

Real-time Door Detection based on AdaBoost learning algorithm

Jens Hensler, Michael Blaich, and Oliver Bittel

University of Applied Sciences Konstanz, Germany

Laboratory for Mobile Robots

Brauneggerstr. 55 D-78462 Konstanz

{jhensler,mblaich,bittel}@htwg-konstanz.de

<http://www.robotik.in.htwg-konstanz.de/>

Abstract. Doors are important landmarks for robot self-localization and navigation in indoor environments. Existing algorithms of door detection are often limited for restricted environments. They do not consider the diversity and variety of doors. In this paper we present a camera- and laser-based approach, which allows finding more than 72% doors with a false-positive rate of 0.008 in static testdata. By using different door perspectives from a moving robot, we detect more than 90% of the doors with a very low false detection rate.

1 Introduction

For an indoor environment, doors are significant navigation points. They represent us the entrance and exit points of rooms. Doors provide stable and semantically meaningful structures for the robot localization. Therefore, robust real-time door detection can be used for many important robot applications. Among these applications are courier, observation or tour guide robots.



Fig. 1. Shows the mobile robot Pioneer2DX. It is equipped with a camera, a 2D laser range finder and an odometer.

In the past, the problem of door detection has been studied several times. The approaches differ in the implemented sensor system and the diversity of environments and doors, respectively. For example, in [6] and [3] only visual



Fig. 2. Illustrates typical door images, taken by the Pioneer2DX camera. The top of the doors are occluded and also the diversity of doors are recognizable: The doors have different poses, colors, lighting situations, as well as different features e.g. door gap or texture on the bottom.

information was used. Others, like [1] apply a additional 2D laser range finder and receive thereby better results.

From this approaches, we found, that there are two major difficulties in autonomous door detection. Firstly, it is often impossible to get the whole door in a single the camera image. In our scenario, the robot camera is close to the ground, such that, the top of the door is often not captured by the robots camera (see figure 1 and 2).

The second difficulty is characterized by the multiplicity and diversity of doors (even for the same door types) in various environments. Like figure 2 depicts, doors can have different poses, lighting situations, reflections, as well as completely different features. The main features, which characterize a door are illustrated in figure 3. A door can be recognized e.g. by its color and texture with respect to the color and texture of the surrounding wall, respectively. Also door gaps or door knobs are indicators of a door. Even though these features are not always present or cannot always be detected in a single camera image, a robust algorithm should detect the door by using the remaining features.

In a recent work [3], this two issues of door detection were solved by extracting several door features out of the robots camera images and by applying the AdaBoost algorithm [5]. The algorithm combines all the weak features of a door candidate to receive a strong door classifier, which allows to decide if a door is found.

For our situation this approach was not sensitive enough. We did not reached the same high detection rate in our complex university environment with a similar system. Therefore, we added a laser based distance sensor. Further weak classifiers were used to improve the detection results. In the experimental result section we demonstrate the performance of our system on a large database of images from different environments and situations.

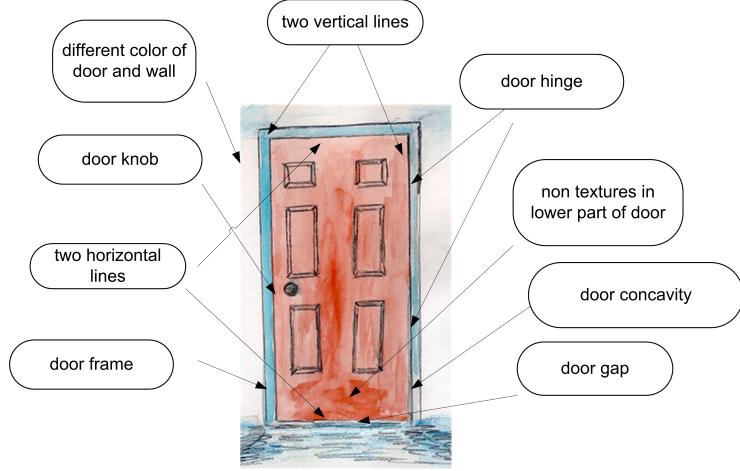


Fig. 3. Door is characterized by several features.

2 The AdaBoost Algorithm

The concept behind all boosting algorithms is to use multiple weak classifiers instead of one strong classifier and solve the decision problem by combining the results of the weak classifiers. It is often easier to find several rules of thumb for a problem instead of one complex rule to solve it. The AdaBoost algorithm is using a training dataset to build up the strong classifier. For this purpose, the algorithm requires, that each weak classifier reaches in the training process at least 50% success and the errors of the classifiers are independent. If this is given, the algorithm is able to improve the error rate by calculating optimal weight for each weak classifier. If $y = h_n(x)$ is the output of a n^{th} weak classifier to the input x and α_n the weight for the n^{th} classifier created in the training process. The strong classifier is given by:

$$H(x) = \text{sign} \left(\sum_{n=1}^N \alpha_n h_n(x) \right) \quad (1)$$

3 Detection of Door Features

As indicated in figure 4, we use the robots camera image and the laser based distance signal for the door detection. Out of the camera image we extract vertical lines to find door candidates in the images. Hereby, we assume that each door has a vertical line on the right and left side. As a consequence, a door is not detected, if the door posts are not visible. In the next step we check each candidate for the seven door features, which represent the weak classifiers: a door can have a certain width, the color of the door can be different from the color of the wall, a door may have a door frame or door knob, the door can stick

out of the wall or not. The result is not effected by a single classifier as long the weighted majority of the other classifier overcomes this loss.

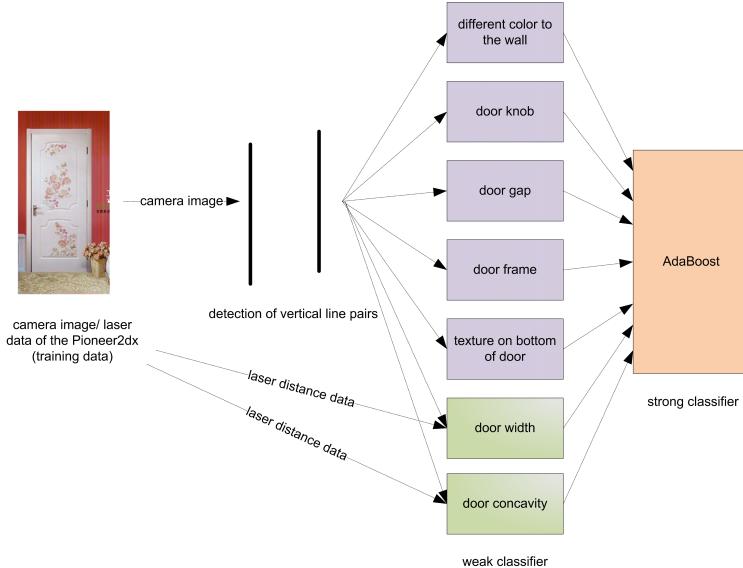


Fig. 4. This Image illustrates our approach to find doors. Several weak classifiers are combined using the AdaBoost algorithm in the training process. The door width and door concavity classifiers are using the laser distance data from the robot in addition to the camera image.

3.1 Vertical Line Pairs

As mentioned before we use the vertical line pairs generated by the door frame as door candidates for the AdaBoost algorithm. We use the Contour Following Algorithm [7] to find vertical lines. Compared to other transformations, this method has the advantage, that we obtain the start and end points of the lines.

Figure 5 shows the typical result of our line detection algorithm. Broken contours caused by door hinges or knobs were overcome by merging vertical lines separated by a small gap. Reflection of the door on the ground causes unexpected long lines, which lead to failures of some weak classifier. Those long lines were eliminated by cutting lines on the ground and optimizing the Canny Edge Detection algorithm.

Not every combination of vertical line pairs in an image correspond to door candidates. The number of candidates was drastically reduced by the following rules:

- The vertical lines need to have a minimal length.



Fig. 5. Vertical line pairs after the Contour Following Algorithm. Two vertical line pairs according to same door candidate. Some combinations can be excluded. Line pairs corresponding to a door candidate are labeled with the same green number in the image.

- The horizontal distance between the two vertical lines has to be between a maximal and minimal value.
- The end points of the vertical lines have a minimal vertical shift.
- If there are lines close together, which all may represent a door candidate according to the earlier rules, only the inner lines are used. The other lines are indicators for a door frame.

3.2 Door Gap

To avoid friction between the door and the ground, doors are constructed with a small gap. As a result, this gap appears brighter or darker than the surrounding area, depending on the illumination of the room behind the door. With a look at vertical profile we determine the maximum or a minimum in the vertical profile (figure 6). For each pixel along the bottom door edge, we compute the minimum and maximum intensities in the vertical profiles. If the mean value of all minimum and maximum values is above or below a certain threshold the classifier found a door.

3.3 Color of the Wall

In many environments doors have a different color than the surrounding walls. Figure 7 illustrates that the 2D hue- saturation histogram of the door is totally

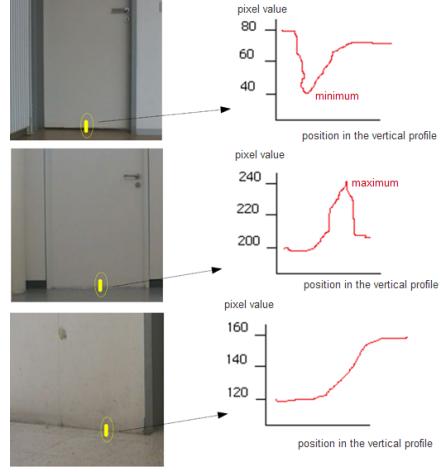


Fig. 6. This figure shows intensities in the vertical profile of a door. In the first image the light in the room behind the door is off. As a result, we receive a minimum in the vertical profile. In contrast, the light in the room behind the door in the center image is on. In consequence, the vertical profile has a maximum in the vertical profile. If there is no door gap, which is shown in the last picture, the vertical profile does not have any minimum or maximum.

different to the histogram of the wall. To measure this feature we use the *histogram intersection* [8] method. Given is a pair of histograms (door D and wall W), each contains n bins. The intersection, bins with the same color, can be calculated:

$$\sum_{j=1}^n \min(D_j, W_j) \quad (2)$$

For a value between 0 and 1 we normalized this formula:

$$H(D, W) = \frac{\sum_{j=1}^n \min(D_j, W_j)}{\sum_{j=1}^n D_j} \quad (3)$$

with

$$H(D, W) < T_{wall} \quad (4)$$

and T_{wall} is the threshold which decides if a door is found or not.

One difficulty here is to choose the right area for the wall and the door. Therefore, we use the distance between the two vertical lines. Additionally, we have to consider that the door frame is not included in this areas.



Fig. 7. Left: Areas which are considered for the ColorWallClassifier. Center: 2D hue-saturation histogram of the wall area. Right: 2D hue-saturation histogram of the door area.

3.4 Texture

The bottom area of a door is usually not textured. In contrast, the wall and other objects are textured in this area. Textures are going to be visible in the Canny Edge [2] detection picture (figure 8). To compute this attribute of a door, we sum up the magnitude values $\nabla I(p)$ in the door bottom area:

$$\frac{1}{|A|} \sum_{p \in A} |\nabla I(p)| < T_{texture} \quad (5)$$

, where p corresponds to a pixel in the area. The sum is divided by the number of the pixels $|A|$. The decision is obtained by comparing this value to a threshold $T_{texture}$. It is important for this classifier to consider that the area has to grow, if the distance between the two vertical lines gets larger.

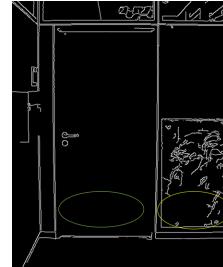


Fig. 8. Shows an image after Canny Edge detection. Green area: No texture in the bottom area of the door. Yellow area: Others objects have often textures in this area.

3.5 Door Knob

For the door knob classifier we use again the line image calculated for the detection of the vertical line pairs. However, for this classifier the vertical lines are

not important, but the almost horizontal lines, which are result from the door knob. The distance between the vertical line pairs is used to find the door knob area. In these two areas (left and right side of a door) the classifier returns a "true", if:

$$\sum (line_{almost_horizontal}) \geq T_{doorknob} \quad (6)$$

where:

$$T_{doorknob} = 2 \quad (7)$$



Fig. 9. Two example images, which show the door knob areas. Door knobs are detected if inside of these areas are at least two almost horizontal lines.

Image 9 shows typical detected door knobs using this weak classifier.

3.6 Door Frame

The door frame is required to install a door inside a wall. The frame can be calculated during the determination of the vertical line pairs. A door frame in an image is characterized by duplicated vertical lines. If there is an additionally vertical line on both sides of the door, the door frame classifier is true. Figure 10 presents the situation.

