Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Door Detection without Apriori Color Knowledge ⋆

Hai-Qiang Zhang, Li-Hua Dou, Jie Chen *

* *Department of Automatic Control, School of Information Science and Technology, Beijing Institute of Technology, Beijing, 100081 China (e-mail: haiqiang.zhang@gmail.com)*

Abstract: Most existent door detection methods assume that the panels' colors are known and rely mainly on this apriori knowledge. However, the color is not always available especially when the environment is entirely unknown. In this paper, a door was defined as a rectangular region with almost a homogeneous color and proper size which one can enter or exit through. The proposed door detection framework features no need for any apriori color knowledge. We adopt the determined finite automaton to model the process which starts from motion detection to obtain the potential door regions roughly, then a combined algorithm is used to extract stable and interest edges which bound the subsequent region growing, finally the door regions could be located accurately by fusing all the preceding results. Experiments under various environments show that this framework is effective and adaptive.

Keywords: door detection; determined finite automaton; mobile robots; image processing; edge accumulation

1. INTRODUCTION

Place recognition plays a pervasive role in the research of indoor mobile robots. If a robot owns the ability to recognize various kinds of places in the environment, its high-level reasoning and behavior control could be further developed, see Anguelov [2004]. Doors in the indoor environment are not only ubiquitous but also informative for robots localization, navigation, etc. A truly autonomous mobile robot should be capable of detecting and interacting with the doors in the environment. For example, in a vision guided mobile robot navigation task, see Cicirelli [2001], the doors in the environment were set as the targets for a robot to detect and reach, and in this way a robot can accomplish tasks such as exploration and surveillance.

Several approaches have been proposed to the door detection problem. Cicirelli [2003] adopted neural classifiers to detect principal sub-components of a door, and a validating algorithm to check whether they are in the proper geometric configuration of a door. Both color and shape of a door are used as apriori knowledge. Stoeter [2000] identified doors in cluster environment by detecting vertical stripes in the image. After a pipeline of regular image processing, the location of possible doors is calculated based on the expected dimensions of doors and the corridor parameters which are derived with vision or from a local map. Thus the door detection is relative to wall detection and the spatial relation between walls and the robot, which make it a bit more complex and difficult. Monasterio [2002] defined that an open door is a squared noisy rectangular segment in the image on the assumption that door panels are textureless. Though this definition worked well with the case of learning to traverse doors, it is not a common definition for a door. In Amir [1999], the color of the door panels is manually specified and used as apriori knowledge. An area with the similar color, the proper width and two vertical boundaries is labeled as a door. As only the color is considered, other door-like objects would be accepted as doors unduly. Wellington [1999] presented a method also combining edges and apriori color information to locate the doors. Edges are extracted and merged to separate line segments. If a set of segments satisfy predefined heuristics, such as they can form a U shape, the bounded region is considered as a candidate that is further tested with color. Anguelov [2004] made the door detecting and modeling process more sophisticated by making use of a laser range finder and a panoramic camera under a probabilistic framework. Besides shape and color, the motion property is considered. Optimized with an expectation maximization algorithm, this framework can extract walls, moving doors and static doors from the environment without apriori color information.

We can infer the common hypothesis used in the above literatures as follows: distinct vertical boundary, a certain width, a homogeneous color of the panels and apriori color knowledge except for Anguelov [2004]. Without loss of generality, we also take the first three hypothesis. However, the doors could be with any color, so if we took color as apriori knowledge, we have to trouble to specify and detect every color of the doors. Moreover, the color information might be unavailable when the environment is entirely unknown. So we define a door as a rectangular region with almost a homogeneous color and proper size which one can enter or exit through. No specified color occurs in this definition and the proposed door detection framework features no need for any apriori color knowledge. We adopt

⋆ This work is sponsored by the Beijing Education Committee Cooperation Building Foundation Project.

the determined finite automaton to model the process which starts from motion detection to obtain the potential door regions roughly, then a combined algorithm is used to extract stable and interest edges which bound the subsequent region growing, finally the doors can be located accurately by fusing all the preceding results.

