Recognition of Objects with Legs Using Model Information and Image from Tilted Camera

Tae-Bum Kwon
Korea University, Dept. of Mech. Eng.
haptics@korea.ac.kr

Jae-Bok Song
Korea University, Dept. of Mech. Eng.
jbsong@korea.ac.kr

Abstract − A mobile robot encounters various objects in the real environment, and it should recognize and exploit objects for successful navigation. Therefore, perception is the most important ability for many navigation techniques such as localization, obstacle avoidance, HRI, and so on. It is, however, very difficult to detect some objects featured by thin legs using only a 2D range sensor, the most popular sensor type for navigation. In this case, a vision sensor is more suitable than a range sensor for perception, but a monocular camera cannot provide the range data, which is essential for the use of the recognized object. To overcome this drawback, this paper proposes a new method to detect the objects with legs using the model information and a tilt camera. In this scheme, a robot detects the candidates for legs in the camera image, generates many possible candidates for object pose (position and orientation), and selects the best candidate by evaluating all candidates using visual information. Various experiments in the real environment showed the proposed scheme was effective in recognition of the objects with legs.

Keywords − Object Recognition, Navigation.

1. Introduction

Several techniques such as mapping, localization, path planning, and obstacle avoidance are required for mobile robot navigation in the real environment. These fundamental and essential techniques basically use the perception ability of a robot. For example, a robot perceives the environment using its sensors and then it can localize itself and avoid obstacles using the perceived information. Therefore, perception is the most basic and important ability for robots including a mobile robot.

Perception performance is related to several factors including sensing ability, types of obstacles, and perception scheme. The sensing ability is limited by the sensor type. For example, a range sensor such as a laser rangefinder, one of the most popular sensors for navigation, provides very accurate and stable 2D range data while a vision sensor such as a stereo camera offers rich color and 3D information. The type of obstacle and the sensing ability are closely related to each other. For example, a laser scanner cannot sense some materials such as a mirror or glass while a sonar sensor can detect them. In addition, a range sensor can generally sense 2D obstacles, whereas a vision sensor can sense 3D obstacles within its field of view. These two factors, sensing ability and types of obstacles, are related to the hardware and environment, and therefore they are not easy to deal with. Therefore, the remaining factor, the perception scheme or algorithm should be improved to overcome the limitations of the other two factors and improve the navigation performance.

This research proposes a perception scheme to detect some objects such as tables and chairs which are frequently encountered during indoor navigation, as shown in Fig. 1. These types of obstacles are important for navigation. For example, a robot localizes itself using these objects, avoids them, stops in front of them to interact with the objects on it, and so on. To perceive these objects in the real environment using a range sensor is, however, a very hard task because this type of sensor usually detects only the legs of a table and a chair. Therefore, a mobile robot may easily fail to exploit these obstacles unless it is able to detect them well.

Fig. 1. Important but hardly detectable objects in real environment.

Several ways of detecting these objects have been studied so far. The method for detecting a door and a table using line features of images were proposed in [1]. In this research, the lines extracted from the image were compared with the models which were made by a user. However, the objects should be placed in a standard pose and very close to the wall to prevent many lines from being extracted between the object and the background.

The information about the color and 3D shape of the objects was used in [2]. During navigation, a robot generated an image using the 3D information of objects, and compared that image with the image obtained by a camera. This method was not practical because of two drawbacks. First, it was assumed that the poses of both a robot and objects were known. Second, it compared all corresponding pixels by color between the images obtained by a camera and generated using the object models. Thus, it was inefficient and could not deal with the sensor uncertainty.

Cue-based object tracking methods were proposed in [3] and [4]. The various cues such as edges and normal vectors extracted from the wireframe models were used to compare the object candidates with the image and to track that object. Moreover, to robustly track the object, this method provided the self evaluation method based on the topological relationship among the cues. In the same way as other vision-based tracking methods, however, it has the initialization problem, and therefore the position and orientation of the object to track was defined manually on the image.