3.7 Door Width

There is a DIN standard (DIN18101) for the door width. Unfortunately the door width can strongly vary. Even here it is not easy to find a strong classifier for a door. But for a weak classifier we bordered the width in values between 0.735m (threshold for minimal width) and 1.110m (threshold for maximal width). To calculate the distance between the two vertical lines we use the laser distance data provided from the robot.



Fig. 10. The red arrows point to areas where duplicated vertical line pairs are found. The door frame classifier will be positive for these door candidates.

3.8 Door Concavity

In many environments doors are recessed into the wall, creating a concave shape for the doorway (illustrated by figure 11). This shape can be obtained using the robot laser distance data. For that the slope between each measured laser distance in the door candidate area is calculated. Figure 12 shows the slope between the different points for the sample laser profile. There exists a maximum and minimum at the position of the door candidate. The Concavity or U-Shape weak classifier can be described by the following rules:

- If we calculate the slope between each measured laser distance point, without considering the door candidate, the standard deviation is almost zero ($T_{standard_deviation}$).
- If we look at the slope at the door frame area we will find values which strongly vary from the calculated mean value.

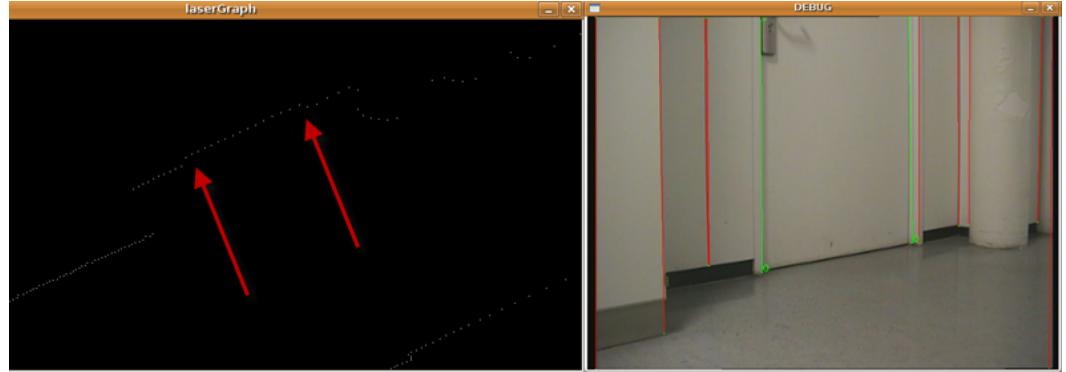


Fig. 11. The red arrows in the laser profile point to a door. The images show that the door is not running with the wall. It is receding or sticking out from the door.

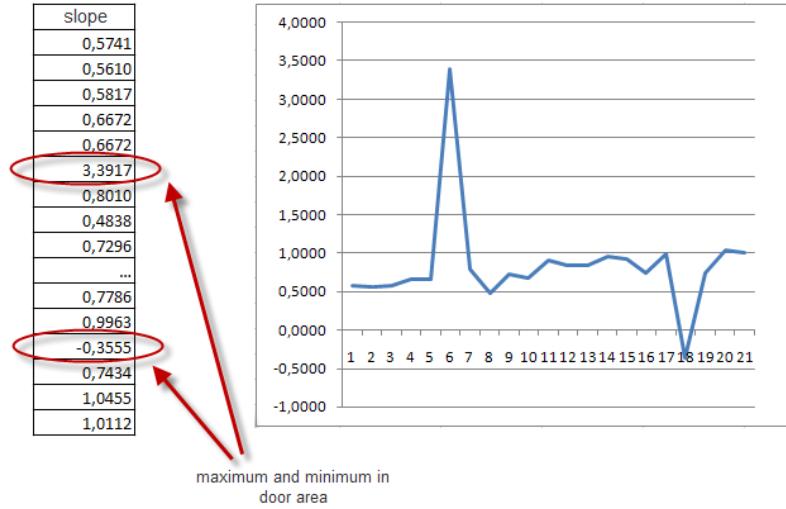


Fig. 12. Slopes between the measuring points from figure 11. We found turning points in the area of the door frame.