2. THE MODEL OF THE DOOR DETECTION

We adopted the determined finite automaton (DFA) to model the door detection process. The DFA was defined according to the formation of a regular DFA as follows:

States $Q := \{S_1, S_2, S_3, S_4\}$ which means that the whole door detection process will switch among the four states: S_1-Start state, S_2-Motion analysis state, S_3-Edges accumulation state and S_4-Region growing state. A whole process which outputs a door successfully starts from S_1, passes through S_2 and S_3 orderly and ends with S_4.

Alphabet $\Sigma := \{I, M\}$. A scene image is denoted as I, and every state update takes the current I as one input and the corresponding motion analysis results M as the other.

Start States $q_0 := \{S_1\}$.

Accept States $F := \{S_1\}$. The DFA will return to this state when the process is finished or fails from other states.

Transition Functions The state diagram is shown in Fig. 1 and described in details as follows:

S_1 The DFA enters this state after initialization or a whole door detection process is complete. in this state, the DFA detects motions in images. Once a motion is confirmed, the DFA will enter S_2.

S_2 The DFA keeps detecting motions in the vision field, and processes the resultant motion regions with proper operations, such as eliminating and merging, to make motion sequences. This state will not come to its end until no motions occur for a long enough period of time, and the DFA enters S_3 if some valid motion sequences exist or S_1 otherwise.

S_3 Motion detection continues and takes the DFA back to S_2 if any motion is detected. Otherwise, the scene can be considered to be still. In this case, the DFA executes edge detection and accumulation continuously and enters the state S_4 when the predetermined loops are reached.

S_4 In this state, no more information from the scene pours into it. The DFA relies on the motion sequences from S_2, edges from S_3 and the last scene image grabbed before entering S_4 (denoted as I_0) to carry out region growing and generate the finial door detection results. The pixels close to each other and with the close color are clustered together and form a region. We describe the region growing in details as follows.

For each motion sequence, the first and last motion region of this sequence are taken as coordinate regions for a door to reside in. The next two steps are taken on each coordinate region for five times:

(1) Randomly select a seed point for region growing inside the coordinate region

(2) Start the stepwise region grow from the seed point on the constraint of result edges from S_3,and record the resultant regions which have proper geometric form,

Then we fuse all proper regions generated in the above steps to a final region that could be a door.

If two candidate regions come up from the first and last motion region respectively, they are merged on the condition that they are close to each other. If any proper region remains, it is accepted as a door.

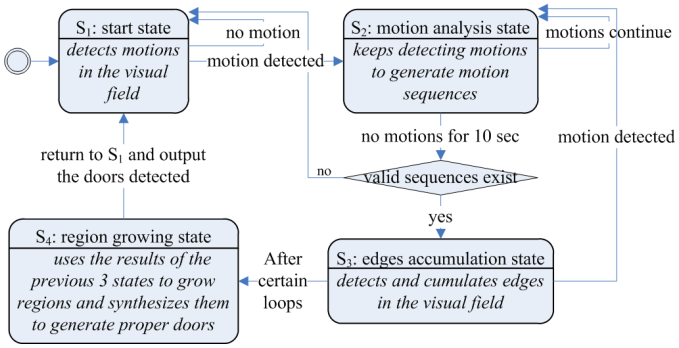

Fig. 1. The state diagram of the DFA

Briefly speaking, the DFA starts from the start state, grabbing the scene images continuously, and transforms its state according to the results of the door elements detection. If the observed scene has an appropriate door, the DFA will output the sub-image occupied by the door if runs successfully, otherwise, it will not output anything.

2.1 Motion Detection

The reason for bringing motion detection into door detection stems from the observation that: a door usually keeps still. If people walk through the door, this movement will generate a motion region sequence among images. For example, in the case that people open the door and move to the camera, the first region in the sequence will probably overlap with the door region and so will the last one if people leave and close the door behind them. Although other improper cases exist, we can extract useful information from the first and last element of the motion region sequence.

Motion detection takes gray level images as input and goes as follows.

