The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Image-Based Segmentation of Indoor Corridor Floors for a Mobile Robot

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

Abstract— **We present a novel method for image-based floor detection from a single image. In contrast with previous approaches that rely upon homographies, our approach does not require multiple images (either stereo or optical flow). It also does not require the camera to be calibrated, even for lens distortion. The technique combines three visual cues for evaluating the likelihood of horizontal intensity edge line segments belonging to the wall-floor boundary. The combination of these cues yields a robust system that works even in the presence of severe specular reflections, which are common in indoor environments. The nearly real-time algorithm is tested on a large database of images collected in a wide variety of conditions, on which it achieves nearly 90% detection accuracy.**

I. INTRODUCTION

Image-based floor detection is an important problem for mobile robot navigation. By knowing where the floor is, the robot can avoid obstacles by navigating within the free space. Detecting the floor, the robot is also able to acquire information that would be useful in constructing a map of the environment, insofar as the floor detection specifically delineates between the floor and the walls. Moreover, localization using an existing map can be guided by floor detection by matching the location of the detected floor with the location of the floor expected from the map. Additional reasons for floor detection include problems such as computing the size of the room.

A significant amount of research has focused upon the obstacle avoidance problem. In these techniques, the primary purpose is to detect the free space immediately around the mobile robot rather than the specific wall-floor boundary. Most of these approaches utilize the ground plane constraint assumption to measure whether the disparity or motion of pixels matches the values that would be expected if the points lie on the ground plane. Sabe et al. [13] use stereo cameras to accomplish this task, while the methods of Stoffler [14] and Santos-Victor [17] rely upon optical flow. An alternate approach was pursued by Lorigo et al. [4], who used a combination of color and gradient histograms to distinguish free space from obstacles.

Only a handful of researchers have considered the floor detection problem in its own right. Similar to the obstacle avoidance approaches, the techniques employed tend to utilize the ground plane constraint. Kim and colleagues [2], [3] and Zhou and Li [20] apply planar homographies to optical flow vectors, while Fazl-Ersi and Tsotsos [19] rely on stereo homographies. The approach of [20] computes only a sparse representation of the floor by classifying sparse feature points, while the other two approaches make a pixelwise decision to result in a dense floor representation.

In the computer vision community, some impressive results have been achieved recently for related problems. Lee et al. [1], for example, have developed a method that is capable of performing geometric reasoning on a single indoor image. Inspired by the early line-drawing interpretation work of Guzman [18], they are able to separate the walls from the floor and ceiling using intensity edges and geometric constraints. In another impressive piece of work, Hoiem et al. [6] also assign labels to pixels based upon image data and class priors, primarily for outdoor scenes. Although the results of both of these approaches are promising, neither operates in real time, thus limiting their application to robotics at the moment. Moreover, the work of [1] requires the ceiling to be visible, which is often not the case when the camera is mounted on a mobile robot that is low to the ground.

In this paper we introduce a novel method for floor detection from a single image for mobile robot applications. Unlike existing techniques, the approach does not make use of the ground plane constraint and therefore does not use homographies, optical flow, or stereo information. As a result, it does not require the camera to be calibrated, not even for lens distortion. Inspired by the work of McHenry et al. [12], our technique combines multiple cues to enable the wall-floor separating boundary to be estimated in the image. One especially noteworthy aspect of our approach is its proper handling of specular reflections. It is not uncommon for indoor scenes to contain significant amounts of reflection of light off the floor, particularly when the overhead lights are bright, the sun is shining through a window, and/or the floor is particularly shiny. These reflections can confuse homography-based approaches, because they cause pixels on the ground plane to violate the ground plane constraint. Reflections are also known for being difficult to model, causing spurious intensity edges and altering the color appearance of the floor. By combining multiple visual cues, our approach is often able to ignore the spurious edges, distinguishing between edges arising from the structure of the scene and those produced by the specular reflections. Another contribution of this work is the introduction of a rather large database