4 Experimental results

To test the performance of the system, a database of 210 test dates were taken with the Pioneer2DX. One test date consists of one camera image, one laser profile as well as the position of the camera at a certain time. We considered pictures of doors and non-doors. From the 210 test dates we took 140 for the training process of the AdaBoost. The residual of the 210 test dates were taken to test our system. In these 70 test dates our door candidates finder algorithm found overall 550 door candidates, where 61 candidates correspond to a real door. The result is shown in a ROC chart¹ (figure 13). As we can see in the chart, the AdaBoost classifier does not reach the best detection rate. In our test the true-positive rate for the AdaBoost classifier reaches a value of 0.72 and a false-positive rate of 0.0082. However, for an object detection system, where a small false-positive rate is more important than a high true-positive rate, the result of the AdaBoost classifier is the best. With a large probability the door will be detected anyway from the robot form another robot position. But a false detection would cause an unwanted detection of a door. We get the same result, if we take a look at the RPC methods² (table 1). The best value of the F-score (combination from precision and recall) is obtained by the AdaBoost classifier.

Typical detected doors are illustrated in figure 14. As can be seen, the algorithm is capable of detecting doors under different lighting situations and

¹ Receiver Operating Characteristic

² Recall Precision Characteristic

	Recall	Fallout	Precision	F-score
DoorWidthClassifier	0.9016	0.2413	0.31791	0.470
DoorConcavityClassifier	0.5737	0.1554	0.3153	0.4069
TextureBottomClassifier	0.7541	0.0920	0.5054	0.6052
ColorWallClassifier	0.0655	0.0348	0.1904	0.0975
DoorGapClassifier	0.5901	0.3149	0.1894	0.2868
DoorKnobClassifier	0.8988	0.1022	0.5145	0.6463
DoorFrameClassifier	0.4426	0.2965	0.1569	0.2317
AdaBoost Classifier	0.7213	0.0082	0.9166	0.80733

Table 1. Result of the RPC methods. The F-score can be interpreted as a weighted average of the precision and recall, where a F-score reaches its best value at 1 and worst score at 0.

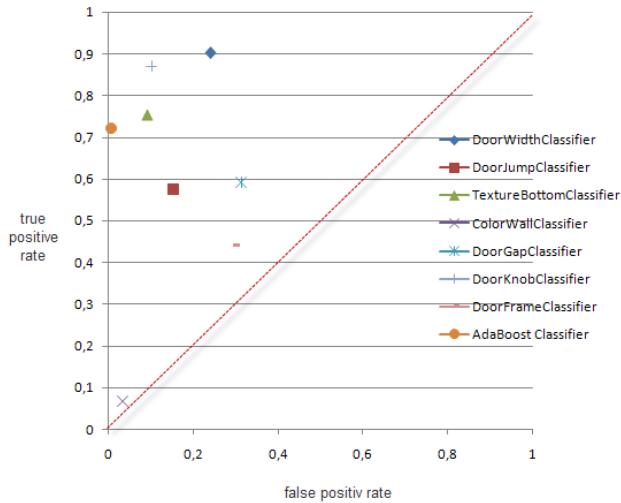


Fig. 13. ROC diagram of all classifiers. The best value is reached at coordinate (0,1). The AdaBoost classifier does not reach the best detection recall ration. However, for an object detection system, where a false detection could have undesired effects, the AdaBoost classifier is the best classifier.

different viewpoints of the robot. Also the absence of one or more door features does not cause a non detection of the door.

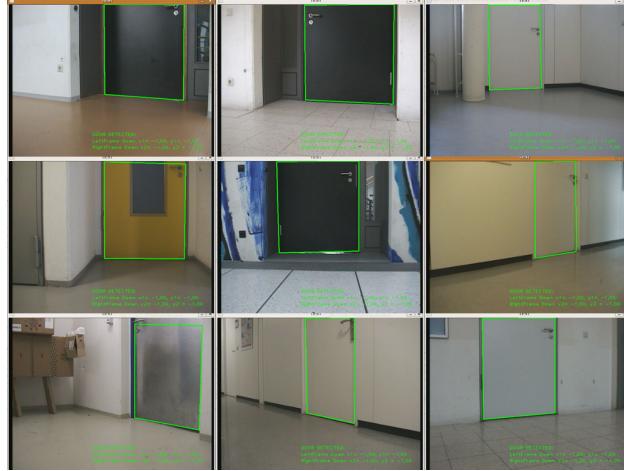


Fig. 14. Typical by the AdaBoost classifier detected doors. The pictures demonstrate that our approach is robust against different robot positions, reflection situation and as well as different door features.

False detections happen by candidates, which have a very similar appearance like a door e.g. in figure 15 a wall is detected as a door. In this example the DoorWithClassifier, DoorJumpClassifier, TextureBottomClassifier and the DoorKnobClassifier are positive.