This paper aims at developing the monocular camera-based object recognition scheme. The proposed scheme recognizes some objects such as a table and a chair, which are difficult to detect by only range sensors because they usually have thin legs and relatively bulky body. In the proposed scheme, the leg of an object is detected and exploited to generate the candidates of objects for solving the initialization problem, and both model information and edges extracted from the real image are continuously compared to track the object reliably. While moving in the real environment, more than one object can appear and disappear on the image and they are autonomously recognized and tracked by this approach. Moreover, a tilted camera model is analyzed and it makes the proposed scheme more applicable to various types of robots having a camera with or without a pan-tilt unit.

The remainder of this paper is organized as follows. Section 2 presents the overall structure of the proposed scheme. Section 3 explains how to extract the vertical segment from the image obtained from a tilted camera. Section 4 describes how to generate the candidates for the legs and objects. Section 5 deals with how to choose the best candidate among several candidates using vision data. Finally, section 6 presents conclusions.

2. Overall Structure

The proposed scheme in this research uses only a monocular camera for reliable object detection. A range sensor such as a laser scanner or an IR scanner can sense the leg part of an object which may be a common but important object for navigation, say, chair, and so on. It is difficult and unreliable, however, to detect legs using a range sensor, to estimate the whole shape or volume of an object based on the range sensor data, and to utilize the detected object for navigation. On the other hand, vision data can offer better candidates for leg positions and object poses than range data. In this case, an almost infinite number of candidates are possible because an accurate range to an object cannot be sensed with a monocular camera, and thus the model information is required to generate the candidates that are similar to the real objects. With the aid of the model information, the number of candidates can be drastically reduced. Moreover, the proposed scheme can take the tilt angle of a camera into consideration to estimate the pose of an object with a tilted camera, which can increase the capacity of this scheme for various applications and robot platforms.

All candidates for objects should be evaluated to choose the best one. One of the most popular and reliable low-level features is the edge which can be easily obtained by a monocular camera. Hence, the edges are used to evaluate the generated candidates in this research. The edges extracted from the image are used to compare the candidates with the real objects. After calculating the similarity between each candidate and the vision data, the best candidates are selected and the object poses are estimated. Figure 2 shows the overall structure of the scheme.

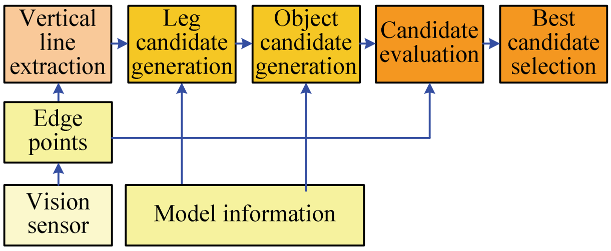

Fig. 2. Overall structure of proposed scheme.

3. Extraction of Vertical Line Segment

3.1 Coordinate Systems with Tilt Angle

When a tilt (or pan-tilt) camera is used for object recognition, the vertical segment of an object can serve as a useful basis for predicting an object leg. However, the vertical line segments of a 3D object are not projected onto the vertical lines in the 2D image. For example, in Fig. 3(b), the vertical segments of legs of a table are seen as the inclined lines in the 2D image.

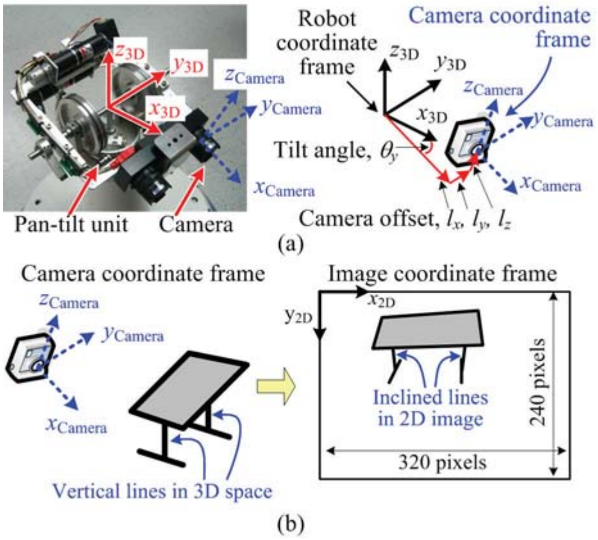

Fig. 3. Camera with pan-tilt unit; (a) geometrical relationshilp between fixed and moving camera coordinate systems, and (b) camera and image coordinate systems.

Figure 3(a) shows a stereo camera system with a pan-tilt unit (but in this research, only one camera was used.). The robot coordinate frame is attached to the center of a mobile robot, and the pan-tilt unit operates with respect to this frame. A camera coordinate frame is attached to the camera and its position and orientation change by the pan-tilt unit. The image is obtained from this camera frame. The camera offsets l_x, l_y, and l_z represent the distances from the robot frame to the moving camera frame along the axes x_{3D}, y_{3D}, and z_{3D}, and in these experiments, l_x, l_y, and l_z are 120mm, 45mm, and 22mm, respectively. One point in 3D space, $\mathbf{X}_{3D}$, is described in the camera frame as $\mathbf{X}_{3D,\text{Camera}}$, and it can also be described with respect to the camera coordinate as follows.

$$\mathbf{X}_{3D,\text{Camera}} = {}^{\text{Camera}}\mathbf{T}_{\text{Robot}} \cdot \mathbf{X}_{3D} \tag{1}$$

$$\begin{aligned} {}^{\text{Camera}}\mathbf{T}_{\text{Robot}} &= \text{Trans}(-l_x,-l_y,-l_z)\cdot\text{Rot}(0,-\theta_y,0) \\ &= \begin{bmatrix} \cos(-\theta_y) & 0 & \sin(-\theta_y) & -l_x \\ 0 & 1 & 0 & -l_y \\ \sin(\theta_y) & 0 & \cos(-\theta_y) & -l_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2}$$

where Trans() and Rot() are the translational and rotational matrix, and θ_y is the tilt angle. Using these equations, the relationship between the 3D space and 2D image can be obtained in consideration of a tilt angle as follows.

$$\begin{aligned} x_{2D} &= \\ &\frac{W_{\text{image}}}{2}\cdot\left(1-\frac{y_{3D}-l_y}{x_{3D}\cdot\cos(\theta_y)-z_{3D}\cdot\sin(\theta_y)-l_x}\cdot\frac{1}{\tan(FOV_{\text{H}})}\right) \\ y_{2D} &= \\ &\frac{H_{\text{image}}}{2}\cdot\left(1-\frac{x_{3D}\cdot\sin(\theta_y)+z_{3D}\cdot\cos(\theta_y)-l_z}{x_{3D}\cdot\cos(\theta_y)+z_{3D}\cdot\sin(-\theta_y)-l_x}\cdot\frac{1}{\tan(FOV_{\text{V}})}\right) \end{aligned} \tag{3}$$

where W_{image} and H_{image} are the width and height of the image, respectively (e.g., 320 and 240 for 320*240 image), and FOV$_V$ and FOV$_H$ are the fields of view in the vertical and horizontal directions, respectively.

3.2 Vertical Line Segment Projected on Image

Figure 4 shows the vertical lines extracted by the Hough method offered in the OpenCV library. Avertical line in 3D space is still vertical on the image as shown in Fig. 4(a) when it is projected with a tilt angle of 0°. It is, however, projected as an inclined line on the image when a tilt angle is not 0° as shown in Fig. 4(b). The relationship between the vertical segment in 3D space and the inclined line projected on the image can be found by differentiating Eq. (3) with respect to z_{3D} which varies along the vertical line segment in 3D space as follows:

$$\begin{aligned} \frac{\partial x_{2D}}{\partial z_{3D}} &= -\frac{W_{\text{image}}\cdot(y_{3D}-l_y)\cdot\sin(\theta_y)}{2(x_{3D}\cdot\cos(\theta_y)-z_{3D}\cdot\sin(\theta_y)-l_x)^2\cdot\tan(FOV_{\text{H}})} \\ \frac{\partial y_{2D}}{\partial z_{3D}} &= -\frac{H_{\text{image}}\cdot(x_{3D}-l_x\cdot\cos(\theta_y)-l_z\cdot\sin(\theta_y))}{2(x_{3D}\cdot\cos(\theta_y)-z_{3D}\cdot\sin(\theta_y)-l_x)^2\cdot\tan(FOV_{\text{V}})} \end{aligned} \tag{4}$$

Using Eq. (4), the slope of the projected lines of the vertical segments on the image can be expressed by

$$\begin{aligned} \alpha_{\text{line}} &= \frac{\partial y_{2D}}{\partial x_{2D}} \\ &= \frac{H_{\text{image}}}{W_{\text{image}}}\cdot\frac{\tan(FOV_{\text{H}})}{\tan(FOV_{\text{V}})}\cdot\frac{x_{3D}-l_x\cdot\cos(\theta_y)-l_z\sin(\theta_y)}{(y_{3D}-l_y)\cdot\sin(\theta_y)} \end{aligned} \tag{5}$$

where α_{line} is the angle of the extracted line with respect to the x_{2D} axis and it can be easily calculated on the image. This equation is exploited for generation of leg candidates in section 4.3.

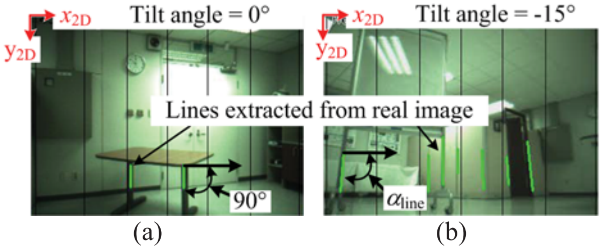

Fig. 4. Results of line extraction from real image; (a) vertical lines extracted from vertical segments in 3D space, and (b) inclined lines extracted from vertical segments in 3D space.

4. Model-based Candidate Generation

4.1 Model Information

If a robot has no information on the object to detect, the entire area over which the object can be located should be searched. In this case, it is difficult for a robot to detect the object in real-time due to the computational complexity. To overcome this difficulty, the proposed method uses the model information on the object. A wireframe model, one of the most popular model types, is employed. Figure 5 shows some examples of the models used in the experiments.

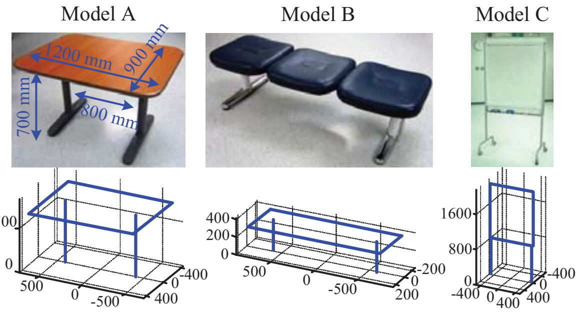

Fig. 5. Examples of some objects and their wireframe models.

4.2 Review of Similarity Range

The object in the image is usually somewhat different from the wireframe model in the actual experiments for several reasons. Since the wireframe model is a simplified model, the image can be distorted to some extent, and feature extraction is not perfect. These factors should be considered when the candidates are generated and evaluated. To this end, we introduced the concept of *similarity range* which is denoted as d_{sim} (in pixels) in our previous work [5]. Suppose the dotted line in Fig. 6(a) represents a candidate. Then the area which encloses the candidate and has a width of $2d_{\text{sim}}$ (in pixels) is selected as the *similarity evaluation area*. The edges within the similarity evaluation area can be used to evaluate the similarity between the candidate and the edges extracted from the real image.

Figure 6 is an example. If the edges are contained in this evaluation area as shown in Fig. 6(b), then the candidate is likely to represent the real object with a small error in the object pose, while the candidate in Fig. 6(c) is not similar to the real object. In this approach, to generate candidates too densely is ineffective because more than two candidates may be considered a similar one if they are closer than the similarity range. The candidate generation and evaluation processes using this concept will be explained in detail in section 4.3 and 5.

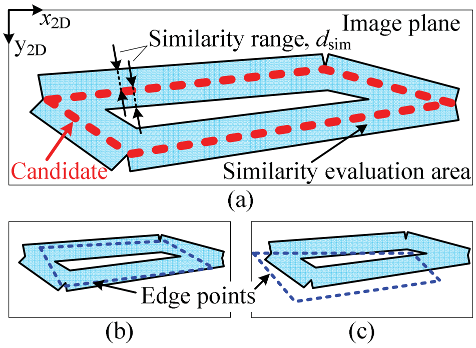

Fig. 6. Concept of similarity evaluation range and area [5].

4.3 Leg Candidate Generation

The vertical line segments of 3D objects extracted from the image can be considered as the possible candidates for legs. To generate the accurate candidates for legs, the direction and distance from the camera to the legs should be known, but only the direction can be obtained using a monocular camera. In this case, many candidates for legs can exist along the direction to the vertical line segments.

Figure 7 is an example of leg candidate generation associated with model C in Fig. 5. Figure 7(a) show the camera image. Seven lines are extracted and these lines can be considered possible legs of a white board. If the parameters related to the camera and the tilt angle are known and α_{line} is calculated on the image, the relationship between x_{3D} and y_{3D} for the directions of legs in 3D space can be expressed using Eq. (5) as follows:

$$y_{3D} = cx_{3D} - c(l_x \cdot \cos(\theta_y) + l_z \cdot \sin(\theta_y)) + l_y$$
$$c = \frac{H_{\text{image}}}{W_{\text{image}}} \cdot \frac{\tan(FOV_{\text{H}})}{\tan(FOV_{\text{V}})} \cdot \frac{1}{\alpha_{\text{line}} \cdot \sin(\theta_y)} \tag{6}$$

Using Eq. (6), the directions of the possible legs in Fig. 7(a) can be calculated and they are depicted in Fig. 7(b) which shows the environment within the horizontal FOV from the camera position to the available range (4m). All leg candidates which can be generated based on both the seven directions and model C are also depicted as circles in Fig. 7(b). Each pair of legs of model C is used to estimate the 3D pose of model C, as shown in Fig. 7(c).

A method for generating the candidates for legs is very important in the proposed scheme, and the same approach that we proposed in our previous work is also used [5]. This method will be shortly explained below using an example environment shown in Fig. 8 and more explanations were detailed in the reference [5]. From the three vertical lines extracted in the image, three possible directions for legs can be considered. Suppose direction 2 is considered in association with model A. The variable l_i denotes the location of leg i and $p_{i\text{-}j}$ represents the location of candidate pair j of model A with leg i as one leg. For example, l_3 is one of the possible legs and it can generate four pairs, $p_{3\text{-}0}$, $p_{3\text{-}1}$, $p_{3\text{-}2}$, and $p_{3\text{-}3}$. d_i represents the distance between two successive candidate legs, l_i and l_{i+1}. To find the more accurate pose of the object, more candidates should be generated. Since the computational burden is proportional to the number of candidates, however, the selection of the optimal value of d_i is of great concern in implementing the proposed scheme.

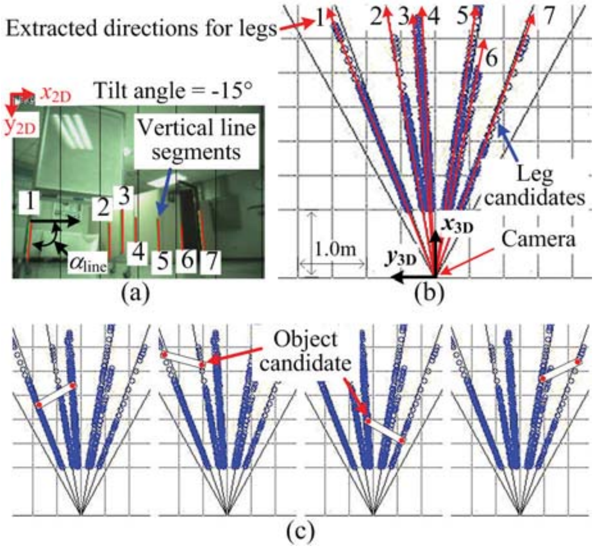

Fig. 7. Example of candidate generation; (a) raw image and extracted vertical line segments, (b) extracted directions for legs and some leg candidates, and (c) examples of object candidates.

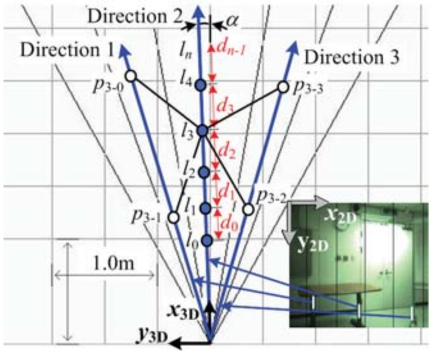

Fig. 8. Example of process of leg candidate generation.

All leg candidates located along one direction in 3D space may be overlapped on one line on the image, and therefore their x values on the image, x_{2D}, are identical. The y value, y_{2D}, however, is not identical when the position of a leg candidate changes along one direction. Both x and y values of a candidate are constrained by Eq. (6), and therefore the following equations can be obtained by differentiating Eq. (3).

$$\frac{\partial y_{2D}}{\partial x_{3D}} = -\frac{H_{\text{image}}}{2\tan(FOV_{\text{V}})} \cdot \frac{-z_{3D} - l_x \sin(\theta_y) + l_z \cos(\theta_y)}{(x_{3D}\cos(\theta_y) - z_{3D}\sin(\theta_y) - l_x)^2} \tag{7}$$

The *y* position of the projected leg and object on the 2D image changes with respect to the *x* position of a candidate in 3D space by this equation. Using this relationship between y_{2D} and x_{3D}, the interval, d_i, can be calculated for a given similarity range, d_{sim}. To generate the candidates at regular intervals, d_{sim}, on the image, the following relation should hold.

$$d_{\text{sim}} = d_i \cdot \frac{\partial y_{2D}}{\partial x_{3D}} \tag{8}$$

By substituting Eq. (7) into Eq. (8), d_i is calculated by

$$d_i = \frac{d_{\text{sim}}}{dy_{2D}/dx_{3D}} = \frac{2d_{\text{sim}} \cdot \tan(FOV_{\text{V}})(l_{i,x_{3D}}\cos(\theta_y) - l_{i,z_{3D}}\sin(\theta_y) - l_x)^2}{H_{\text{image}}(l_{i,z_{3D}} + l_x\sin(\theta_y) - l_z\cos(\theta_y))} \tag{9}$$

where $l_{i,x3D}$ and $l_{i,z3D}$ are the *x* and *z* coordinates of leg *i*.

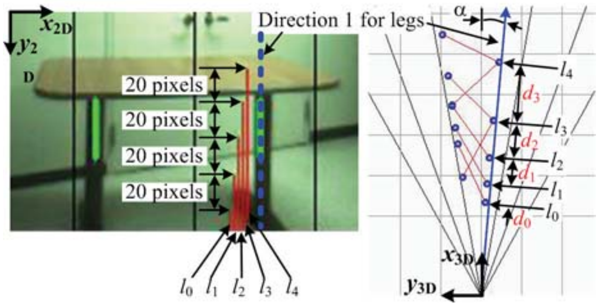

Fig. 9. Similarity range and interval of leg candidate generation.

Figure 9 shows an example of leg candidate generation. Along direction 1, several legs, l_0~l_4, are generated by Eq. (9) with the similarity range of 20 pixels. Five vertical lines in the left image show the projected image of the generated leg candidates. The differences between the values of y_{2D} of all successive candidates are 20 pixels, the similarity range. The smaller the similarity range is, the more leg candidates are generated. All legs are projected onto the same vertical line, the dotted line, because they are located in the same direction. In the left figure of Fig. 9, however, they are drawn side by side to show each leg clearly.

4.4 Object Candidate Generation

Each leg candidate generates an object candidate in 3D space, and it has its own shape transformed into the image plane by Eq. (3). Figure 10 shows some examples for model A in Fig. 5. For example, candidates 1 and 2 are generated by leg candidate l_4 in Fig. 9. These poses in the 3D coordinates are transformed into the 2D image coordinates as shown in Fig. 10(b), and they are stored in memory as the object candidates associated with model A.

