Table Recognition through Range-based Candidate Generation and Vision-based Candidate Evaluation

Tae-Bum Kwon
Dept. of Mechatronics
Korea University
Seoul, Korea
haptics@korea.ac.kr

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

Munsang Kim
Center for Intelligent Robotics
Frontier Intelligent Robotics Program,
Seoul, Korea
munsang@kist.re.kr

Abstract: For successful navigation of an indoor mobile robot, a robot must perceive the environment reliably. Although most objects are treated as obstacles to avoid from the navigation point of view, a table needs to be recognized in many applications because it can serve as a landmark or a goal position for some tasks. However, it is very difficult to detect a table featured by thin legs using only a 2D range sensor because this type of sensor can sense only the table legs. In this case, visual information can be used to improve perception, but a vision sensor often provides relatively less accurate and less stable than a range sensor. In this research, a simple but practical method to detect a table is proposed using both vision and range sensors. In this scheme, a robot detects table legs using a range sensor, generates many possible candidates for table pose (position and orientation), and selects the best candidate by evaluating all candidates using visual information. The detected table can then be mapped and exploited during navigation. Various experiments in the real environment showed the proposed scheme was effective in table detection.

1. INTRODUCTION

In order to navigate in the real environment, lots of navigation techniques are required, such as mapping, localization, path planning, obstacle avoidance, and so on. Almost all techniques are based on the robot's perception. For example, a robot maps the perceived environment, estimates its pose using the sensed environment, and plans the path which detours around the recognized obstacle. Therefore, perception is the most basic and important ability.

Three factors can influence the performance of perception; sensing ability, type of obstacle, and perception scheme. The sensing ability is limited by the sensor type. For example, a laser scanner offers very accurate and long range data, but an IR scanner gives less accurate and shorter range data than a laser scanner. The type of obstacle and the sensing ability are also closely related. For example, a range scanner can sense 2D obstacles, whereas a stereo vision sensor can sense 3D obstacles. These two factors, sensing ability and obstacle types, are related to the hardware and environment and not easy to deal with. Therefore, the perception scheme, or algorithm is important and should be improved to overcome the limitations of the sensing ability and types of obstacles.

A table is one of the important objects which are encountered in indoor navigation, as shown in Fig. 1. A robot should avoid it or stop in front of it to interact with objects on it. To perceive a table in the real environment, however, is a very hard task because a range sensor usually detects only the table legs and a vision sensor observes not only a table surface but also other objects and background together. Therefore, a mobile robot can easily collide with a table if the robot cannot detect it well.

Fig.1 Hardly detectable table in real environment.

Several ways of detecting a table have been studied so far. The method to detect a door and a table using line features of images were proposed [1]. In this research, the lines extracted from image were classified into a vertical line and a horizontal line, and compared with the models which were made by an operator. But the objects should be in a standard pose and very close to the wall because many lines between the object and the background could be extracted if the objects were not close to the wall.

Another object detection method which uses the information about the color and 3D shape of the objects in its database was proposed 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 vision sensor. This method was not practical because of two drawbacks. First, it was assumed that the poses of both the robot and the objects were known and the image viewed at the robot pose could be generated using object information. Therefore, it could detect only the table fixed at the known position. Second, it compared all corresponding pixels by color between the images obtained by a vision sensor and generated using object models. Thus, it was inefficient

* Corresponding Author

and could not deal with the uncertainty of the sensor.

The recognition system to detect furniture and categorize it was proposed in [3]. In this method, polygons and rectangular surfaces were extracted from the image and used as a feature. Various types of objects such as tables, chairs, beds, and so on were analyzed and some generalized models were made. Then, features were extracted from the image and grouped by the shape and orientation in 3D space to categorize the detected objects. This method can offer a category representation scheme which is analytical and systematic, but suffers from the serious drawback that the features should be extracted very accurately. Therefore, only line drawings were used and no experimental results were shown with the real image obtained by a vision sensor in the real environment.

This paper is focused on developing the simple and practically useful scheme to detect a table. This method uses both range and vision sensors to use both sensors' advantages. By this approach, a robot can detect a table while it moves in the real environment, and the detected table can be mapped and avoided.

The remainder of this paper is organized as follows. Section 2 presents the overall structure of the proposed scheme. Section 3 describes how to detect the table legs using a range sensor and how to generate the candidates for the table pose. Section 4 describes how to choose the best candidate among several candidates using the vision sensor data. Section 5 represents the experimental results of table detection and avoidance. