Visual Detection of Lintel-Occluded Doors by Integrating Multiple Cues Using Data-Driven MCMC

Zhichao Chen, Yinxiao Li, and Stanley T. Birchfield
Electrical and Computer Engineering Department, Clemson University
Clemson, South Carolina 29634
Email: {zhichac, yinxial, stb}@clemson.edu

***Abstract*— We present an algorithm to detect doors in images. The key to the algorithm's success is its fusion of multiple visual cues, including standard cues (color, texture, and intensity edges) as well as several novel ones (concavity, the kick plate, the vanishing point, and the intensity profile of the gap below the door). We use the Adaboost algorithm to determine the linear weighting of the various cues. Formulated as a maximum a posteriori probability (MAP) problem, a multi-cue functional is minimized by a data-driven Markov Chain Monte Carlo (DDMCMC) process that arrives at a solution that is shown empirically to be near the global minimum. Intensity edge information is used in the importance probability distribution to drive the Markov chain dynamics in order to achieve a speedup of several orders of magnitude over traditional jump diffusion methods. Unlike previous approaches, the algorithm does not rely upon range information and yet is able to handle complex environments irrespective of camera pose, lighting conditions, wall or door color, and reflections. Moreover, the algorithm is designed to detect doors for which the lintel is occluded, which often occurs when the camera on a mobile robot is low to the ground. The versatility of the algorithm is tested on a large database of images collected in a wide variety of conditions, on which it achieves approximately 90% detection rate with a low false positive rate. Versions of the algorithm are shown for calibrated and uncalibrated camera systems. Additional experiments demonstrate the suitability of the algorithm for near-real-time applications using a mobile robot equipped with off-the-shelf cameras.**

I. Introduction

Over the past decade much progress has been made toward solving the problem of simultaneous localization and mapping (SLAM) [29]. Primarily, the focus has been upon recovering the geometric structure of the scene by estimating the Euclidean coordinates of a number of points, irrespective of the identity of the object to which they belong. Building interiors, however, are designed to be highly structured environments, in which the various components (e.g., walls, floors, doors, and corridors) are placed in a highly predictable manner in relation to one another. Detecting these semantically meaningful components is extremely important for mobile robot navigation in such environments. Doors, in particular, are important landmarks for navigation because they occur frequently, provide the entrance and exit points of rooms, and are intentionally designed to provide stable and semantically meaningful structures for determining the location of an observer.

Much of the previous work on door detection has relied upon 3D range information available from sonar, laser, or stereo vision sensors [16], [24], [27], [1]. We are interested, however, in using off-the-shelf cameras for detecting doors, 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 textured/untextured floors, similarly colored walls and doors, low-contrast edges, bright reflections, variable lighting conditions, and changing robot pose, as shown in the figure.

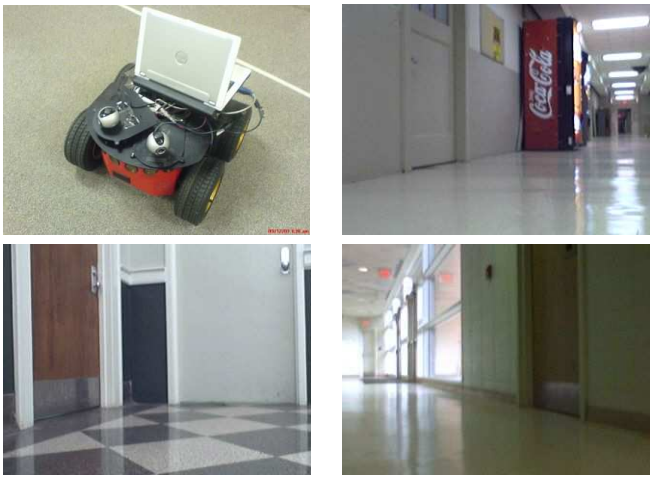

Fig. 1. TOP-LEFT: Our robot is equipped with two non-overlapping off-the-shelf webcams, mounted on top.TOP-RIGHT: An image taken by one of the cameras, showing a door whose color is the same as the surrounding wall and whose lintel is not visible. BOTTOM: Two additional examples, showing doors at drastically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

In the computer vision community, the most popular approach to object detection is to train a classifier to distinguish

between positive and negative examples of rectangular image patches, an approach which has been applied successfully to objects such as faces and cars with clearly identifiable local features [26], [30], [9]. Doors, however, present a challenge for such an approach because they often have no locally unique features to indicate their presence. Rather, a door is observed, it seems, by the interplay of a number of different geometric and photometric cues whose spatial relation to one another is not pixel-based due to the unknown pose.

As a result, in this paper we present a solution to the problem based upon combining multiple cues. Our approach augments standard features such as color, texture, and vertical intensity edges with novel geometric features such as the concavity of the door and the gap below the bottom door edge. The approach builds on our previous research [3] by incorporating these features into a maximum a posteriori (MAP) framework. Adaboost [7] is used to compute the linear weighting of the different features, and a Data Driven Markov Chain Monte Carlo (DDMCMC) technique is used to explore the solution space [15]. By incorporating intensity edges in the importance proposal distribution, a significant speedup is achieved in comparison to the traditional jump diffusion methods. Experimental results on a large database of images show the versatility of the algorithm in detecting doors in a variety of challenging environmental conditions. The algorithm is incorporated into a nearly real-time system that detects doors as the robot drives down a corridor. In addition, a modified version of the algorithm is presented for uncalibrated camera systems common in non-robotics applications.

II. Previous work

Many door detection systems use only range information, without cameras. Early work involved sonar sensors [24], while more recent work utilizes laser range finders [2], [20]. In these approaches, the detector requires the door plane to be distinguishable from the wall plane either because the door is recessed, a molding protrudes around the door, or the door is slightly open. Thus, if a door is completely flush with the wall, such detectors will be unable to find it.

Perhaps the most popular approach to door detection involves combining range sensors with vision. Kim and Nevatia [16] extract both vertical (post) and horizontal (lintel) line segments from an image, then analyze whether these segments meet minimum length and height restrictions, verifying door candidates by a 3D trinocular stereo system. Stoeter et al. [27] extract vertical lines in the image using the Sobel edge detector followed by morphological noise removing, then combine the resulting lines with range information from a ring of sonars to detect doors. In contrast, the system of Anguelov et al. [1] does not use intensity edges at all but rather the colors along a single scan of an omnidirectional image combined with a laser range finder. Doors are first detected by observing their motion over time (i.e., whether an open door later becomes closed, or vice versa) in order to learn a global mean door color. Doors are then detected in an expectation-maximization framework by combining the motion information with the door width (as estimated by the laser range finder) and the similarity of image data to the learned door color. This approach assumes that the doors are all similarly colored, that the mean color of the doors and walls are significantly different from each other, and that enough doors change state to learn the appropriate color models. Another piece of interesting research is that of Hensler et al. [12], who augment our recent vision-only algorithm [3] with a laser range finder to estimate the concavity and width of the door, which are then combined with other image-based features.

A few researchers have focused upon the much more difficult problem of detecting doors using vision alone, without range information. Monasterio et al. [21] detect intensity edges corresponding to the posts and then classify the scene as a door if the column between the edges is wider than a certain number of pixels, an approach that assumes a particular orientation and distance of the robot to the door. Similarly, Munoz-Salinas et al. [22] apply fuzzy logic to establish the membership degree of an intensity pattern in a fuzzy set using horizontal and vertical line segments. Jauregie et al. [13] propose an approach to identify the door by recognizing the door edges using color information. Rous et al. [25] 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 mean 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. Murillo et al. [23] present a color-based approach to door detection that works well when the door color is uniform throughout the environment.

