RIGHT: The intensity profile of a vertical slice around the bottom edge. The dark region caused by the shadow of the door indicates the presence of the door. (Alternatively, if the light in the room were on, a bright region would indicate the door's presence.)

Figure 7. LEFT: A door with a kick plate. RIGHT: The segmentation of the image using the method of Felzenszwalb *et al*. [5]. The kick plate is the olive green region at the bottom of the door.

detected by thresholding the histogram intersection [2] between the wall color model ϕ_{wall} and the color histogram ϕ_{door} computed between two vertical line segments:

$$\frac{\sum_{i=1}^{M} \min(\phi_{door}[i], \phi_{wall}[i])}{\sum_{i=1}^{M} \phi_{door}[i]} > \tau_{color},$$

where $\phi_{door}[i]$ and $\phi_{wall}[i]$ are the numbers of pixels in the ith bin of the candidate door and wall histograms, respectively, and M is the number of bins. Because our present work is focused upon a single image, we assume that the wall histogram is available from an additional computation. We use the HSV (hue, saturation, value) color space because of its insensitivity to illumination changes compared with RGB.

2.5. Kick plate

Some doors have kick plates near the bottom which provide an additional cue for door detection. The image is first segmented using the method of Felzenszwalb *et al*. [5], as shown in Figure 7, and a region R in the segmented image is considered as a kick plate if the following tests succeed:

- The region R is located between two vertical lines $x = x_{left}$ and $x = x_{right}$, $x_{left} < x_{right}$:

$$\begin{aligned} \min_x\{(x,y) \in R\} &\geq x_{left} \\ \max_x\{(x,y) \in R\} &\leq x_{right} \end{aligned}$$

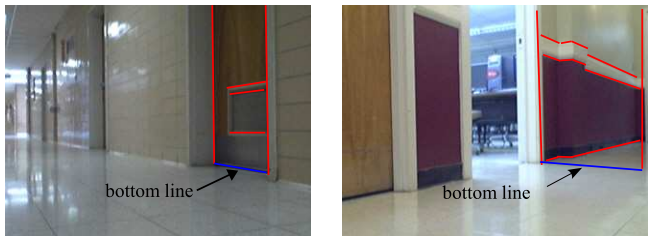


Figure 8. LEFT: The bottom line of a door or wall/floor boundary intersects the vanishing point. RIGHT: In contrast, a distracting line caused by shadows does not.

- The bottom of R is near the bottom of the two vertical lines
- The width and height of R are within a specified range.

Kick plates are visible in a number of doors.

2.6. Texture

The bottom part of the door is usually untextured, whereas areas of the wall and occluding objects often contain significant texture. We compute the texture energy by summing the magnitude of the gradient in the region A between two vertical lines and below the vanishing point:

$$\frac{1}{|A|} \sum_{p \in A} |\nabla I(p)| < \tau_{tex}, \tag{5}$$

where $\nabla I(i)$ is the intensity gradient of pixel p in the region, and $|A|$ is the number of pixels in the region.

2.7. Vanishing point

The vanishing point provides an additional test, as shown in Figure 8. The vanishing point is computed as the mean of the intersection of pairs of non-vertical lines:

$$\begin{bmatrix} wv_x \\ wv_y \\ w \end{bmatrix} = \sum_{i,j} \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} \times \begin{bmatrix} a_j \\ b_j \\ c_j \end{bmatrix}, \tag{6}$$

where the sum is over all pairs of lines i and j, each line is described by an equation $ax + by + c = 0$, $\times$ denotes the cross product, and the result is expressed in homogeneous coordinates. The vanishing point is determined by dividing by the scaling factor: $[\, v_x \quad v_y \,]^T$. If at least three lines on a door candidate pass near the vanishing point, then the test succeeds.

3. Adaboost training

Adaboost [6] is an algorithm to combine multiple weak classifiers into a strong classifier. As long as each weak classifier performs with at least 50% success, and the errors

of the different classifiers are independent, then the algorithm is able to improve upon the error rate by optimally selecting the weights for the weak classifiers. In our system, we use five weak classifiers: (1) concavity, (2) intensity changes along the bottom line, (3) color difference from the wall, (4) texture energy, and (5) intersection of the bottom line with the vanishing point.

