ADABOOST BASED DOOR DETECTION FOR MOBILE ROBOTS

Jens Hensler, Michael Blaich and Oliver Bittel

Laboratory for Mobile Robots, University of Applied Sciences, Konstanz, Germany

Keywords: Door detection, AdaBoost, Learning algorithm, Mobile robot.

Abstract: Doors are important landmarks for robot self localization and navigation in indoor environments. Existing algorithms for door detection are often limited to restricted environments. They do not consider the large intra-class variability of doors. In this paper we present a camera- and laser-based approach which allows finding more than 82% of all doors with a false positive rate less than 3% in static test sets. By using different door perspectives from a moving robot, we detect more than 90% of doors with a very low false detection rate.

1 INTRODUCTION

In an indoor environment doors constitute significant landmarks. They represent the entrance and exit points of rooms. Therefore, robust real-time door detection is an essential component for indoor robot applications (e.g. courier, observation or tour guide robots).

In the past, the problem of door detection has been studied several times. The approaches differ in the implemented sensor systems and the diversity of environments and doors, respectively. For example, in (Murillo et al., 2008) and (Chen and Birchfield, 2008) only visual information was used. Others, like (Anguelov et al., 2004) apply an additional 2D laser range finder and thereby receive better results.

From these approaches we find that there are two major difficulties in autonomous door detection. Firstly, it is often impossible to cover the entire door in a single camera image. In our scenario, the robot camera is close to the ground so that the top of the door is often not captured by the robot's camera (see figure 1).

Figure 1: Illustrates typical door images, taken by the robots 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.

The second difficulty is characterized by the large intra-class variability of doors (even for the same door types) in various environments. As shown in figure 1, doors can have different poses, lighting situations, reflections, as well as completely different features. The main features of a door are illustrated in figure 2. A door can be recognized e.g. by its color or texture with respect to the color or texture of the surrounding wall. Door gaps or door knobs are indicators of a door, too. Even if some of these features are detected in a single camera image, a robust algorithm should detect the door by using the remaining features.

In recent work (Chen and Birchfield, 2008), these two issues of door detection were solved by extracting several door features from the robots camera images and applying the AdaBoost algorithm (Freund and Schapire, 1999). The algorithm combines all weak features of a door candidate to receive a strong door classifier, which allows to decide whether a door is found or not.

For our situation this approach is not sensitive enough. We could not reach the same high detection rate in our complex university environment with a similar system (see section 4). Therefore, we add 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.
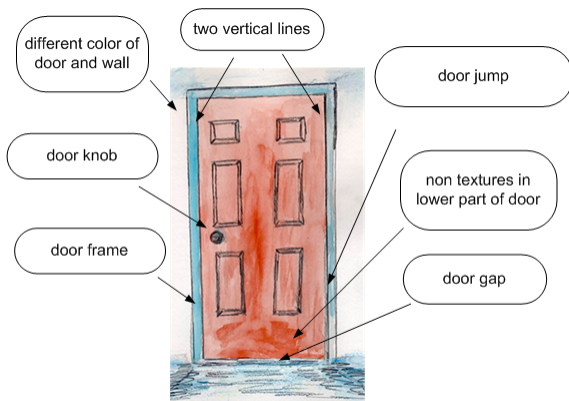
Figure 2: Characterizing features of doors.

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. Hereby, the weak classifiers are build up to solve binary decisions. The AdaBoost algorithm uses a training dataset to build up a strong classifier. For this purpose, the algorithm requires that each weak classifier reaches at least 50% success rate in the training process 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. The output of the nth weak classifier to the input x is $y_n = h_n(x)$. If every y_n is weighted with an α_n which is created during the training process, the strong classifier is given by:

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

3 DETECTION OF DOOR FEATURES

As mentioned before, we use the robot's camera image and the laser-based distance signal for door detection. Out of the camera image we extract vertical lines to find door candidates in the images. For this preselection 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 seven door features which represent the weak classifiers: a door can have a certain width *WidthClassifier*, the color of the door can be different from the color of the wall *Col-*

orWallClassifier, a door can have a texture at the bottom or not *TextureBottomClassifier*, a door may have a door frame *FrameClassifier* or a door knob *KnobClassifier*, also a door gap *GapClassifier* is possible; finally, the door can stick out of the wall *JumpClassifier*. The buildup of the weak classifiers is described in the sections below. Each classifier resolves a binary decision. The best threshold for each classifier is measured with ROC curves by varying the threshold until the best one is found. The classifiers *GapClassifier*, *ColorWallClassifier* and *TextureBottomClassifier* are similarly implemented like in (Chen and Birchfield, 2008) and not further mentioned here.

3.1 Preselection

During the preselection vertical line pairs generated by the door frame represent door candidates for the AdaBoost algorithm. To receive vertical lines we apply the Contour Following Algorithm (Neira and Tardos, 2008). Compared to other transformations, this method has the advantage, that we obtain the starting and end points of these lines.

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

- The vertical lines need to have a minimal length.
- 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 outer lines are indicators for a door frame.