While these previous systems have achieved some success, no vision-only system has yet demonstrated the capability of handling a variety of challenging environmental conditions (changing pose, similarly colored doors and walls, strong reflections, and so forth) in the presence of the lintel-occlusion that often occurs when the camera is low to the ground and the door is nearby.

III. Bayesian formulation

In this section, we formulate the problem in a Bayesian framework. Assuming that the camera is oriented so that the image plane is perpendicular to the floor, a door $\mathbf{d} = (x_l, x_r, y_{lt}, y_{lb}, y_{rt}, y_{rb})$ in an image I is represented using the six coordinates of the four end points of two vertical line segments ℓ_l and ℓ_r, as shown in Figure 2. Our goal is to find the number of doors in the image, along with the location of each door. Invoking the first-order Markov assumption, and letting $\mathbf{d}_1, \ldots, \mathbf{d}_k$ to represent the sequence of k doors from left to right in the image, the goal is

expressed probabilistically as maximizing

$$p(k, \mathbf{d}_1, \ldots, \mathbf{d}_k|I) = p(\mathbf{d}_1|I)p(k|I)\prod_{i=2}^{k} p(\mathbf{d}_i|I, \mathbf{d}_{i-1}). \quad (1)$$

A straightforward approach to solve this problem would be to apply reversible jump Markov chain Monte Carlo (RJMCMC) [8] to simultaneously estimate the number k and the joint configuration of all the doors. Such an approach, however, is notoriously computationally intensive. In contrast, the formulation above suggests that a simpler approach might be sufficient: search the image from left to right (or vice versa) in a sequential manner to find the doors one at a time. Since the coupling between the doors can be modeled as a prior on location, the primary problem becomes to find a single door given the image by maximizing $p(\mathbf{d}|I)$, which, according to Bayes' rule, is

$$p(\mathbf{d}|I) \propto p(I|\mathbf{d})p(\mathbf{d}). \quad (2)$$

Following the sum rule of Kittler and colleagues [18], [17], [28], this is similar to maximizing the following functional:

$$E(\mathbf{d}) \quad = \quad \Psi_{data}(I|\mathbf{d}) + \Psi_{prior}(\mathbf{d}), \quad (3)$$

where Ψ_{data} and Ψ_{prior} are combinations of expert opinions for the likelihood and prior, respectively.

IV. Prior Model

The prior model captures information about the expected door configuration without considering image evidence relevant to the particular door being found. We formulate the prior of a door candidate as a linear combination of various pieces of evidence:

$$\Psi_{prior}(\mathbf{d}) = \sum_{i=1}^{N_{prior}} \alpha_i f_i(\mathbf{d}), \quad (4)$$

where N_{prior} is the number of tests aggregated, and α_i is the weight that governs the relative importance of the ith test, $\sum_{i=1}^{N_{prior}} \alpha_i = 1$. All the tests f_i are normalized to yield values between 0 and 1.

Our approach employs two tests ($N_{prior} = 2$). First, we compare the width w of the door in world coordinates with the expected width $\hat{w}$ of real doors:

$$f_1(\mathbf{d}) = \exp\left\{-\frac{(w-\hat{w})^2}{2\sigma_w^2}\right\}. \quad (5)$$

Several real world doors were measured in order to determine the expected door width and standard deviation, leading to $\hat{w} = 0.96$ m, and $\sigma_w = 0.08$ m. To enable the measuring of the door width in world coordinates, the camera is first calibrated by capturing an image of a piece of paper of known dimensions placed on the floor. The normalized Direct Linear Transformation (DLT) [10] is used to calculate a homography between the floor plane and the image plane, which enables the image to world transformation of the two points at the bottom of the door.

Secondly, we expect the height of the two vertical door edges (in world coordinates) to be similar to the height of a standard door. However, since the lintel (top) of the door could be occluded, it may be impossible to measure the height of these two edges. Instead, we measure the height h_h of the hinge edge between the ground plane and the horizontal plane passing through the optical center of the camera. If the image plane is perpendicular to the ground plane, the ratio $\rho = w/h_h$ of the door width to this height should be constant, leading to

$$f_2(\mathbf{d}) = \exp\left\{-\frac{(\rho-\hat{\rho})^2}{2\sigma_\rho^2}\right\}, \quad (6)$$

where $\hat{\rho}$ is the ratio of a standard door. Based on measurements of 25 real doors, and a camera height of 0.32 m above the ground, we obtain $\hat{\rho} = 3.0$ and $\sigma_\rho = 0.2$.

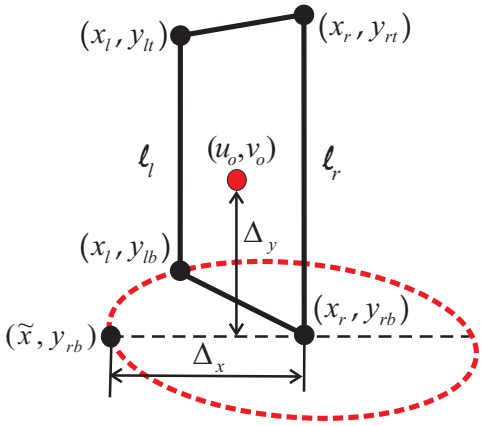

Fig. 2. Door configuration: A door candidate can be described using two vertical lines ℓ_l and ℓ_r in an image, defined by four end points (x_l, y_{lt}), (x_l, y_{lb}), (x_r, y_{rt}) and (x_r, y_{rb}). Assuming that the door rotates around the right vertical line, the bottom left corner of the door is located at $(\tilde{x}, y_{rb})$ when the door is parallel to the image plane. Δ_x is the width of the aligned door in image coordinates, while Δ_y is the height from the bottom of the door to the horizontal line passing through the principal point (u_0, v_0) of the image.

Figure 2 illustrates the computation of ρ. Given the homography H between the image and world coordinate systems, $H^{-1}\begin{bmatrix} x_r & y_{rb} & 1 \end{bmatrix}^T$ yields the homogeneous world coordinates of the intersection of the door's rotation axis (assuming that it rotates around its right edge) and the ground plane. Let C be the 3×3 homogeneous matrix representing, in world coordinates, the circle centered at this location with radius equal to the width of the door. Then $C' = H^{-T}CH^{-1}$ describes the ellipse in the image tracing the coordinates of (x_l, y_{lb}), the bottom of the left line, as the door rotates about its hinge. For any image point along this ellipse, the world width of the door is, of course, the same. Thus, the point $(\tilde{x}, y_{rb})$, which is the intersection of C' with the line $y = y_{rb}$, indicates the image location of the bottom of the door that would result if the door were parallel to the image plane. (Note that in most cases the ellipse will not be aligned with the image axes, so that the line $y = y_{rb}$ is *not* the major axis of the ellipse.) Assuming unity aspect ratio of the image sensor, this yields the ratio as

$$\rho = \frac{w}{h_h} = \frac{\Delta_x}{\Delta_y} = \frac{x_r - \tilde{x}}{y_{rb} - v_0}. \quad (7)$$

V. Data model

Similar to the prior model, the data model is formulated as a linear combination of evidences:

$$\Psi_{data}(I|\mathbf{d}) = \sum_{j=1}^{N_{data}} \beta_j g_j(I|\mathbf{d}), \quad (8)$$

where the weights β_j are chosen based on Adaboost, $\sum_{j=1}^{N_{data}} \beta_j = 1$, and all the tests g_j are normalized between 0 and 1. We now describe the $N_{data} = 9$ tests.

A. Image gradient along edges (g_1)

Perhaps the most distinguishing visual characteristic of doors in images is the change in intensity that usually accompanies the sides of the door. As a result, we measure

$$g_1(I|\mathbf{d}) = 1 - \exp\left\{ - \sum_{\mathbf{x}\in\mathcal{R}_e} \left| (\nabla I(\mathbf{x}))^T \mathbf{n}(\mathbf{x}) \right| \right\}, \quad (9)$$

where $\mathbf{x}$ is a pixel on the door edge, $\nabla I(\mathbf{x})$ is the 2D image gradient at the pixel, and $\mathbf{n}(\mathbf{x})$ is the normal to the door edge at the pixel. The set $\mathcal{R}_e$ of pixels includes the left and right edges of the door, the bottom edge, and the top edge (if $x_{lt} \neq 0$ and $x_{rt} \neq 0$). The gradient of the image is computed using a 3-tap separable Gaussian derivative filter with a standard deviation of 1 pixel.

B. Placement of top and bottom edges (g_2 and g_3)

After computing the gradient of the image, vertical and horizontal line segments are found by applying the Douglas-Peucker line fitting algorithm [5], as modified in [3] to use an adaptive threshold, to the output of the Canny edge detector. The horizontal segments between the two vertical door edges are compared to determine the topmost segment, which is then extended to intersect with the vertical edges, yielding expected values $\hat{y}_{lt}$ and $\hat{y}_{rt}$ for the top points of the two edges. If there is no such segment, then $\hat{y}_{lt} = \hat{y}_{rt} = 0$, indicating a lintel-occluded door. Similarly, the bottom endpoints of the vertical edges are compared with the line forming the boundary between the wall and the floor:

$$g_2(I|\mathbf{d}) = \exp\left\{ -\frac{(y_{lt} - \hat{y}_{lt})^2}{2\sigma_t^2} - \frac{(y_{rt} - \hat{y}_{rt})^2}{2\sigma_t^2} \right\} \quad (10)$$

$$g_3(I|\mathbf{d}) = \exp\left\{ -\frac{(y_{lb} - \hat{y}_{lb})^2}{2\sigma_b^2} - \frac{(y_{rb} - \hat{y}_{rb})^2}{2\sigma_b^2} \right\} \quad (11)$$

where $\hat{y}_{lb}$ and $\hat{y}_{rb}$ are the intersection of the wall-floor boundary with the two vertical edges, computed as in [19]. The standard deviations σ_t and σ_b are set to 5 pixels.

C. Color and texture (g_4 and g_5)

Color provides a helpful cue to distinguish the door from its surroundings. We use the Bhattacharyya coefficient [14]

$$g_4(I|\mathbf{d}) = \sum_{i=1}^{N_{col}} \sqrt{\phi(i)\phi_{wall}(i)} \quad (12)$$

to compare the normalized color histogram ϕ computed using the pixels inside the door parallelogram, and the normalized color histogram ϕ_{wall} that models the colors in the wall, where $N_{col} = 8^3$ is the number of bins in the histogram. The HSV (hue, saturation, value) color space is used due to its insensitivity to illumination changes compared with RGB. In our system the robot builds a model of the wall color automatically as it moves down the corridor, described later.

Although the top half of a door may contain windows, signs, or flyers, the bottom half of a door is often untextured. Texture is measured using the entropy of the normalized histogram of the image gradient magnitude and phase:

$$g_5(I|\mathbf{d}) = 1 - \frac{1}{\eta} \sum_{i=1}^{N_{tex}} \sum_{j=1}^{M_{tex}} \phi_{tex}(i,j) \log\left(\phi_{tex}(i,j)\right), \quad (13)$$

where i and j index the magnitude and phase, respectively, in the histogram ϕ_{tex}; $N_{tex} = 16$ and $M_{tex} = 8$ are the number of bins along the two directions of the histogram; and $\eta = \log(N_{tex}M_{tex})$ is the entropy of a uniform distribution, used to normalize the result. We use entropy because of its ability avoid being distracted by strong intensity edges that may occur near the bottom of the door, such as the boundary of the kick plate. Typically, such edges are fairly localized in magnitude and phase, thus resulting in low entropy.

D. Kick plate (g_6)

Fig. 3. Left: A door with a kick plate. Right: The segmentation of the image using the efficient minimum-spanning-tree algorithm [6], in which the kick plate is identifiable as the olive green region at the bottom of the door.

Some doors have kick plates near the bottom which provide an additional cue for door detection. The image is first segmented using the efficient minimum-spanning-tree algorithm [6], as shown in Figure 3. A region in the segmented image is considered as kick plate if it is located within the two vertical door lines and above the bottom of the door candidate, and if its height is about a quarter of the height of the door candidate. This results in

$$g_6(I|\mathbf{d}) = \exp\left\{ -\frac{(k_x - \hat{k}_x)^2}{2\sigma_{kx}^2} \right\} \exp\left\{ -\frac{(k_y - \hat{k}_y)^2}{2\sigma_{ky}^2} \right\}, \quad (14)$$

where (k_x, k_y) is the centroid of the kick plate, $\hat{k}_x = x_r - x_l$ is the x coordinate of the centroid of the door candidate, and $\hat{k}_y = \frac{1}{4}\Delta y$ is the expected y coordinate of the centroid of the kick plate, assuming that the height of the kick plate is approximately one half the distance from the bottom of the door to the principal point. We set $\sigma_{kx} = \frac{1}{4}(x_r - x_l)$ and $\sigma_{ky} = \frac{1}{4}\Delta y$. If no kick plate is detected, then $g_6(I|\mathbf{d}) = 0$.

E. Vanishing point (g_7)

The vanishing point provides an additional test, as shown in Figure 4. The vanishing point is computed as the mean of the intersection of pairs of non-vertical lines *inside* the door region:

$$\begin{bmatrix} wv_x \\ wv_y \\ w \end{bmatrix} = \frac{1}{N_v} \sum_{i,j} \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} \times \begin{bmatrix} a_j \\ b_j \\ c_j \end{bmatrix}, \quad (15)$$

where the sum is over all pairs of lines i and j, N_v is the number of such pairs, each line is described by an equation $a_i x + b_i y + c_i = 0$, $\times$ denotes the cross product, and the result is expressed in homogeneous coordinates. The vanishing point $\mathbf{v} = [\, v_x \quad v_y \,]^T$ is determined by dividing by the scaling factor w. Vanishing point consistency is measured by

$$g_7(I|\mathbf{d}) = \exp\left\{-\frac{(v_x - \hat{v}_x)^2}{2\sigma_{vx}^2}\right\} \exp\left\{-\frac{(v_y - \hat{v}_y)^2}{2\sigma_{vy}^2}\right\}, \quad (16)$$

where $\hat{v} = [\, \hat{v}_x \quad \hat{v}_y \,]^T$ is the vanishing point estimated as the mean of the intersection of pairs of non-vertical lines *outside* the door. We set σ_{vx} and σ_{vy} as $\frac{1}{10}$ of the width and height, respectively, of the image.