Candidate generation for legs and objects are easily implemented based on geometry. The detailed process for model A were described in this section, and the process for other models are almost identical. If a robot wants to detect a new object, only candidate generation process for the new model is required and many potential objects which a robot encounters during navigation can be easily recognized.

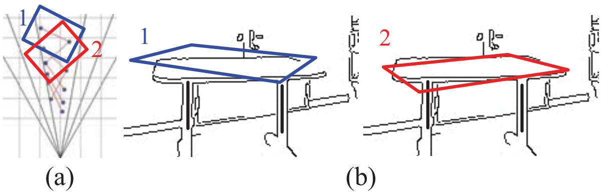

Fig. 10. Examples of object candidate generation.

5. Vision-based Candidate Evaluation

After candidate generation, every candidate should be evaluated to determine how similar it is to the real object. The lines extracted from the image are good features for comparing the candidates with the sensed environment because the wireframe model consists of lines. The line-based evaluation is, however, difficult and unreliable in the real environment, and therefore in this research the edges are used to compare the candidates with the sensed environment using the modified Hausdorff distance (MHD), which is one of the popular object matching methods in the image processing field [6]. All candidates have the shapes transformed into the image plane and let $A=\{a_1,\ldots, a_{Na}\}$ denotes the edge points projected on the image. The dotted line in Fig. 11 is one example. $B=\{b_1,\ldots, b_{Nb}\}$ denotes the edge points extracted from the camera image. Only points within the similarity evaluation area are considered, and then the MHD between two edge sets *A* and *B*, $h_{\text{mod}}(A,B)$, is calculated by

$$h_{\text{mod}}(A,B) = \frac{1}{N_a}\sum_{a\in A}\min_{b\in B}\|a-b\| \tag{10}$$

where N_a and N_b mean the number of edge points of set *A* and *B*, respectively. When the edges are extracted near the candidate, the difference, MHD, between the candidate and extracted edge is small. Then, the reliability of a candidate, *R*, is simply defined by

$$R = 1.0 - \frac{h_{\text{mod}}(A,B)}{d_{\text{sim}}} \tag{11}$$

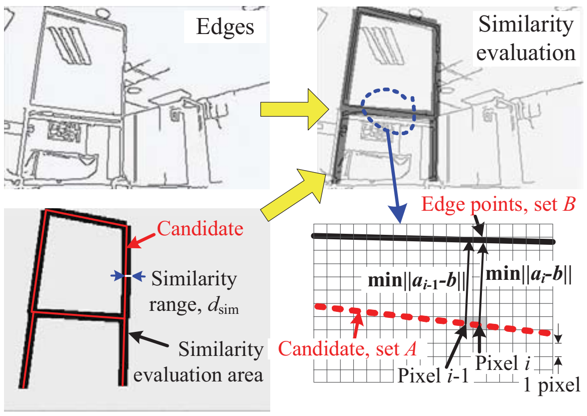

Fig. 11. Candidate evaluation based on MHD within the similarity evaluation area.

The reliability R means how similar the edges are to the candidate. If the candidate is similar to the real object, then the reliability R is close to 1. Figure 12 shows examples of several reliabilities with $d_{\mathrm{sim}} = 10$ pixels. After calculating the reliabilities of all generated candidates, the best candidate can be chosen to estimate the object pose.

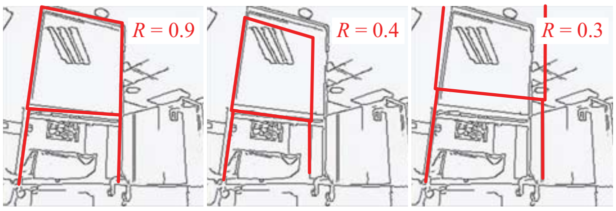

Fig. 12. Examples of candidate evaluations.

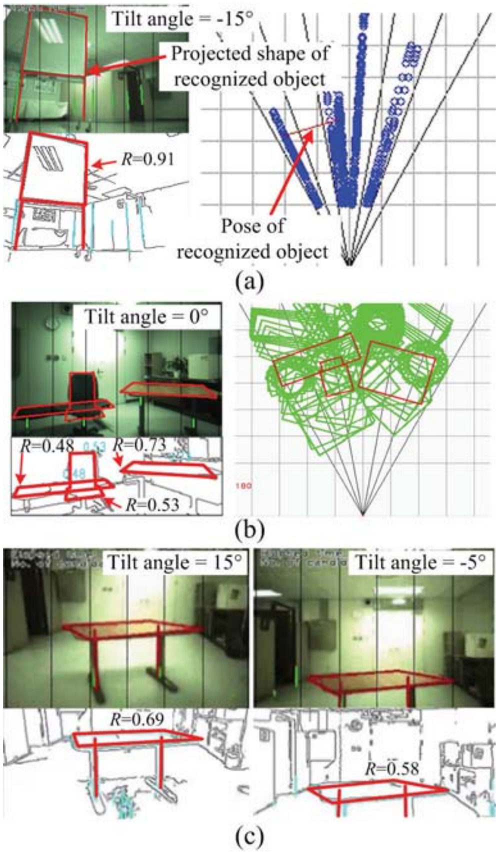

Fig. 13. Results of object detection in real environment.

By the proposed scheme, the objects on which a robot has information can be detected in the real environment. Figure 13 shows the experimental results with $d_{\mathrm{sim}} = 5$ pixels and different tilt angles in the real environment. In Fig. 13(a), the height of a camera is lower than 300mm and it sees the environment upward with a tilt angle of -15°. About 150 candidates were generated and the best one was selected with very high reliability. More than one object, three objects in this case, can be detected together from one image, as shown in Fig. 13(b). The height of a camera is higher than 1m and one table was recognized with two tilt angles in Fig. 13(c). As shown in these results, the reliability of object recognition is usually not high and object recognition is a difficult task because a lot of edge points are extracted from both objects and backgrounds. The detected object can be exploited for various purposes, such as localization, obstacle avoidance, and so on.

6. Conclusions

In this paper, the monocular camera-based scheme to detect objects with legs was proposed. First, legs are detected by the vertical line segments of objects extracted from the camera image, and then many candidates are generated based on the model information. Finally, the similarity between each candidate and the edges extracted from the image is evaluated and the best candidate is chosen as an object pose. This object detection scheme was validated by experiments in the real environment. From this research, the following conclusions have been drawn.

1. The object detection can be conducted without initialization before tracking because a robot generates and evaluates many possible candidates using the 3D model information and similarity range. Therefore, a robot can detect some objects that appear and disappear during navigation in real time.
2. The proposed scheme generates and evaluates object candidates in consideration of the tilt angle of a camera. Therefore, a robot can detect objects using the model information when the camera sees them upward or downward, and the proposed scheme can be exploited by various types of robots.

Acknowledgements

This paper was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Knowledge Economy of Korea.

References

[1] D. S. Kim, and R. Nevatia, "A method for recognition and localization of generic objects for indoor navigation," in *Proc. 2nd IEEE Workshop on Applications of Computer Vision*, Florida, 1994.

[2] H. Takeda, A. Ueno, M. Saji, T. Nakano and K. Miyamoto, "A robot recognizing everyday objects," in *Proc. 2000 IEEE/RSJ Int. Conf. on IROS*, Kagawa, 2000.

[3] M. Vincze, M. Ayromlou, W. Ponweiser, and M. Zillich, "Edge Projected Integration of Image and Model Cues for Robust Model-based Object Tracking," *Trans. on IJRR*, Vol. 20, No. 7, pp. 533-552, 2001.

[4] M. Vincze, M. Schlemmer, P. Gemeiner, and M. Ayromlou, "Vision for Robotics," *IEEE Rob. and Auto. Magazine*, Dec. 2005, pp. 2-14.

[5] T. B. Kwon, and J. B. Song, "Recognition of Objects with Legs Using Vision-based Candidate Generation and Evaluation" in *Proc. of URAI2007*, POSTECH, 2007.

[6] M. P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," *in Proc. of IAPR Int. Conf. on Pattern Recognition*, Jerusalem, 1994.