In a next step we tested the system as a Player [4] driver on our Pioneer2DX robot. Hereby, we used two different environments. We also received a good result. In the first environment (basement of the university) all doors were detected, except two, which could not be measured by the laser system of the robot. The result of the first environment is shown in the figure 16. In the second environment (office environment) each door, beside glass doors, were detected. The problem here is, that we received wrong laser distances because the laser is going through the glass. Also the reflection on the ground of these doors is higher than on other doors.

5 Conclusion and future work

In this paper we presented an approach for a laser and camera-based door detection system. By using the AdaBoost algorithm we created a system with a detection rate with more than 72% and a very low error rate of 0.008. It is a combination of several weak classifiers e.g the color of the wall, door knob or door gap. In the result section we used the ROC and RPC methods to demonstrate

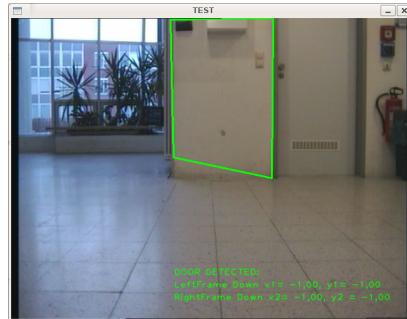


Fig. 15. Picture illustrates a sample false-positive error of the AdaBoost classifier. In the sample a wall, which looks similar to a door, is detected as door.

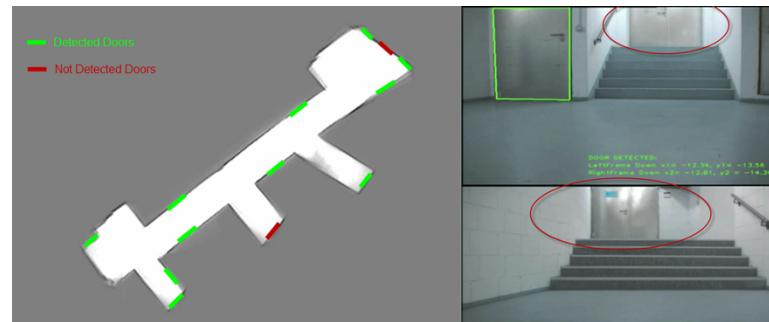


Fig. 16. The Image depicts the result of the robot test run in the basement environment. Detected doors are marked in the map. The doors that have not been detected by the algorithm (red ellipses in the second image) are in position which could not be measured by the laser.

that none of the other weak classifier can replace the strong classifier created by the AdaBoost algorithm. The system has the ability to find doors in realtime. With an Intel Core Duo 2.4GHz processor we reached a performance of 12fps. There are multiple possibilities to improve the system. Firstly, the training set can be enlarged. More test data would improve the alpha values for each weak classifiers. If we use the system in a new environment, it will provide a better result, if we add test data of this environment. Secondly, the weak classifiers can be modified and new weak classifiers can be added. E.g the ColorWallClassifier can be improved, if the system automatically learns the wall color of the environment. New classifiers could use the door hinges or the light switch on the door side. For future work it would be interesting to integrate this system in an autonomous map building system. That means that the robot has the ability to create a map of an unknown environment and mark doors in it. Secondly, the detection of a door plate would also be interesting to navigate the robot better through unknown environments. Furthermore, the detection of open doors with new classifiers would be interesting.

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 (ICRA)*, 2004.
2. John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.
3. Zhichao Chen and Stanley T. Birchfield. Visual detection of lintel-occluded doors from a singel image. *IEEE Computer Society Workshop on Visual Localization for Mobile Platforms*, 1(1):1–8, 2008.
4. Toby H. J. Collett, Bruce A. MacDonald, and Brian Gerkey. Player 2.0: Toward a practical robot programming framework. In *Australasian Conference on Robotics and Automation*, Sydney, 2005.
5. Yoav Freund and Robert E. Schapire. A short introduction to boosting. *J. Japan. Soc. for Artif. Intel.*, 14(5):771–780, 1999.
6. A. C. Murillo, J. Košecká, J. J. Guerrero, and C. Sagüés. Visual door detection integrating appearance and shape cues. *Robot. Auton. Syst.*, 5.
7. Jos Neira and Juan Domingo Trados. Computer vision. In *Universidad de Zaragoza*, Zaragoza, Spain, 2008.
8. Michael Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 1(1):11–32, 1991.