Fig. 4. The bottom line of a door intersects the vanishing point (left), while a distracting line caused by shadows does not (right).

F. Concavity (g_8)

In many environments doors recede into the wall, creating a concave shape for the doorway. A simplified concave door structure is illustrated in Figure 5, leading to two observations regarding the intensity edges. First, a slim "U" exists consisting of two vertical lines (the door edge and jamb) and one horizontal line (between the door frame and floor). Secondly, 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. Then we define

$$g_8(I|\mathbf{d}) = \exp\left\{-\frac{(w_u - \hat{w}_u)^2}{2\sigma_u^2} - \frac{(y - y_b - \Delta\hat{y}_{rec})^2}{2\sigma_r^2}\right\}, \quad (17)$$

where $\hat{w}_u = \frac{1}{4}(x_r - x_l)$, $\Delta\hat{y}_{rec} = 2$ pixels, $\sigma_u = 2$ pixels, and $\sigma_r = \frac{1}{4}(x_r - x_l)$, where $x_r - x_l$ is the width of the door in the image. The first factor tests the slim "U" by comparing its width w_u with an expected width $\hat{w}_u$, and the second factor tests the recession.

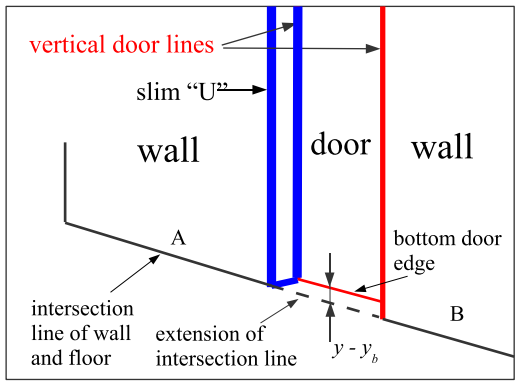

Fig. 5. A concave door exhibits a slim "U" to its side, as well as a recession of the bottom edge. These two geometric cues are exploited in the concavity measure.

G. Gap below the door (g_9)

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 due to the shadow. 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 and perform a test to ensure that the extremum value is near the center of the profile, and that the difference between the extremum value and the average of the surrounding values is above a threshold. The gap below the door is then measured as the ratio of the number of pixels N_g that satisfy the bottom gap test to the total number of pixels N_b along the bottom edge:

$$g_9(I|\mathbf{d}) = \frac{N_g}{N_b}. \quad (18)$$

VI. Door detection using DDMCMC

Markov Chain Monte Carlo (MCMC) is a stochastic algorithm for finding the extremum of a functional by discrete search of the state space. The sequence of states encountered during the search forms a Markov Chain, and the state with the best score is retained as the solution once the search terminates. At the heart of the algorithm is a jump from the current state $\mathbf{d}$ to the next state $\mathbf{d}'$ according to the proposal probability $q(\mathbf{d}'|\mathbf{d})$. Using the Metropolis-Hastings method [11], the new state is then accepted according to the probability

$$p(\mathbf{d} \to \mathbf{d}') = \min\left(1, \frac{E(\mathbf{d}')q(\mathbf{d}'|\mathbf{d})}{E(\mathbf{d})q(\mathbf{d}|\mathbf{d}')}\right), \quad (19)$$

where the equation corresponds to a maximizing of E and hence a maximum a posteriori (MAP) estimate of the door.

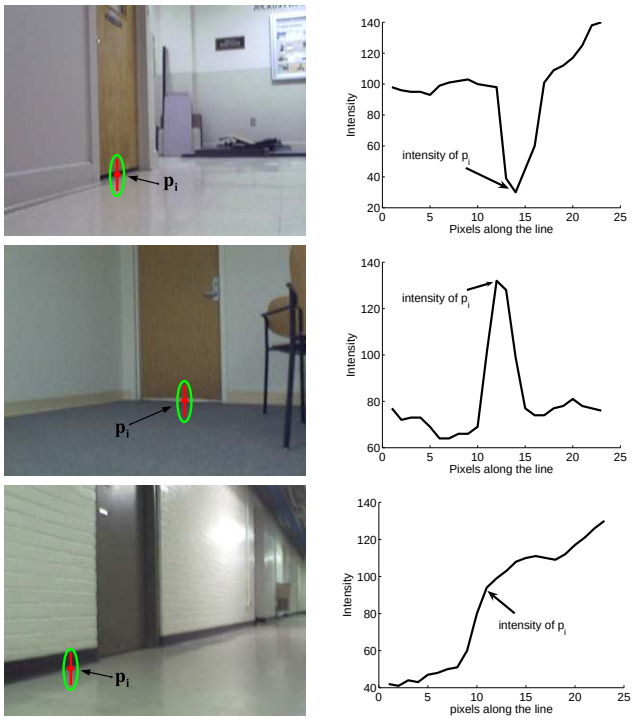

Fig. 6. The intensity profile of a vertical slice through the bottom door edge. LEFT: Images of three doors. TOP-RIGHT: The intensity profile of a vertical slice through the bottom edge, showing a dark trough caused by the shadow of the door. MIDDLE-RIGHT: Alternatively, if the light in the room is on, the presence of the door is evident from the bright peak. BOTTOM-RIGHT: However, if the vertical slice covers the wall-floor intersection, the intensity profile changes smoothly, and no sharp peak occurs.

The speed of MCMC depends critically on the design of the proposal probability distribution q. If a uniform distribution is used, then all parameters of the state vector are equally likely to change. With such a blind search, the Markov chain will experience exponential waiting time before each jump. Because we are interested in real-time applications, we turn to *data-driven* MCMC (DDMCMC) in order to speed up this computation. In DDMCMC, the proposal distribution $q(\mathbf{d}'|\mathbf{d})$ is replaced by $q(\mathbf{d}'|\mathbf{d}, I)$ so that image data may be taken into account when computing the next state.

Inspired by our previous work [3], we exploit the fact that doors are frequently aligned with intensity edges in the image. For DDMCMC, the proposal distribution is split according to the individual x and y elements as $q(\mathbf{d}'|\mathbf{d}, I) = q(x_l, x_r|I)q(y_{lt}, y_{lb}, y_{rt}, y_{rb}|x_l, x_r)$. The x coordinates are found by drawing from the distribution $q(x_l, x_r|I) = q(x_l|\ell_l^*)q(x_r|\ell_r^*)q(\ell_l^*, \ell_r^*)$, where $q(\ell_l^*, \ell_r^*)$ expresses that two vertical lines are drawn at random from the detected vertical line segments and ordered such that ℓ_l^* is to the left of ℓ_r^*, and the x coordinates are then drawn from Gaussian distributions centered at ℓ_l^* and ℓ_r^*, respectively.

For the y coordinates, we find the horizontal line segment closest to connecting the top of the two vertical lines ℓ_l and ℓ_r. We define the intersection of this line segment with the two vertical lines as y_{lt}^* as y_{rt}^*. If no such horizontal line segment can be found, then the door is assumed to be lintel-occluded, and $y_{lt}^* = y_{rt}^* = 0$. Similarly, we define the intersection of the wall-floor boundary with the two vertical lines as y_{lb}^* and y_{rb}^*. Then the y coordinates are found by drawing independently from Gaussian distributions centered at these locations.

Algorithm: `Door Detection`

Input: Single color image I
Output: List of detected doors $(k, \mathbf{d}_1, \ldots, \mathbf{d}_k)$

1) *Preprocessing:*
   a) Compute the Canny edges of I
   b) Apply modified Douglas-Peucker algorithm [3] to fit straight line segments to the Canny edges
   c) Classify the line segments as vertical or horizontal
   d) Segment I using the graph-based algorithm [6]
2) *Door detection:*
   a) Set $k \leftarrow 0$
   b) Set `done` $\leftarrow$ FALSE
   c) Repeat while `done` is FALSE
      i) Set $\phi^* \leftarrow 0$
      ii) Repeat until convergence:
         A) Randomly select two adjacent vertical line segments ℓ_l and ℓ_r
         B) Intersect the nearby horizontal line segments with ℓ_l and ℓ_r to get configuration
         C) Compute the perturbed configuration $\mathbf{d} = (x_l, x_r, y_{lt}, y_{lb}, y_{rt}, y_{rb})$ by perturbing the points using a Gaussian distribution: $x_i \leftarrow x_i + \eta_i$ and $y_{ij} \leftarrow y_{ij} + \eta_{ij}$, where $\eta_i \sim \mathcal{N}(0, \sigma_x^2)$ and $\eta_{ij} \sim \mathcal{N}(0, \sigma_y^2)$ for $i \in \{l, r\}$ and $j \in \{t, b\}$
         D) Compute the energy $\phi \leftarrow E(\mathbf{d})$
         E) Compute $p \leftarrow \min\left(1, \frac{\phi}{\phi^* + \epsilon}\right)$
         F) With probability p, set $\mathbf{d}^* \leftarrow \mathbf{d}$ and $\phi^* \leftarrow \phi$
      iii) If $\phi^* > \tau$, then $k \leftarrow k+1$ and $\mathbf{d}_k \leftarrow \mathbf{d}^*$, else `done` $\leftarrow$ TRUE
   d) Return $(k, \mathbf{d}_1, \ldots, \mathbf{d}_k)$

Fig. 7. Pseudocode of the DDMCMC-based door detection algorithm.

Once a certain number of jumps have not yielded any increase in the score, the search stops, and the resulting state is considered a door if the score is above a threshold. The process then repeats to look for additional doors, excluding line segments near doors that have already been detected in the image from the distribution $q(\ell_l^*, \ell_r^*)$. When no more doors can be found, the process is complete. Pseudocode for the algorithm is given in Figure 7, where ϵ is a small number to avoid dividing by zero, $\sigma_x = 1$ and $\sigma_y = 3$ pixels.

VII. UNCALIBRATED CAMERAS

The only part of the algorithm that uses calibration information are the prior terms, described in Section IV. Removing these terms yields a system that requires no

calibration, which may be useful for non-robotic applications in which the height of the camera above the floor is not known. One issue with an uncalibrated system is estimating the door color. For a calibrated system, the robot is run along the corridor, and the width between vertical lines is compared with the expected door width in world coordinates to yield the likelihood, based on width alone, that the vertical lines arise from a door. These preliminary decisions are used to build a color model of the wall, which then is used in the color cue when the full system is run. For an uncalibrated system, however, we cannot do this. Instead we rely on the relative width of doors.

To detect relative door width, we rely on an assumption that doors in an environment are likely to have similar widths in the world. Given an image of two doors, 1D projective geometry can be used to compute their relative width. The scenario is illustrated in Figure 8, which shows a top-down view of two doors projected onto an image. In the 1D world coordinate system, the door posts lie at locations A, B, C, and D along the line containing the wall. In homogeneous coordinates, therefore, these posts lie at locations $(A, 1)$, $(B, 1)$, $(C, 1)$, and $(D, 1)$, and $(0, 1)$ is the point at infinity associated with the line containing the posts. In the image, we can detect the vertical lines associated with these edges, yielding horizontal image coordinates of a, b, c, and d. Now suppose that we also detect the vanishing point in the image and determine that its horizontal coordinate is v. Translating the origin of both coordinate systems to the leftmost edge, we can use the following two equations to relate a and v to their world coordinates: $H_{2\times 2}\begin{bmatrix}0 & 1\end{bmatrix}^T = \begin{bmatrix}0 & 1\end{bmatrix}^T$ and $H_{2\times 2}\begin{bmatrix}1 & 0\end{bmatrix}^T = \begin{bmatrix}v-a & 1\end{bmatrix}^T$, to solve for the 2×2 homography matrix $H_{2\times 2}$ that maps world coordinates to image coordinates:

$$H_{2\times 2} = \begin{bmatrix} v-a & 0 \\ 1 & 1 \end{bmatrix}. \tag{20}$$

Once $H_{2\times 2}$ has been calculated, it is easy to see that the homogeneous coordinates of the image projection of B are given by $H_{2\times 2}\begin{bmatrix}B-A & 1\end{bmatrix}^T = \begin{bmatrix}(B-A)(v-a) & B-A+1\end{bmatrix}^T$, so that the relationship between the world point B and the image point b are given by $b-a = \frac{(B-A)(v-a)}{B-A+1}$, since A and a are the origins of the two coordinate systems. Rearranging leads to an expression of the width of the first door:

$$\text{width}_1 = B - A = \frac{b-a}{v-b}. \tag{21}$$

Similarly, again using (20), the width of the second door is given by:

$$\text{width}_2 = D - C = \frac{d-a}{v-d} - \frac{c-a}{v-c} = \frac{(d-c)(v-a)}{(v-c)(v-d)}. \tag{22}$$

Now these widths are only computed up to an unknown scaling factor. Therefore, the important cue is not the absolute value of either width, but rather whether the widths are nearly equal to each other. After measuring the relative width of the two doors, the result is verified by checking the similarity of color between them, as well as the consistency of the cross ratio between a, b, c, and d as the robot moves, in order to verify that A, B, C, and D are collinear. If all of these tests pass, then the area between the two doors is considered as wall and is used to build the wall color model.

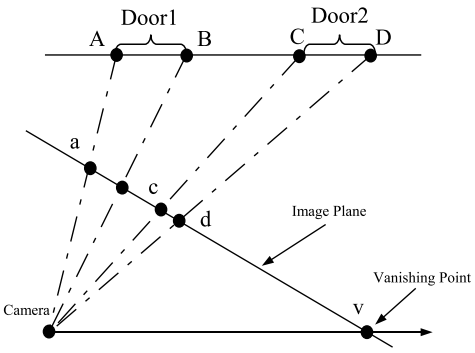

Fig. 8. The relative width of two doors can be computed using only image measurements, given the vanishing point.

VIII. Experimental results

To test the performance of the system, we created a database consisting of 577 images of doors in more than twenty different buildings exhibiting a wide variety of different visual characteristics. The images were captured by an inexpensive Logitech QuickCam Pro 4000 mounted 32 cm above the floor on an ActivMedia Pioneer P3AT mobile robot. Of these images, 100 were used for training the algorithm, and the remaining 477 images were used for testing. The algorithm was implemented in Visual C++ and runs at a speed of 4 frames per second on 320×240 images using a 1.6 GHz Dell Inspiron 700m laptop computer.

Our approach combines as many cues as possible, based on the assumption that additional cues will contribute to improving the detection results. To investigate the contribution of each individual cue, we developed separate door detectors that use just that cue. From the ROC curves of these single-cue detectors applied to the testing images, shown in Figure 9, we see that there is a vast difference in the importance of the different cues. The most important single cue is our novel measure of concavity, which alone can achieve an equal error rate (EER) of 80%. Following this cue are the door width, the bottom gap, and color, each of which achieves an EER greater than 70%. The least important cues are the vanishing point and the kick plate.

The Adaboost algorithm [7] was then applied to the training images to automatically determine the weights of the various cues to be used in the final algorithm. Figure 10 shows the ROC curve of two versions of the algorithm, along with our previous line-based approach [3]. The two versions of the algorithm are the slower MCMC algorithm, which detects 90% of the doors with a false positive rate of one non-door for every 14.3 images on average, and the faster DDMCMC algorithm, which detects 86% of the doors with a false positive rate of one non-door for every 11.1 images on average. Both versions achieve essentially the

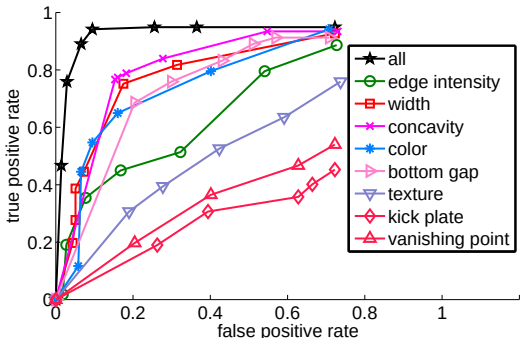

Fig. 9. ROC curves showing the performance of our algorithm compared with the performance of single-cue detectors.

same EER at 90%. While our previous approach worked well on the smaller database used in that paper, its limitations are more apparent on this larger database which exhibits greater variety in appearance and pose.

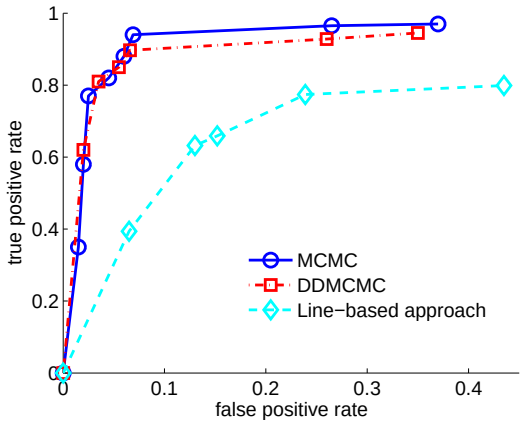

Fig. 10. ROC curves showing the performance of the MCMC and DDMCMC algorithms, as well our previous line-based approach [3] on the database of 477 images.

Some example doors unsuccessfully or incompletely detected by our previous line-based algorithm [3] but successfully detected by the current system are shown in Figure 11. Because it is able to explore the entire solution space, the stochastic search of the MCMC / DDMCMC algorithm achieves a nearly global optimal solution. In contrast, the doors located by the previous algorithm are highly dependent on the positions of the detected vertical lines because it does not include a mechanism for overcoming line detection errors by other cues. In addition, the line-based algorithm does not model the door as a polygon in the image, leading to disconnected line segments that often do not extend the height of the door.

A comparison of the speed of MCMC versus DDMCMC is demonstrated in Figure 12, where it is seen that DDMCMC achieves a speedup of nearly 30 times over MCMC. In addition, the figure shows that an additional speedup results from using Adaboost to weight the different cues according to their importance rather than weighting them equally. For DDMCMC, the Adaboost weighting does not play an important role in the early iterations, but it does increase the speed significantly in later iterations. Both MCMC and DDMCMC compute essentially the same solution, but stopping the iterations of the latter causes less harm in the results. The progress of the DDMCMC algorithm on an example image is shown in Figure 13, where it can be seen that the results are not significantly different when the algorithm is stopped after achieving a value 90% of the maximum, and the yet the speed is increased by more than a factor of 14.

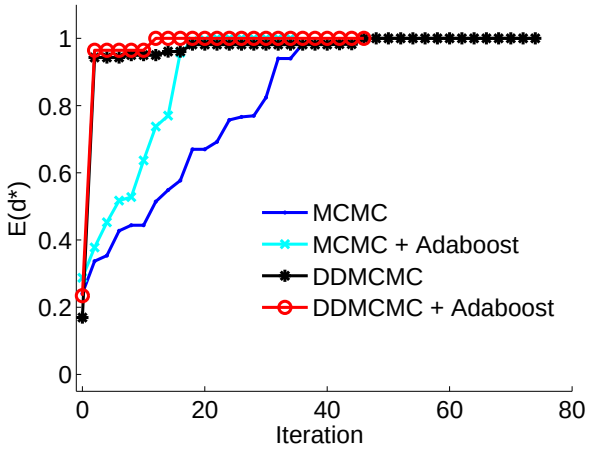

Fig. 12. Comparison of the speed of MCMC and DDMCMC with and without Adaboost weighting, from averaging runs in ten typical images. DDMCMC with Adaboost converges in an average of seven iterations.

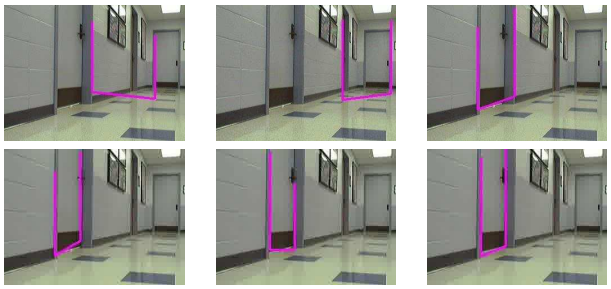

Fig. 13. In lexicographic order, six sequential candidates found by the DDMCMC door detection algorithm on an example image. By the third candidate (top-right), the door is essentially detected, and the algorithm has achieve 90% of the maximum score. By the sixth candidate (bottom-right), the algorithm has settled onto a better solution but at the expense of much more computation time since the time is not linear in the number of candidates. Later candidates require more jumps, so that achieving the result of the sixth candidate requires more than 14 times as much computation as that of the third candidate.

Results of the DDMCMC algorithm with and without calibration are shown in Figure 14. While the calibrated system does achieve higher accuracy, the difference between the two systems is minor, particularly for low false positive rates. These data suggest that, in situations in which calibration is not feasible (e.g., detecting doors in images captured for non-robotics applications), the decrease in accuracy should be small.