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

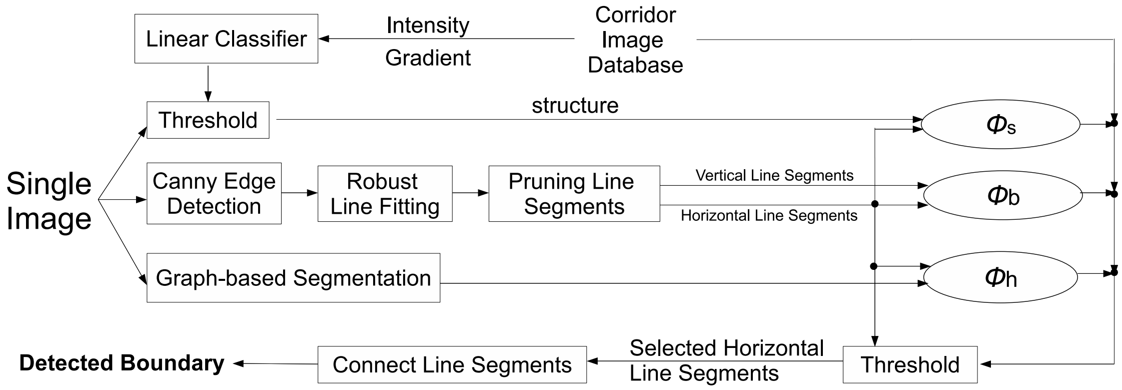

Fig. 1. Flowchart of the proposed method for floor detection.

of more than 400 indoor corridor images from dozens of different buildings, exhibiting a wide range of conditions. On this challenging dataset, our algorithm is able to successfully detect the floor on around 90% of the images.

Figure 1 shows an overview of our approach. Horizontal and vertical line segments are detected, and three cues are used to evaluate the likelihood of each horizontal line segment being on the wall-floor boundary. The weighted sum of these values is thresholded, and the remaining segments are connected to form the estimated wall-floor boundary. Note that our approach operates on a single image, without stereo or motion information.

II. DETECTING LINE SEGMENTS

A. Detecting and classifying line segments

The first step of the approach is to detect intensity edges by applying the Canny edge detector [8] to the grayscale image. Then a robust line fitting method is applied to the intensity edges to obtain a set of line segments. We use the Douglas-Peucker algorithm [9], with the modification described in [11] to improve the retention of small line segments that occur at the bottom edge of doors. Each line segment is defined by two endpoints in the image. Line segments are divided into two categories: vertical and horizontal. Based on over three hundred corridor images and the ground truth, we determined a tight slope range to the vertical line segments, so that a line segment is classified as vertical if its slope is within ±5° of the vertical direction. Horizontal line segments are given a wider slope range: A line segment is classified as horizontal if its slope is within ±45° of the horizontal direction. All other slopes are discarded.

B. Pruning line segments

Due to the noisy conditions of real-world scenes, the procedure just described often produces spurious line segments that are not related to the wall-floor boundary. We apply two additional steps to prune such segments. First, we discard segments whose length is less than a threshold (60 pixels for vertical lines, and 15 pixels for horizontal lines). Then we compute the intersections of the horizontal line segment pairs, after which we compute the mean of the y coordinate of the intersections inside the image to yield an estimate of the vanishing line. For any pair of horizontal line segments, the intersection point is calculated by the cross product between the two line extensions, using homogeneous coordinates:

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

where each horizontal line is described by $ax + by + c = 0$, and the intersection point $\begin{bmatrix} v_x & v_y \end{bmatrix}^T$ is determined by dividing by the scaling factor w. Once the intersection point has been detected, all horizontal line segments that lie above the vanishing line are discarded. The result of detection, classification, and pruning of line segments is illustrated in Figure 2.

III. SCORE MODEL FOR EVALUATING LINE SEGMENTS

Not all of the horizontal line segments that remain from the pruning step will be related to the wall-floor boundary. To determine the likelihood that a horizontal segment ℓ_h is near this boundary, we compute a weighted sum of scores for three individual visual cues:

$$\Phi_{total}(\ell_h) = w_s \overline{\phi}_s(\ell_h) + w_b \overline{\phi}_b(\ell_h) + w_h \overline{\phi}_h(\ell_h), \qquad (2)$$

where w_s, w_b, and w_h are the weights, and $\overline{\phi}_s(\ell_h)$, $\overline{\phi}_b(\ell_h)$, and $\overline{\phi}_h(\ell_h)$ are the three individual scores, which are now described.

A. Structure Score

We have found that a surprisingly effective cue for distinguishing the walls from the floor in typical corridor environments is to simply threshold the image. This approach

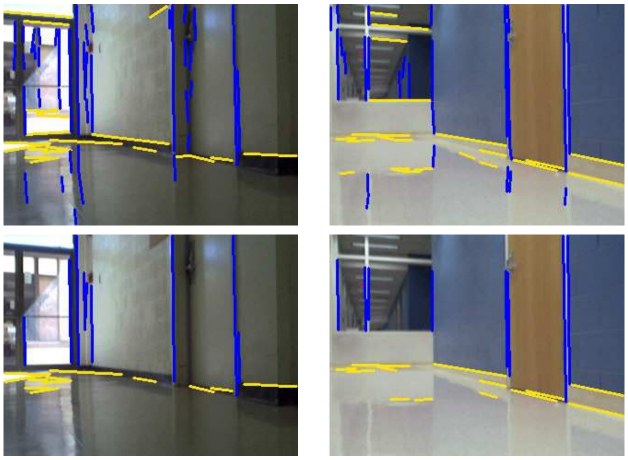

Fig. 2. The wall-floor boundary of typical corridor images is difficult to determine due to strong reflections and shadows. TOP: Two images, with the result of the modified Douglas-Peucker line fitting algorithm applied to Canny edge detection overlaid. The line segments are classified into two groups: vertical (blue) and horizontal (yellow). BOTTOM: Line segments have been pruned according to length and the vanishing point, as described in the text, to reduce the influence of reflections and shadows.

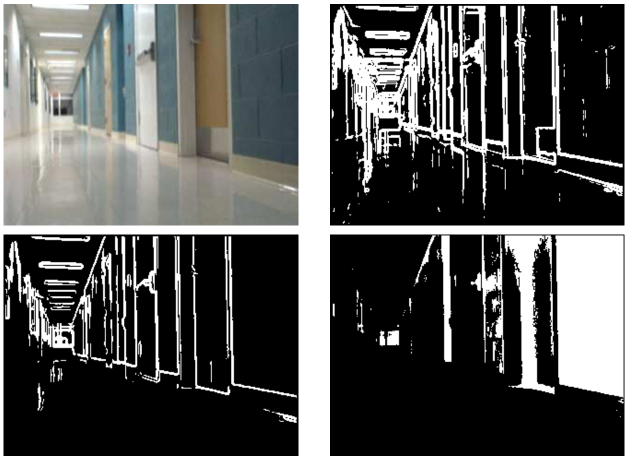

Fig. 3. TOP-LEFT: A typical corridor image. TOP-RIGHT: Pixels with gradient magnitude greater than a threshold are shown in white. BOTTOM-LEFT: Using the separating curve from Fig. 5, the edge pixels mostly follow the boundaries of the wall, door frames, and floor. BOTTOM-RIGHT: The original image thresholded by a value determined by the separating curve, thus revealing the structure of the corridor.

works especially well in environments in which the walls are darker than the floor, and it also takes advantage of the fact that the baseboard is often darker than the floor due either to its painted color, shadows, or collected dirt. In fact, even when the walls, doors, and floor are nearly white, the technique is more effective than one might at first expect. This is partly due, perhaps, to the shadows under the doors that appear no matter the color of the surfaces in the corridor. In some environments, the walls are lighter than the floor, in which case thresholding will still distinguish between the two but with reverse binary labels compared with those situations just described.

An important step is to determine the value of the threshold to use. Our proposed approach to thresholding, which will be described in a moment, involves examining the structure of the scene, i.e., the intensity edges. In Figure 3, for example, the top-right image containing the pixels with large gradient magnitude reveals the edges of the doors, lights, wall, and so forth. A human observer looking at this type of image could infer the structure of the scene with little difficulty, as was observed by Lee et al. [1]. One idea that we tried is to compute the desired threshold as the average graylevel intensity of these intensity edges. While this approach works fairly well, the distracting intensity edges caused by the reflections on the floor skew the computed threshold in such a way as to reduce the quality of the thresholded image. Therefore, we first discard these intensity edges in a manner described in a moment, in order to result in a relatively clean thresholded image. Figure 3 illustrates the process.

For comparison, Figure 4 shows the output of two standard algorithms based on the gray-level histogram, Ridler-Calvard [15] and Otsu [16], on the same image. Compared with our approach, the standard techniques mistakenly label reflective pixels on the floor, due to the failure of the simplified model of a bimodal gray-level histogram to accurately capture the subtle complexities of indoor scenes. Table I provides a quantitative comparison using the images from our corridor image database. The table shows the percentage of images for which the thresholded result does not contain spurious pixels on the floor.

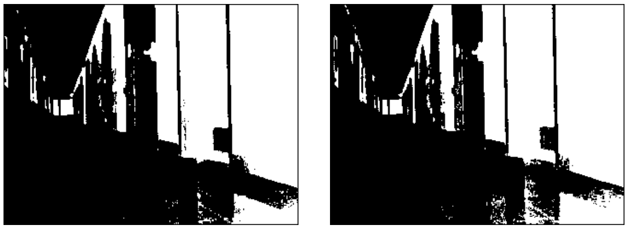

Fig. 4. Results of two standard thresholding algorithms on the same image as the previous figure: Ridler-Calvard [15] (left), and Otsu [16] (right). Notice the spurious pixels on the floor due to reflection and shadows.

	Ridler-Calvard [15]	Otsu [16]	Ours
correctness	62%	66%	82%

TABLE I

QUANTITATIVE COMPARISON OF OUR THRESHOLDING METHOD WITH TWO STANDARD ALGORITHMS. SHOWN ARE THE PERCENTAGE OF IMAGES WITHOUT SPURIOUS PIXELS ON THE FLOOR.

We now describe our approach to determining the threshold value. The intensity edges that arise due to reflections on the floor tend to have very high intensity values but quite low gradient magnitude values, the latter being because of the inherent blur that occurs because floors are not perfectly reflective surfaces. To test this hypothesis, we used our database of over 400 images. We manually selected over 800 points on these images that lie on true edges on the walls in the world (i.e., they lie on door frames, etc.), and we also randomly selected the same number of points that are not on true edges (i.e., they are on the ceiling or the floor). An SVM-based classifier [7] was used to find the best separating hyperplane to distinguish between the two classes of data using the intensity value and the gradient magnitude of each point. Figure 5 shows the training values along with the separating curve. (The hyperplane in the higher dimensional space defined by the polynomial kernel function becomes a curve when projected back into the original feature space.)

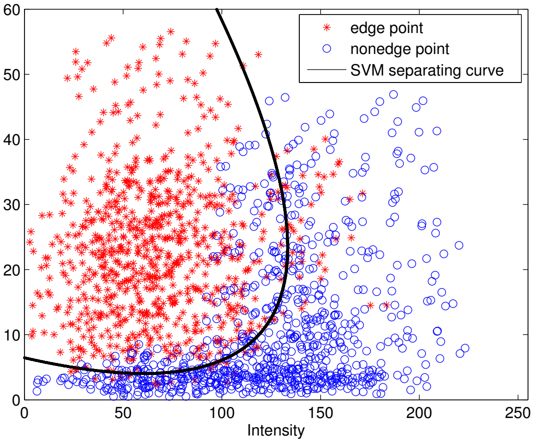

Fig. 5. Linear-based classification of pixels on horizontal line segments. The x coordinate is the intensity of the pixel, while the y coordinate is its gradient magnitude. From the training data, some pixels are edge points (red stars), while others are nonedge points (blue circles). A polynomial kernel based classifier separates the two groups of points by an optimal curve (black).

From the figure, it is clear that taking both the gradient magnitude and intensity value into account yields better separation than using either alone. We define the set ξ' to denote the pixels in the image whose intensity value and gradient magnitude cause them to lie above the SVM-based separating curve.

The average intensity of the pixels in this set determines the threshold that is used for extracting the main structure in a corridor image:

$$\tau_{LC} = \frac{1}{|\xi'|} \sum_{(x,y)\in\xi'} I(x,y). \tag{3}$$

The structure image is therefore defined as the binary image resulting from thresholding the image using this value:

$$\mathcal{S}(x,y) = \left(I(x,y) > \tau_{LC}\right). \tag{4}$$

Since the true edges are generally aligned with the structure of the corridor, we compute the structure score ϕ_s of a line segment ℓ_h by measuring the distance of each pixel in the line segment to the nearest non-zero pixel in the structure image:

$$\phi_s(\ell_h) = \sum_{(x,y)\in\ell_h} d[(x,y),\mathcal{S}], \tag{5}$$

where $d[(x,y),\mathcal{S}]$ computes the distance between the point (x,y) and the structure image $\mathcal{S}$. For fast computation, we use the chamfer algorithm to compute the distance. The score is normalized using a Gaussian distribution with a standard deviation σ_s:

$$\bar{\phi}_s(\ell_h) = \exp\left\{-\frac{\phi_s(\ell_h)}{2\sigma_s^2}\right\}, \tag{6}$$

where $\sigma_s = 10$.

B. Bottom Score

The vertical line segments provide an important cue to provide independent evaluation of whether a given horizontal line segment is likely to be on the wall-floor boundary. First, we discard all vertical line segments whose bottom point does not extend below the middle of the image. This step helps to ignore vertical line segments that arise due to texture on the wall or ceiling, since the camera on the robot is low to the ground and facing straight ahead. Then, we sort the remaining vertical segments according to their x coordinate and connect their bottom endpoints to yield a polyline that extends from the left side of the image to the right side. Even though this polyline is a rough approximation of the wall-floor boundary, it is fast to compute and generally reliable enough to help guide the evaluation. The bottom score of a horizontal line segment ℓ_h is computed as the distance of all of its pixels to the polyline ℓ_b:

$$\phi_b(\ell_h) = \sum_{(x,y)\in\ell_h} d[(x,y),\ell_b], \tag{7}$$

where $d[(x,y),\ell_b]$ computes the distance between the point (x,y) and the polyline. To normalize the score, we use the Gaussian distribution with a standard deviation σ_b:

$$\bar{\phi}_b(\ell_h) = \exp\left\{-\frac{\phi_b(\ell_h)}{2\sigma_b^2}\right\}, \tag{8}$$

where $\sigma_b = 30$.