Let h_n be the nth weak classifier, and let $y = h_n(x)$ be the output of the classifier to input x. In our case, x is the image and y is a binary label indicating whether a door was detected by the weak classifier. The strong classifier is given by a weighted sum of the weak classifiers:

$$\Psi(x) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n h_n(x)\right), \tag{7}$$

where α_n is the scalar weight found by AdaBoost indicating the importance of the weak classifier h_n, and $N = 5$. The weights are determined in an iterative manner according to

$$\alpha_n = \frac{1}{2}\left(\ln \frac{1-\varepsilon_n}{\varepsilon_n}\right), \tag{8}$$

where the error ε_n is given by

$$\varepsilon_n = Pr_{i\sim D_n}[h_n(x_i) \neq y_i] = \sum_{i:h_n(x_i)\neq y_i} D_n(i). \tag{9}$$

In this equation the output $y_i \in \{-1, +1\}$ is the ground truth for the training set, and $D_n(i)$ is the weight assigned to the ith training example on round n.

4. Experimental results

To test the performance of the system, a database of 309 door images was collected in twenty different buildings exhibiting a wide variety of visual characteristics. The images were taken by an inexpensive Logitech QuickCam Pro 4000 mounted 30 cm above the floor on an ActivMedia Pioneer P3AT mobile robot. Of these images, 100 were used for training the algorithm. On the remaining 209 images used for testing, the algorithm detects 90% of the doors with a false positive rate of 0.05 non-doors on average per image. The algorithm was implemented in Visual C++ and runs at a speed of 5 frame/s on a 1.6 GHz Dell Inspiron 700m laptop computer.

Figure 9 shows some typical doors detected by our system. As can be seen, our algorithm is capable of detecting doors in the hallway under different illumination conditions and different viewpoints, with either the same color as the wall or a different color, even in a cluttered environment. Note that in the second and third images in the first row, the algorithm successfully excludes a vending machine and a cabinet. From the ROC curves of the individual weak classifiers shown in Figure 10, it is clear that a significant

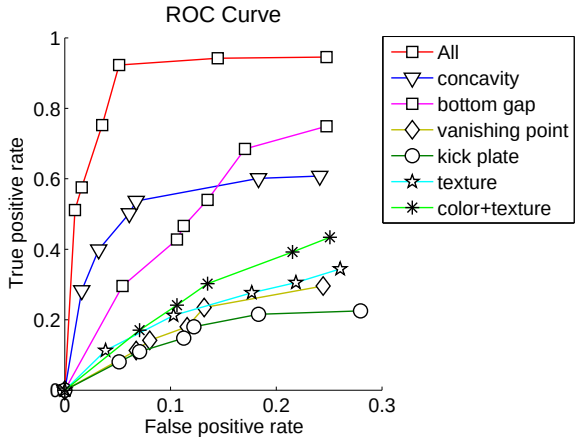


Figure 10. ROC curve showing the performance of our algorithm compared with the performance of single-cue detectors.

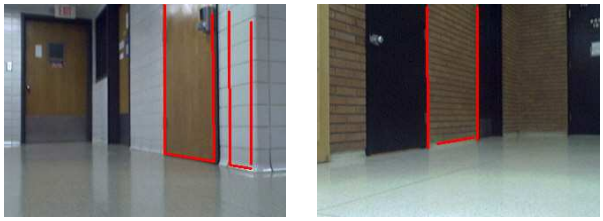


Figure 11. Some errors of the algorithm. LEFT: One door is successfully detected, but another is missed due to lack of contrast along its bottom edge coupled with strong reflections on the floor; in addition a false positive occurs because of distracting horizontal edges. RIGHT: A dark door that is flush with the wall fails the concavity and bottom gap tests and hence is missed, while edges on the wall are erroneously detected.

improvement is achieved by combining the classifiers in the Adaboost framework. Moreover, the features of concavity and gap below the door play an extremely important role compared with other features. Of course, the system is not perfect, and some errors are shown in Figure 11 for completeness.

To demonstrate the utility of the algorithm, we used the robot shown in Figure 1 equipped with two webcams with diverging optical axes. As the robot moved down a corridor, doors were detected on both sides of the hallway by the algorithm by processing the images on-line. Doors were tracked from frame to frame by a local search procedure employing a constant velocity assumption. Doors that were not repeatedly detected a certain number of image frames were regarded as false positives and discarded.