Typical doors detected by our system are shown in Fig-

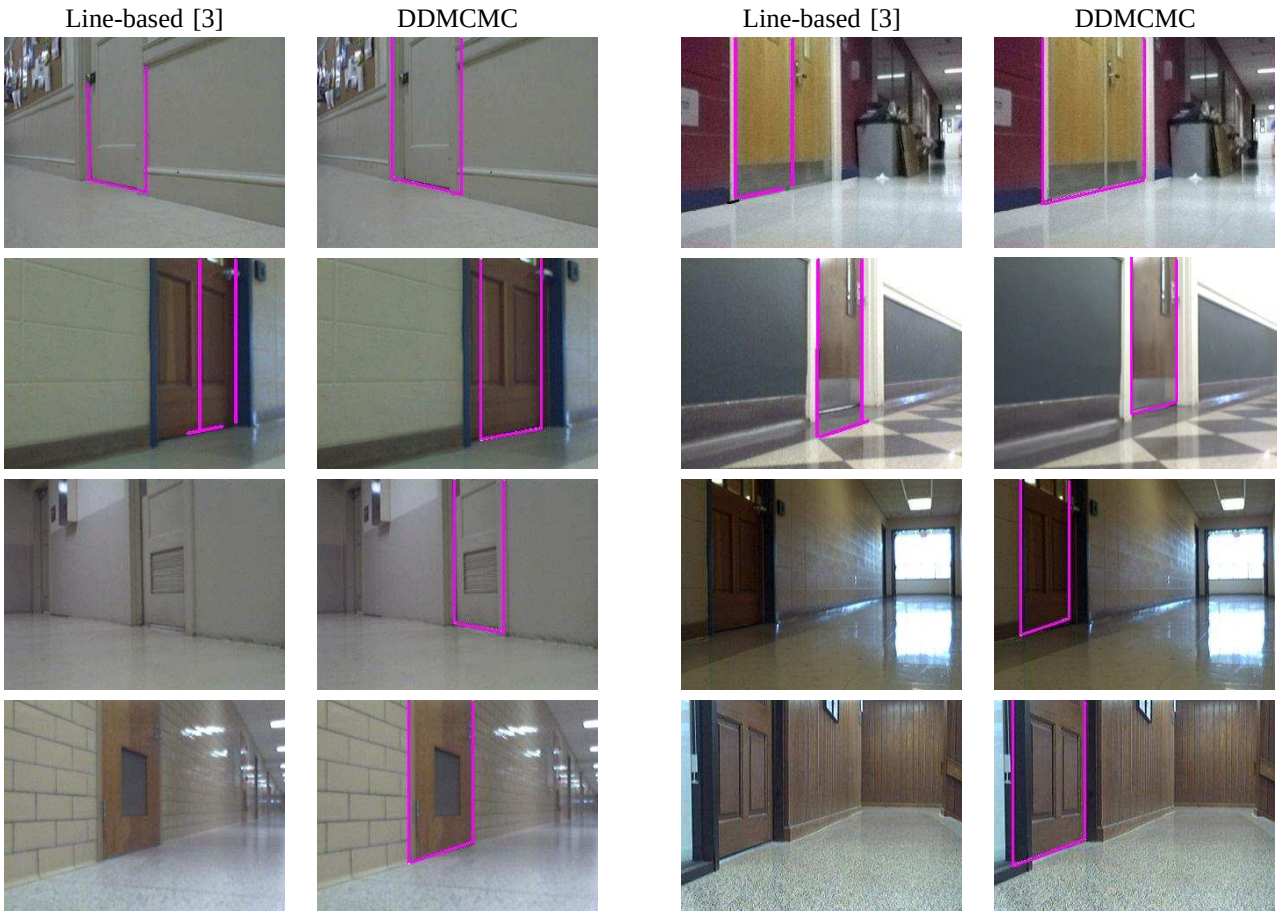

Fig. 11. Example doors successfully detected by stochastic search but unsuccessfully or incompletely detected by our previous line-based algorithm [3].

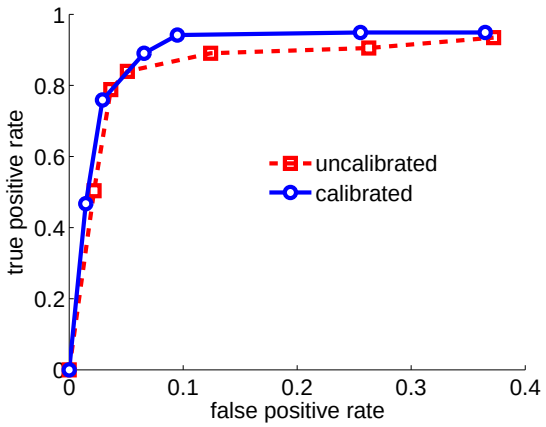

Fig. 14. ROC curves of DDMCMC with and without calibration.

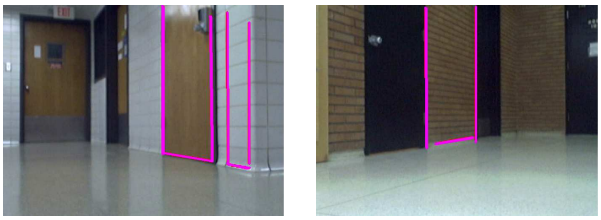

Fig. 16. False negatives and positives. LEFT: One door is successfully detected, but another is missed due to lack of contrast along its bottom edge coupled with strong reflections on the floor; in addition a false positive occurs because of distracting horizontal edges. RIGHT: A dark door that is flush with the wall fails the concavity and bottom gap tests and hence is missed, while edges on the wall are erroneously detected.

ure 15, where it can be seen that our algorithm is capable of detecting doors in difficult conditions including a variety of illumination conditions and viewpoints, in a cluttered environments, and even when the door has the same color as the wall. For completeness, some errors of the algorithm are shown in Figure 16.

To demonstrate the utility of the algorithm, we applied it to live video from 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. Doors that were not repeatedly detected a certain number of image frames were regarded as false positives and discarded. Figure 17 shows the results of five trials in which the robot was manually driven along approximately the same path at a speed of 0.2 m/s down a 40 m $\times$ 15 m corridor with a 90-degree turn. 25 closed doors in the corridor were detected and tracked with 100% accuracy, that is, 5 detections out of 5 trials. A door partly occluded by water fountain was also