Figure 6 illustrates this computation for two typical corridor images. The horizontal line segments that benefit most from this computation are highlighted with red ellipses. We can see that this computation is especially helpful for reducing the likelihood of considering line segments on the wall to be part of the wall-floor boundary.

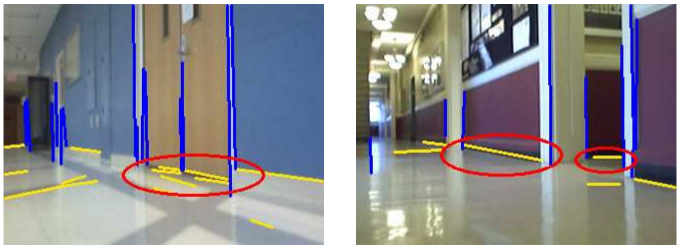

Fig. 6. Horizontal and vertical line segments detected in two images. The red ellipses highlight the horizontal segments with the highest bottom score, due to their proximity to the bottom endpoints of nearby vertical line segments.

C. Homogeneous Score

In many cases, the floor of a typical corridor environment is fairly homogeneous in its color throughout. In contrast, there tend to be moderate to high amounts of texture on the wall regions due to decorations, posters, door knobs, kick plates, nameplates, windows, and so forth. Similarly, the lights in the ceiling cause texture in that region as well. To take advantage of this information, we perform color-based segmentation of the image to favor horizontal line segments which are located just above large homogeneous regions, since the floor is generally the largest homogeneous region in the image.

We employ the graph-based segmentation algorithm of Felzenszwalb and Huttenlocher [5] because it is computationally efficient, requires few parameters (e.g., the minimum size of a region), and produces reasonable results. The results of this algorithm on a couple of typical corridor images are shown in Figure 7. Notice that the floor is the largest homogeneous region in both images, which is often the case in our image database. Occasionally, disturbance from reflection or texture on the floor prevent this cue from being successful, which helps to motivate the need for multiple cues.

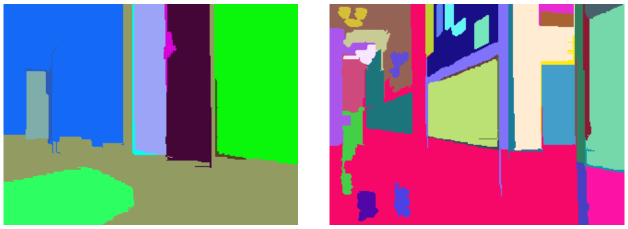

Fig. 7. The result of graph-based segmentation [5] on two corridor images used in Figure 6, with each region assigned a random color. Note that the floor is almost a homogeneous area in both images, while the wall and doors are divided into several smaller regions.

The homogeneous score of a horizontal line segment is computed as

$$\bar{\phi}_h(\ell_h) = \frac{|\mathcal{R}|}{|\mathcal{R}_{\max}|}, \tag{9}$$

where $|\mathcal{R}|$ denotes the number of pixels in the region $\mathcal{R}$ just below the line segment, and $\mathcal{R}_{\max}$ is the maximum region size among all the segments found by the graph-based segmentation.

D. Detecting the wall-floor boundary

Each horizontal segment ℓ_h for which $\Phi_{total}(\ell_h) > \tau_\phi$, where τ_ϕ is a threshold, is retained. These remaining line segments are then ordered from left to right in the image, and their endpoints are connected. At the left and right borders of the image, the lines are extended. This results in a polyline stretching across the image defining the boundary between the wall and the floor.

IV. Experimental Results

To test the performance of our algorithm, an image database of more than 400 corridor images was taken in twenty different buildings exhibiting a wide variety of different visual characteristics. The images were captured by a Logitech QuickCam Pro 4000 webcam mounted about 30 cm above the floor on an ActivMedia Pioneer P3AT mobile robot. The images were processed by an algorithm implemented in the C++ programming language on a 2.4 GHz Core 2 processor (Dell XPS M1330 laptop). Although the computation time varies somewhat according to the number of detected line segments, the algorithm runs at approximately 5 frames/sec.[1]

For all environments, the equation of the SVM-based separating curve is

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 3.5 & 26 & 475 \\ 26 & 50 & -125 \\ 475 & -125 & 10 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0, \tag{10}$$

the weights for the individual scores are $w_s = 1.6$, $w_b = 0.75$ and $w_h = 1.0$, respectively, and the total threshold is $\tau_\phi = 2.7$.

To evaluate the algorithm, the images in the database were manually labeled by clicking on a number of points and then fitting a B-spline curve to yield a ground truth wall-floor boundary. We define the error of the algorithm applied to an image as the number of pixels misclassified as floor or non-floor, normalized by the total number of ground truth floor pixels. Equivalently, the error can be computed as the sum, over all the columns $x = 0, \ldots, width - 1$ in the image, of the difference between the ground truth y coordinate $y_{GT}^{(x)}$ and the estimated y coordinate $\hat{y}^{(x)}$:

$$r_{err} = \frac{\sum_x \left| \hat{y}^{(x)} - y_{GT}^{(x)} \right|}{\sum_x \left| height - y_{GT}^{(x)} \right|}, \tag{11}$$

where the image is of size $width \times height$, and the subtraction in the denominator arises from the convention that the y coordinate is with respect to the top of the image. We set a threshold of 10%, so that the detection for an image is considered a failure for a particular image if $r_{err} > 0.1$ for that image. Using this convention, our approach correctly detects the floor region in 89.1% of the image database.

Figure 8 presents the results of our algorithm on some typical corridor images. The first row displays wall-floor boundaries that extend upward from left to right in the image, while the second row shows the reverse situation. In the third row, both sides of the corridor are visible, so that the boundary extends in both directions. And the fourth row shows floors with extremely strong reflections on the floor, where the floor and wall are again distinguished using only low-level information. From these results, we can see that our approach is capable of detecting floors in corridors under different illumination conditions and perspectives. In addition, Figure 9 shows some successful results on images downloaded from the internet, showing the versatility of the approach.

[1] See http://www.ces.clemson.edu/˜stb/research/floor_detection for videos of the results.