Firstly, we calculate the frame difference of the last two images. Most pixels in the difference image were with non-zero values because of the glint of the lights, the reflections ,etc. Thus thresholding is applied and only the significant difference is reserved, as shown in equation (1)

$$tdf_{[i,j]} = \begin{cases} 255 & |fc_{[i,j]} - fp_{[i,j]}| > 20 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where fc denotes the current frame of the scene images, fp denotes the frame previous to fc, and $[i, j]$ are the row and column coordinates of a pixel.

Thresholding works well to filter disturbance in the image, but brings segment phenomenon. That is, if the overlaps of a moving object residing in two neighbor frames have the same color, they would be considered as still regions rather than parts of a moving object, thus the moving object would be separated. Fig. 2 shows such a case, where

the blue pixels indicate moving pixels, that the moving robot on the left-down corner was separated into a set of scattered regions. This problem will be handled later.

Secondly, in order to avoid using massive memories for the frames cache and speed up the motion detection, we record the motions occurred during a past period (denoted as MP) via a motion history image (denoted as $mhi_{[i,j]}$) which is the same size (320*240) as the scene images. The pixels of $mhi_{[i,j]}$ contain timestamps which indicate that they were moving pixels at that time. The update equation for $mhi_{[i,j]}$ is:

$$mhi_{[i,j]} = \begin{cases} t & tfd_{[i,j]} \neq 0 \\ 0 & tfd_{[i,j]} = 0, \text{and } t - mhi_{[i,j]} > MP \\ mhi_{[i,j]} & \text{otherwise} \end{cases} \tag{2}$$

where t is the time when the last processed image was grabbed.

Then mhi is clustered to generate motion segments: adjacent pixels are clustered together if the difference between their timestamps are below a threshold th, and the resultant regions which are too small, large or sparse are eliminated. Next, a recursive method is adopted to handle the segment phenomenon mentioned previously. If the rectangle with the minimum area that covers two motion regions mp_1 and mp_2, denoted as CR, satisfies (3), where W_R and H_R denote the width and height of the region R respectively, mp_1 and mp_2 are merged as CR.

$$\begin{cases} W_{CR} < W_{mp_1} + W_{mp_2} + min(W_{mp_1}, W_{mp_2}) \\ H_{CR} < H_{mp_1} + H_{mp_2} + min(H_{mp_1}, H_{mp_2}) \end{cases} \tag{3}$$

This association rule is too simple to handle the complex situations like there are several intersectant movements in the scene, but it works well for the current situation, because the image is not capable of containing several independent motion sequences at the same time. Fig. 3 demonstrates an example of reasonable merging, in which the white borders outline the motion regions. The over merged case exist when there are several moving objects in the view field and they are close to each other.

Fig. 2. Segments

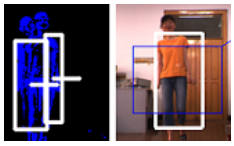

Fig. 3. Proper merging

Finally, the motion segments are assigned to one of the three motion region sequences maintained by the DFA following these rules: If a motion segment is close to the last element of one or more sequences, it is added to the closest sequence; otherwise, if there is any empty sequence, it is taken as the first element of this one, or else we clean up the shortest sequence and put it at the head. Fig. 5 demonstrates two motion sequences, where the letter S and E denotes the center of the first and last motion region in a sequence respectively, and each center of all the other segments in the sequence are denotes with a color point between S and E. The wine sequence is formed by someone entering and exiting later through the door, while the yellow one is the result of a robot moving around slightly.

2.2 Edge Accumulation

Edges provide abundant information of objects in an image. However, in a natural indoor environment, edges might be excessive and diverse from time to time due to the illumination, motion, reflection and so on. An example is shown in Fig. 4, an edge image obtained by the Canny algorithm. Although the parameters were manually setup to keep the edges of the door perfectly, the contour of the door is still difficult to find out. We present a method combining edge filtering and accumulation to extract the stable and interest edges from the natural scene for door detection.

The method is some kind of specific and goes as follows. Firstly, we convert the scene image from RGB color space to HSV space, and apply Canny edge detector on the S component. Then, we use the assumption that the door edges are generally close to horizontal or vertical, so we adopted Hough transform to find out the lines in the resultant edge image that lies between -5 °and +5 °in the horizontal or vertical directions. Only the edge pixels that lie on those lines are reserved. In this way, we filter out the uninterested edges.

We apply the steps above on each newly grabbed image and accumulate the resultant edges images continuously for a certain times N. The final binary edge image is obtained by a fixed thresholding $N/3$ on the accumulation image. The red pixels in Fig. 5 were the final edge pixels. It's perceptible that only the obvious and stable edges hold, which provide the exact doorframe information we need for door detection.

Fig. 4. The results of Canny

Fig. 5. Edges and motion sequences

2.3 Region Growing

Without loss of generality, we assume that the panels of a door are almost the same color, so if we can find a seed pixel that resides inside the door and start region growing under the constraints of the edges obtained from section 2.2, the region of the door can be extracted effectively.

The whole process of region growing is a little bit complex, because we prefer few reliable doors to many false doors. The rule is that if the three RGB values of a certain pixel $I_{(x,y)}$ satisfy $|I_{(x,y)} - I'_{(x,y)}| \leq th$, where th is a threshold and $I'_{(x,y)}$ is among the four-neighbors pixels of $I_{(x,y)}$ and is among a growing cluster, $I_{(x,y)}$ is labeled and merged into this cluster. When the labeling is stopped, the candidate region is the minimum rectangle that covers all the labeled pixels. Finally, if the candidate region satisfies

the basic constraints, that is it is of the proper size and the ratio of labeled pixels to all pixels in the region is over a predetermined percentage, it is accepted as a proper region.

For each of the three motion sequences, if it is long enough and generated by a moving person, we took the following steps on the first and last motion regions for five times:

(1) Randomly select a pixel from the region as the seed point which should not be on the mask edges,
(2) Perform region grow inside the color image with the RGB threshold increasing from 1 to 20. All the proper regions generated in the 20-time loop are recorded, and then find the minimum rectangle that covers them. If the rectangle satisfies the basic constraints, it is taken as a door candidate.

All the candidates generated in this five-time loop are merged and checked further to form the candidates associated with the first and last motion region respectively. If two candidate regions come up from the first and last motion region respectively, they are merged on the condition that they are close to each other. If any proper region stands, it is accepted as a door.

3. EXPERIMENTS

The proposed framework was tested under various scenes. Fig. 6 shows four of them and the corresponding door detection results. In each of the four images, the area indicated by the rectangle of white borders is considered as a door; the results of edges accumulation are expressed by red line segments, and the orange rectangles mark the first or last motion regions in a certain valid motion sequence; the orange blocks are snapshots of the region growing processes.

As we can see, the edges in the top-left and the bottom-right scenes are much heavier than the other two cases. These differences mainly stem from the diverse reflectance, while the parameters used in Hough transform and edge accumulation are the same. As a result, the door in the top-left image covers only the pane of the door, while the one in the top-right image covers both the pane and the glass(the black part) that resides in the doorframe. The same phenomena can bee seen in the bottom two images.

In despite of the aforementioned differences in the door detection results, the doors were rather satisfactory visually and this process goes well under variant regular scenes. The accuracy of the door detection results is hard to debate, because the true doorframe is blurred in the imaging process. With a view to the purpose of door detection, it is reasonable to accept the results shown in Fig. 6 as being accurate enough.

The process runs on the onboard PC of a Pioneer®-3 robot with a Pentium®3-800MHz CPU, a 128MB memory and runs on the Linux operating system. The imaging device is a digital stereo vision camera named Bumblebee™. The process runs about 3 fps, and takes 18% of the memory and very little CPU time. The frame rate is low because some stereo processes are ongoing together with the door detection. So we can embed this detection process into a mobile robot with little performance reduction.

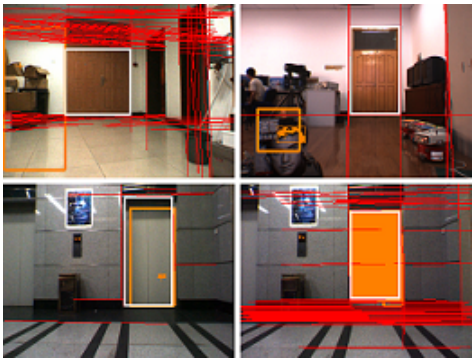

Fig. 6. The door detection results under four scenes

However, the process failed sometimes. For example, the door was dissevered by its own texture, or the edges were too weak to limit the region growing effectively, or the color component we choose was not suitable for image processing. These cases exist because there are so many magic parameters inside the process, and it is difficult to make all of them work adaptively. Still, we made a pilot study on the fixed parameters as follows.