3.2 Weak Classifiers

To improve our AdaBoost algorithm in comparison to (Chen and Birchfield, 2008) we use four additional weak classifiers. At first the door knob classifier will be explained. For this classifier we used again the line image calculated during the preselection of the door candidates. However, for this classifier not the vertical lines are important, but the almost horizontal lines which result from the door knob. We use a height from about 0.9m based on the bottom end of the vertical lines to find the door knob areas. In these two areas (left and right side of a door) the classifier returns 'true' if at least two almost horizontal lines are found.

The second classifier is a door frame classifier. A door frame is required to install a door inside a wall. The frame can also be calculated during the determination of the vertical line pairs. A door frame in an image is characterized by duplicated vertical line pairs. If there is a close additional vertical line on both sides of the door, the door frame classifier is positive.

Farther we use the door width to get one more weak classifier. There is a DIN standard (DIN18101) for door width. Unfortunately the DIN width values vary strongly. Even here it is not easy to find a strong classifier for a door. For a weak classifier we bordered the width in values between 0.735m and 1.110m. To calculate the distance between the two vertical lines we use the laser distance data provided from the laser range finder.

At last we consider that in many environments doors are receded into the wall, creating a concave shape for the doorway (see figure 3). This shape can be obtained using the robot laser distance data. For this the slope between each measured laser distance in the door candidate area is calculated. There exists a maximum and minimum slope value at the position of the door candidate (see figure 4). The *JumpClassifier* 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.

- If we look at the slope at the door frame area we will find values which strongly vary from the calculated mean value.

4 EXPERIMENTAL RESULTS

To test the performance of the system, a database of 210 test sets were taken with the Pioneer2DX robot. One test set consists of one camera image and one laser profile at a certain time. We considered pictures of doors and non-doors. From the 210 test sets we took 140 for the training process of the AdaBoost. The residual sets were taken to test our system. In these 70 test sets our preselection algorithm found overall 550 door candidates, of which 61 candidates correspond to real doors. The result for each weak classifier and the strong AdaBoost classifier is shown in a ROC space diagram (see figure 5). As we can see in the ROC space, the AdaBoost classifier reach the best detection rate. In our test the true-positive rate for the AdaBoost classifier reaches a value of 82% and a false-positive rate of 3%. We receive the same result if we take a look at the RPC methods (table 1). The best value of the F-score (combination from precision and recall) is obtained by the AdaBoost classifier.
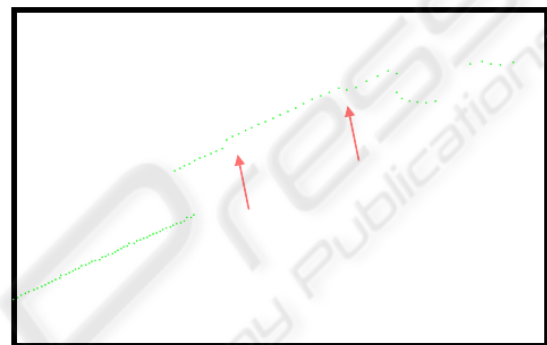

Figure 3: The red arrows in the laser profile point to a door. The images show that the door is not running with the wall. It is receding or sticking out from the door.
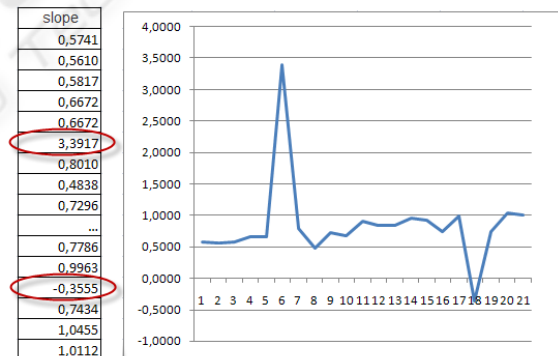

Figure 4: Slopes between the measuring points from figure 3. We found turning points in the area of the door frame. It's used for the *JumpClassifier*.

Typically detected doors are illustrated in figure 6. As can be seen, the algorithm is capable of detecting doors under different lighting situations and different viewpoints of the robot. It should be noted, that the absence of one or more door features does not cause a non-detection of the door. Figure 7 shows a false positive detection. They happen through walls or other objects which look very similar to doors.

As a next step, we looked at the result without the

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

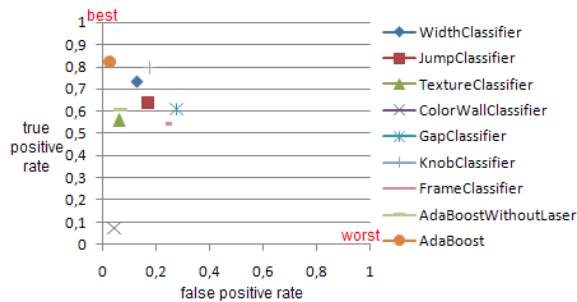|	Recall	Fallout	Precision	F-score
WidthClassifier	0,73	0,13	0,45	0,56
JumpClassifier	0,64	0,19	0,35	0,45
TextureClassifier	0,56	0,06	0,56	0,56
ColorWallClassifier	0,07	0,05	0,18	0,10
GapClassifier	0,61	0,35	0,24	0,34
KnobClassifier	0,80	0,19	0,39	0,53
FrameClassifier	0,54	0,29	0,25	0,34
AdaBoostWithoutLaser	0,61	0,07	0,56	0,58
AdaBoost	0,82	0,03	0,79	0,81

Figure 5: ROC space diagram of all classifiers. The best value is reached at coordinate (0,1). The AdaBoost classifier - the weighted combination from the other weak classifiers - reaches the best detection rate.

laser range finder (similar to (Chen and Birchfield, 2008), see table 1 and figure 5 *AdaBoostWithoutLaser*). This classifier combination (*TextureBottomClassifier*, *ColorWallClassifier*, *GapClassifier*, *KnobClassifier* and *FrameClassifier*) did not reach the same high result (detection rate 60% and false positive rate 7%). With this result we claim that in a strongly varying indoor environment with different kinds of doors a camera-based door detection is not strong enough to build up a powerful AdaBoost classifier. Further classifiers like the *JumpClassifier* and *WidthClassifier* can improve the result essentially.

Another advantage of the laser range finder is that the position of detected doors can be measured exactly. In combination with the robot position the doors can be marked in an existing map. The result is a map with doors as additional landmarks for improved robot localization and navigation.

We tested the system as a Player (Collett et al., 2005) driver on our Pioneer2DX robot. We used two different environments. In the first environment (basement of the university) all doors were detected (see figure 8). In the second environment (office environment) each door, except glass doors, was detected (see figure 9). The problem here is, that we

Figure 6: Typically by the AdaBoost classifier detected doors. The pictures demonstrate that our approach is robust against different robot positions, reflection situation as well as different door features.

Figure 7: 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.

received wrong laser distances, because the laser is going through the glass.

5 CONCLUSIONS 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 built a system with a detection rate of more than 82% and a very low error rate of 3%. It is a combination of several weak classifiers, e.g the color of the wall, door knob or door gap. We used the ROC and RPC methods to demonstrate that none of the other weak classifiers can re-
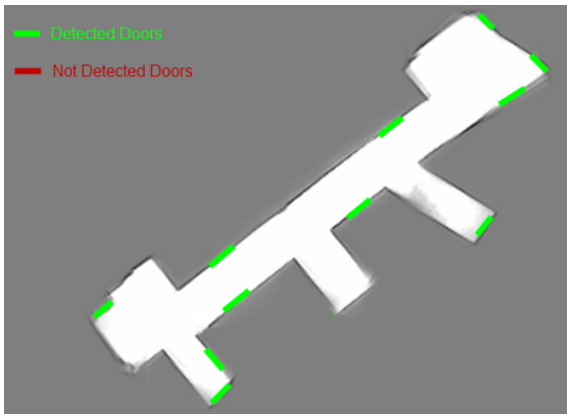
Figure 8: The image depicts the result of the robot first test run in the basement environment. Detected doors are marked green in the map. Each door was found.
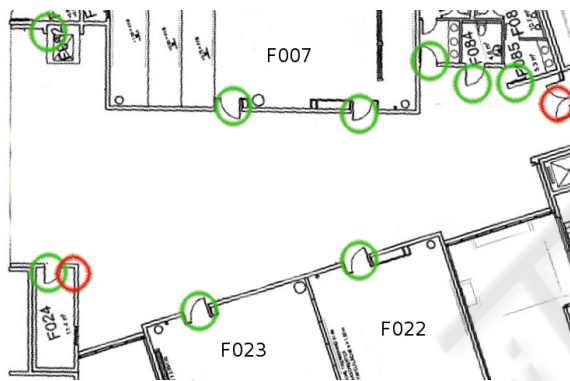

Figure 9: Results of the second robot test run in the office environment. All detections are marked with green circles in the map. The non-detection (glass door) and false detection (see figure 7) are marked with red circles.

place the strong classifier created by the AdaBoost algorithm. Furthermore it was shown, that without the laser range finder, we could not reach the same high detection rate. 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 several 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. Moreover, the detection of a door plate would be interesting to navigate the robot better through unknown environments. In addition, we should look for new classifiers, which allow to detect open doors.

REFERENCES

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

Chen, Z. and Birchfield, S. T. (2008). Visual detection of lintel-occluded doors from a single image. *IEEE Computer Society Workshop on Visual Localization for Mobile Platforms*, 1(1):1–8.

Collett, T. H. J., MacDonald, B. A., and Gerkey, B. (2005). Player 2.0: Toward a practical robot programming framework. In *Australasian Conference on Robotics and Automation*, Sydney.

Freund, Y. and Schapire, R. E. (1999). A short introduction to boosting. *J. Japan. Soc. for Artif. Intel.*, 14(5):771–780.

Murillo, A. C., Košecká, J., Guerrero, J. J., and Sagüés, C. (2008). Visual door detection integrating appearance and shape cues. *Robot. Auton. Syst.*, 5(6):512–521.

Neira, J. and Tardos, J. D. (2008). Computer vision. Universidad de Zaragoza, Spain.