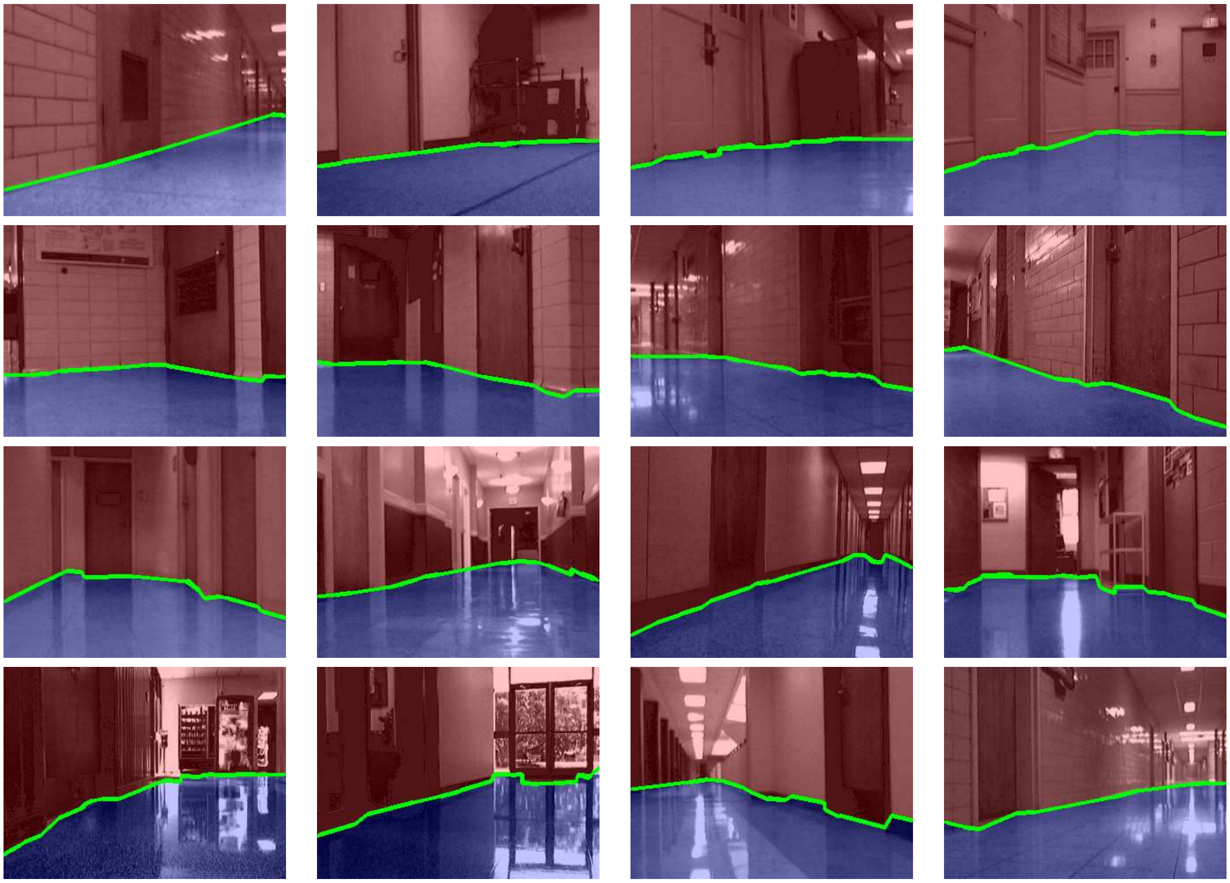

Fig. 8. Examples of floor successfully detected by our algorithm. Note the variety of floor materials, floor reflectivity, relative pose of the floor with respect to the robot, and lighting conditions (Best viewed in color).

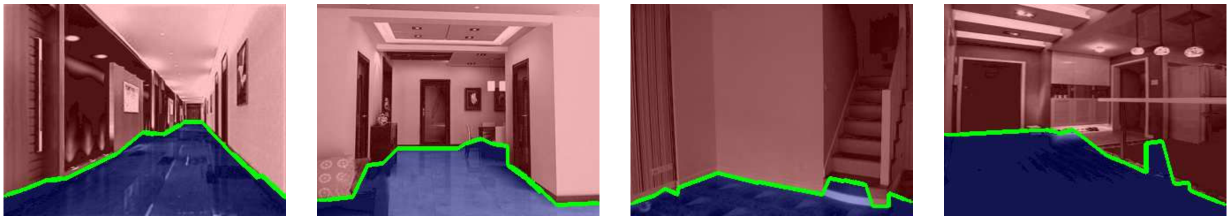

Fig. 9. Results of our algorithm on images downloaded from the web (Best viewed in color).

Some examples where the algorithm fails are shown in Figure 10. In the first image from the left, the checkered floor lead to many horizontal line segments that are mistakenly interpreted by our score model, and the graph-based segmentation detects many small pieces on the floor region rather than a single homogeneous segment. For the second image, the wall texture also results in many horizontal line segments that distract the algorithm. For the third image, the shadows and reflection dominate the line segment detection due to the poor gain control of the camera, making the final result less precise. And for the fourth image, the far glass door with no absolute edges makes it difficult for even a human observer to precisely locate the wall-floor boundary, though the results of the algorithm are even worse due to the lack of horizontal segments in that region of the image.

It is difficult to compare these results with existing techniques. The impressive work of Hoiem et al. [6] is aimed primarily at outdoor environments, while the recent work of Lee et al. [1] is designed to reconstruct indoor scenes when the ceiling is visible. Neither system is real time. Nevertheless, Figure 11 shows some successful results of our algorithm working on three of the failure examples given in the latter paper. Perhaps the closest work to our own in terms of purpose and scope is that of Kim and colleagues [2], [3], which requires two image frames to segment the floor from the rest of the scene. Because their approach does not contain a specific mechanism to handle strong reflections, it is doubtful that it would work successfully on the many images in our database that contain such extreme lighting conditions. Similarly, the obstacle avoidance system of Lorigo et al. [4] is not designed to handle strong reflections.

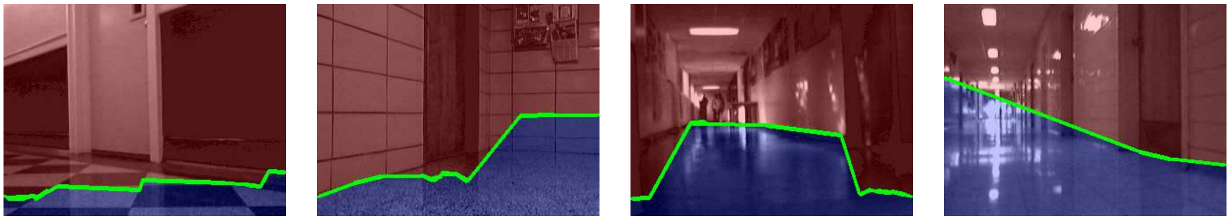

Fig. 10. Examples for which our algorithm fails to properly detect the floor. From left to right, the failures are caused by strong texture on the floor, texture on the wall, an overly dark image from poor image exposure, and excessive bright lights at the end of the corridor (Best viewed in color).