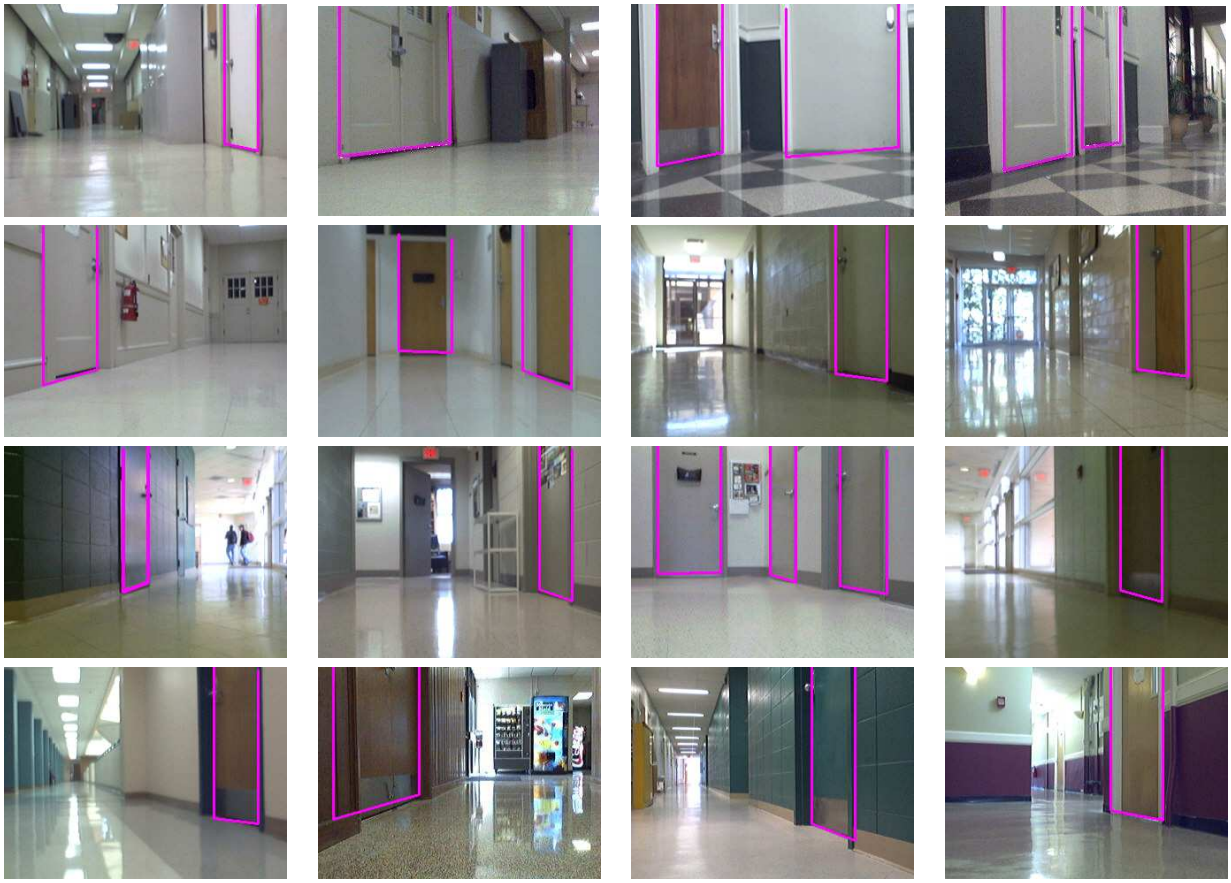

Fig. 15. Doors successfully detected by our algorithm. Note the variety of door widths, door and wall colors, relative pose of the door with respect to the robot, floor texture, and lighting conditions. Distant doors are not considered by the algorithm.

detected and tracked. However, the doors of a cabinet were detected mistakenly because they look similar to real doors. Overall, the detection rate was 100% with a false positive rate of 0.008 per meter driven.

IX. Conclusion

We have presented a vision-based door detection algorithm based on Data-Driven Markov Chain Monte Carlo (DDMCMC). Models of doors utilizing a variety of features, including color, texture, and intensity edges are presented. We introduce two novel geometric features that increase performance significantly: concavity and bottom-edge intensity profile. The Bayesian formulations are constructed and a Markov chain is designed to sample proposals. The features are combined using Adaboost to determine the linear weighting. Doors are detected based on the idea of maximizing a posterior probability (MAP). Using the Monte Carlo technique, our algorithm is able to explore the complex solution space and achieve a nearly global optimal solution. Data-driven techniques, such as edge detection, are used to compute importance proposal probabilities, which drive the Markov Chain dynamics and achieve speedup in comparison to the traditional jump diffusion methods. On a large database of images collected in a wide variety of conditions, the algorithm achieves approximately 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 a

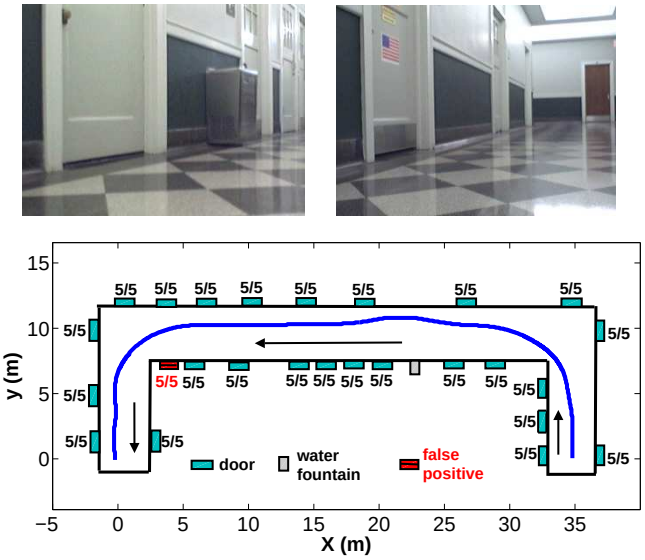

Fig. 17. Top: Two images of the hallway environment near our laboratory. Bottom: 2D plan view of the hallway, along with the results of the experiment. Beside each door is indicated the number of detections / total number of trials.

laptop.

There is much room for future work in this area. Additional features can be incorporated into the Adaboost framework to increase performance. One such cue that could improve results would be door knobs, whose detection will be a challenge due to their extremely small size in the image. Coupling camera calibration with 3D line estimation would enable pose and distance measurements to facilitate the building of a geometric map. In addition, open and closed doors can be distinguished using motion information inside the door, specifically the motion parallax of features inside the room that are visible when the door is open. Our ultimate goal is to integrate the algorithm into a complete navigation system that is able to build a map of an environment and then drive down a corridor and turn into a specified room.

X. 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] R. Barber, M. Mata, M. Boada, J.Armingol, and M. Salichs. A perception system based on laser information for mobile robot topologic navigation. In *The 28th Annual Conference of the IEEE Industrial Electronics Society*, 2002.

[3] Z. Chen and S. T. Birchfield. Visual detection of lintel-occluded doors from a single image. In *Workshop on Visual Localization for Mobile Platforms (in association with CVPR)*, June 2008.

[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] D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

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

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

[8] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[9] T. K. H. Schneiderman. A statistical method for 3D object detection applied to faces and cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.

[11] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, Apr. 1970.

[12] J. Hensler, M. Blaich, and O. Bittel. Real-time door detection based on AdaBoost learning algorithm. In *International Conference on Research and Education in Robotics (Eurobot)*, 2009.

[13] E. Jauregi, E. Lazkano, and B. Sierra. *Mobile Robots Navigation*, chapter Approaches to door identification for robot navigation, pages 241–261. InTech, 2010.

[14] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, Feb. 1967.

[15] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, Nov. 2005.

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

[17] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[18] J. Kittler, A. Hojjatoleslami, and T. Windeatt. Weighting factors in multiple expert fusion. In *Proceedings of the Eighth British Machine Vision Conference*, 1997.

[19] Y. Li and S. T. Birchfield. Image-based segmentation of indoor corridor floors for a mobile robot. In *Proceedings of the IEEE Conference on Intelligent Robots and Systems (IROS)*, Oct. 2010.

[20] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. Rusu, B. Marthi, G. Bradski, K. Konolige, B. Gerkey, and E. Berger. Autonomous door opening and plugging in with a personal robot. In *IEEE International Conference on Robotics and Automation*, 2010.

[21] 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.

[22] R. Muñoz-Salinas, E. Aguirre, and M. García-Silvente. Detection of doors using a genetic visual fuzzy system for mobile robots. *Autonomous Robots*, 21(2):123–141, September 2006.

[23] A. C. Murillo, J. Košecká, J. J. Guerrero, and C. Sagüés. Visual door detection integrating appearance and shape cues. *Robotics and Autonomous Systems*, 56(6):512–521, June 2008.

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

[25] 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.

[26] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[27] 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.

[28] D. M. J. Tax, M. van Breukelen, R. P. W. Duin, and J. Kittler. Combining multiple classifiers by averaging or multiplying? *Pattern Recognition*, 33:1475–1485, 2000.

[29] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[30] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.