There are 27 parameters in all. Five of them are relative to the system configurations, such as maximum door numbers, image buffer size and the times for edge accumulation operation. These parameters effect the door detection framework slightly. Another nine parameters are for the motion detection and the door's geometry, for example, the range of motion segment area, the minimum ratio of valid motion pixels and the range of the door's width and height. These parameters play an important role in validating the door detection results, however, they are closely correlative to the imaging geometry. Thus it is unlikely to make them adaptive in large-scale. The left thirteen parameters are the most basic and critical ones which are used in the motion detection and analysis module, the Hough transform and Canny edge detector module and the region growing validation module. As these parameters work bad when the illumination makes against these image processing techniques, the framework will fail to find doors in the scene.

Generally speaking, it is hard to present a framework that works well under any condition. We will try more sophisticated image processing techniques and framework control methods to make all the parameters work together better, and the process can work under more scenes.

4. CONCLUSION

This paper presents a new sophisticated framework of door detection in natural indoor environment with no need for any specified color information as apriori knowledge. Without loss of generality, the door was defined as a rectangle region with almost a homogeneous color and proper size which one can enter or exit through.

We use a frame difference based algorithm for motion detection and a simple association method for assigning motion region to a certain motion sequence. The first and/or last motion region in a sequence are expected to overlap with the region occupied by a door. We rely on the stable and straight edges in the vertical or horizontal

directions to bound a door region. These edges are initially obtained via Canny edge detector, and filtered to reserve the horizontal or vertical ones which are accumulated to generate the desired edges. The final region growing starts from a randomly selected point in the candidate region, and clusters the pixels with a similar color under the bound of the edges. The resultant regions were checked whether they were with the proper size and merged if necessary. If any region remains, it is accepted as a door. The region growing works in a stepwise way with the threshold stepping up from 1 to 20. This is a necessary strategy to obtain the best region growing result when working under natural indoor environment. The whole process was modeled by the determined finite automaton which is proved suitable for connecting and fusing the elements detection results.

Experiments under various environments show that this framework is effective and wild adaptive, and the final door location is accurate. However, the framework is composed of many image processing techniques, so there are quite a few magic parameters. The framework may failed to detect the door in vision field if one of them does not fit the current scene. More work is needed to make them adaptive and work together better.

ACKNOWLEDGEMENTS

We would like to thank to all the members of the Intelligent Mobile Robot Group at Beijing Institute of Technology for their fruitful insights and commentary.

REFERENCES

Dragomir Anguelov, Daphne Koller, Evan Parker, and Sebastian Thrun. Detecting and modeling doors with mobile robots. *IEEE International Conference on Robotics and Automation(ICRA)*, 4:3777–3784, 2004.

G. Cicirelli, T. D'Orazio, and N. Ancona. Door detection in images based on learning by components. *Proceedings of SPIE*, 4572:304–309, 2001.

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

Sascha A. Stoeter, Frederic Le Mauff, and Nikolaos P. Papanikolopoulos. Real-time door detection in cluttered environments. *Proceedings of the 15th IEEE International Symposium on Intelligent Control(ISIC 2000)*, 187–192, 2000.

Inaki Monasterio, Elena Lazkano, Inaki Rano, and Basilio Sierra. Learning to traverse doors using visual information. *Congress of Intelligent forecasting, fault diagnosis, scheduling and control*, 60:347–356, 2002.

Eyal Amir, and Pedrito Maynard-Reid. *http://www-formal.stanford.edu/eyal/cs223b/introduction.html*.

Carl K. Wellington, Roger A. Bock, and Bruce A. Maxwell. Combining color and edge information to find door locations in an image. *Proceedings of SPIE*, 3837:185–193, 1999.