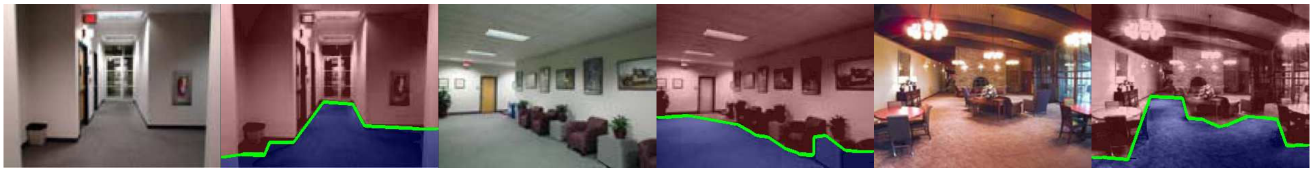

Fig. 11. Results of our algorithm working on three failure examples given in Lee et al. [1] (Best viewed in color).

V. CONCLUSION AND FUTURE WORK

We have presented an image-based floor detection algorithm using an uncalibrated camera. The floor is detected by a camera mounted on a mobile robot, which maintains a low perspective of the scene. The novel approach combines the results of applying three different visual cues to test the validity of horizontal line segments detected in the image. Our approach achieves nearly 90% detection of the wall-floor boundary on a rather large database of over 400 images captured in a variety of environments exhibiting difficult conditions such as extreme reflection. The algorithm is suitable for real-time mobile robot applications using an off-the-shelf camera. One limitation of the current approach is its tendency to get confused when the floor is highly textured, or when the image is especially dark due to poor gain control.

VI. ACKNOWLEDGMENT

The authors would like to thank Zhichao Chen for help in capturing the corridor image database.

REFERENCES

[1] D. C. Lee, M. Hebert, and T. Kanade. “Geometric Reasoning for Single Image Structure Recovery”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[2] Y.-G. Kim and H. Kim. “Layered Ground Floor Detection for Vision-based Mobile Robot Navigation”. In *Proceedings of the IEEE lnternational Conference on Robotics and Automation*, 2004.

[3] X.-N. Cui, Y.-G. Kim, and H. Kim. “Floor Segmentation by Computing Plane Normals from Image Motion Fields for Visual Navigation”. In *International Journal of Control, Automation, and Systems*, 7(5):788-798, 2009

[4] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimson. “Visually-Guided Obstacle Avoidance in Unstructured Environments”. *Proceeding of the IEEE International Conference on Robotics and Automation(ICRA)*, 1997.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. *International Journal of Computer Vision*, 59(2):167-181, September 2004.

[6] D. Hoiem, A. A. Efros, and M. Hebert, “Geometric Context from a Single Image”. *International Conference on Computer Vision (ICCV)*, 2005.

[7] C. Chang and C. Lin. “LIBSVM – A Library for Support Vector Machines. *http://www.csie.ntu.edu.tw/%7Ecjlin/libsvm/*

[8] J. Canny. “A computational approach for edge detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679-698, 1986.

[9] 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.

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

[11] Z. Chen and S. T. Birchfield. “Visual Detection of Lintel-Occluded Doors from a Single Image”. In *Workshop of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[12] K. McHenry, J. Ponce, and D. Forsyth. “Finding Glass”. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[13] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara. “Obstacle Avoidance and Path Planning for Humanoid Robots using Stereo Vision”. In *Proceeding of the IEEE International Conference on Robotics and Automation*, 2004.

[14] N. O. Stoffler, T. Burkert, and G. Farber. “Real-Time Obstacle Avoidance Using an MPEG-Processor-based Optic Flow Sensor ”. *International Conference on Pattern Recognition*, Barcelona. September 2000.

[15] T. Ridler and S. Calvard. “Picture Thresholding Using an Iterative Selection Method”. *IEEE Transactions on Systems, Man and Cybernetics*, 8(8):630-632, 1978.

[16] N. Otsu. “A threshold selection method from gray-level histograms”. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62-66, 1979.

[17] J. Santos-Victor and G. Sandini. “Uncalibrated obstacle detection using normal flow”. *Machine Vision and Applications*, 14(2):158-177, 1995.

[18] A. Guzman. “Decomposition of a visual scene into three dimensional bodies”. In *Proceedings of Fall Joint Computer Conference*, 1968.

[19] E. Fazl-Ersi and John K. Tsotsos. “Region Classification for Robust Floor Detection in Indoor Environments ”. In *Proceedings of the 6th International Conference on Image Analysis and Recognition*, pages 717-726, 2009.

[20] J. Zhou and B. Li. “Robust ground plane detection with normalized homography in monocular sequences from a robot platform ”. In *Proceedings of the International Conference on Image Processing*, 2006.