Figure 12 shows the results of ten trials in which the robot was manually driven along approximately the same path at a speed of 0.2 m/s down a 40×15 m corridor with a 90-degree turn. Of the 22 doors in the corridor, 20 of them were detected with 100% accuracy, that is, ten detections out of ten trials. However, one door (a in the figure) was

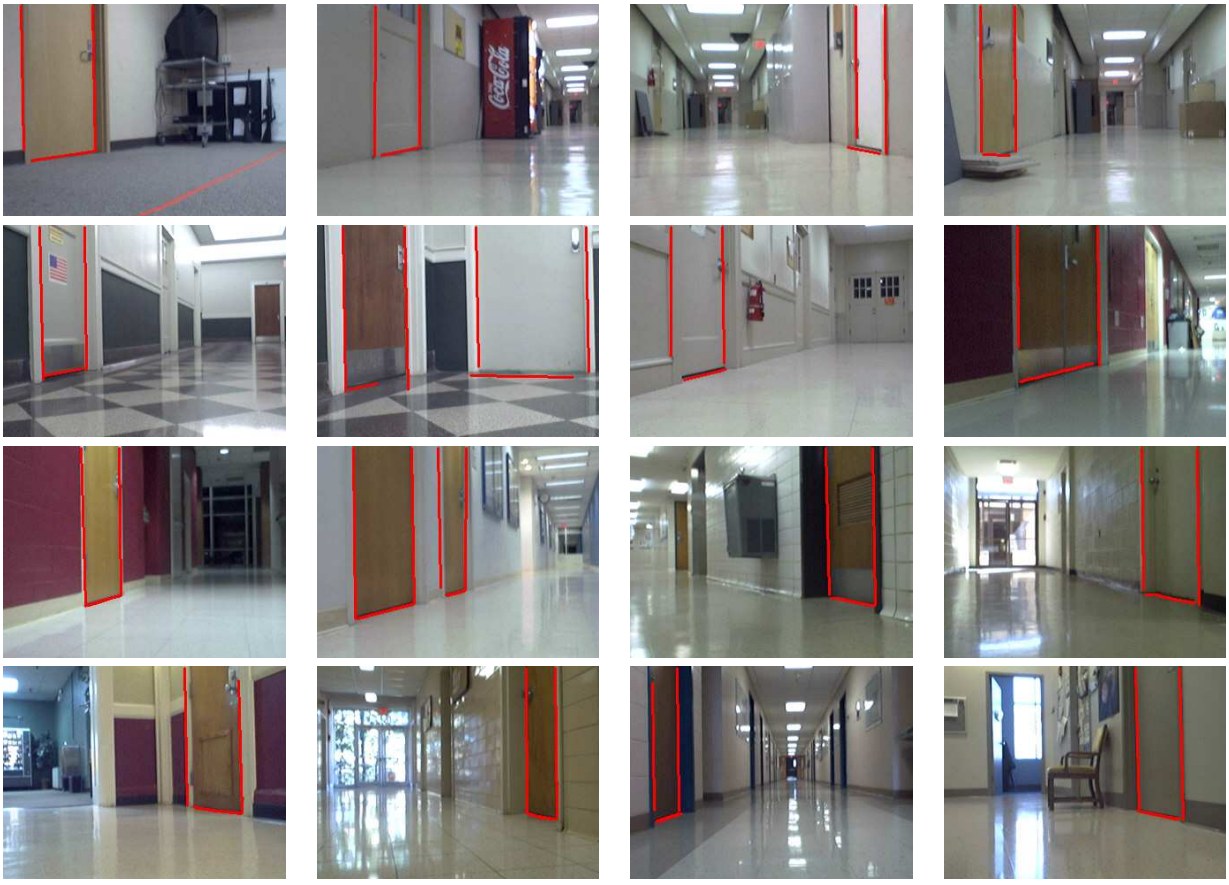


Figure 9. Examples of doors successfully detected by our algorithm. Note the variety of door widths, door and wall colors, relative pose of the door with respect to the robot, floor texture, and lighting conditions. Distant doors are not considered by the algorithm.

detected only 70% of the time due to an occluding water fountain, and one door (b) was detected 80% of the time because it is flush with the wall. In 60% of the trials, no false positives were detected in the entire corridor, although a convexity in the wall was erroneously detected as a door in the remaining four trials. Overall, the detection rate was 97.7% with a false positive rate of 0.008 per meter driven.

5. Conclusion and future work

We have presented a vision-based door detection algorithm for robot navigation using an uncalibrated camera. The doors are detected by a camera mounted on a mobile robot, from which low vantage point the lintel is often not visible. Door candidates are first sought by detecting the vertical lines that form the door frame, and then by applying constraints such as size, direction, and distance between segments. Within these door candidates, several features are measured. The algorithm augments standard features such as color, texture, and vertical intensity edges with novel geometric features based on the concavity of the door and the gap below the bottom door edge. The features are combined in an Adaboost framework to enable the algorithm to operate even in the absence of some cues. Tested on a large database exhibiting a wide variety of environmental conditions and viewpoints, the algorithm achieves more than 90% detection rate with a low false positive rate. The approach is suitable for real-time mobile robot applications using an off-the-shelf camera, and preliminary experiments demonstrate the success of the technique.

There is plenty of room for future work in this area. First, the color model of the walls should be learned automatically as the robot drives down the corridor. Secondly, the algorithm at present only detects closed doors. To detect open doors, the algorithm could be modified to observe the motion parallax of features inside the room that are visible when the door is open. More generally, additional features such as motion, door knobs, or shape, can be incorporated into the Adaboost framework to increase performance, and the interdependence between adjacent doors can be taken into consideration as well. In addition, calibration of the camera, along with 3D line estimation, would enable pose and distance measurements to facilitate the building of a geometric map. Ultimately, our goal is to integrate the al-

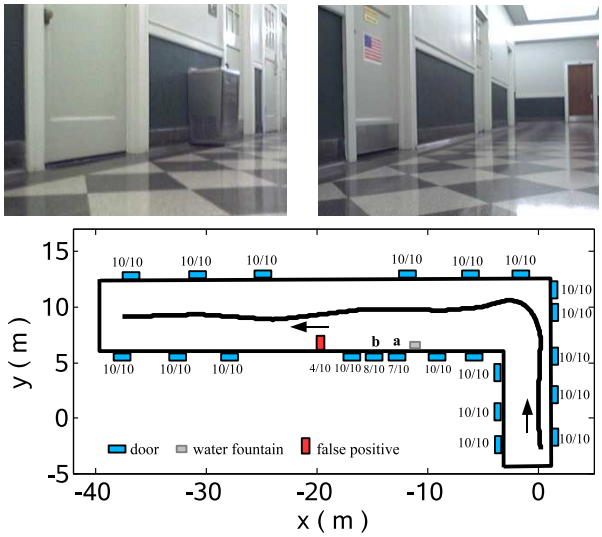


Figure 12. Two additional examples. TOP: Two images of the hallway environment near our laboratory. BOTTOM: 2D plan view of the hallway. Beside each door is indicated the number of detections / total number of trials. 100% detection rate is achieved for all but two doors.

gorithm into a complete navigation system that is able to map an environment, drive down a corridor, and turn into a specified room.

Acknowledgment

This work was supported by a Ph.D. fellowship from the National Institute for Medical Informatics, Medical Media Lab.

References

[1] D. Anguelov, D. Koller, E. Parker, and S. Thrun. Detecting and modeling doors with mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.

[2] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 232–237, June 1998.

[3] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[4] G. Cicirelli, T. D'Orazio, and A. Distante. Target recognition by components for mobile robot navigation. *Journal of Experimental & Theoretical Artificial Intelligence*, 15(3):281–297, 2003.

[5] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[6] Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, Sept. 1999.

[7] D. Kim and R. Nevatia. A method for recognition and localization of generic objects for indoor navigation. In *ARPA Image Understanding Workshop*, 1994.

[8] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[9] I. Monasterio, E. Lazkano, I. Rano, and B. Sierra. Learning to traverse doors using visual information. *Mathematics and Computers in Simulation*, 60(3):347–356, Sept. 2002.

[10] R. Munoz-Salinas, E. Aguirre, M. Garcia-Silvente, and A. Gonzalez. Door-detection using computer vision and fuzzy logic. In *Proceedings of the 6th WSEAS International Conference on Mathematical Methods & Computational Techniques in Electrical Engineering*, 2004.

[11] I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish: An office-navigating robot. *AI Magazine*, 16(2):53–60, 1995.

[12] M. Rous, H. Lupschen, and K.-F. Kraiss. Vision-based indoor scene analysis for natural landmark detection. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[13] S. A. Stoeter, F. L. Mauff, and N. P. Papanikolopoulos. Real-time door detection in cluttered environments. In *Proceedings of the 15th IEEE International Symposium on Intelligent Control*, 2000.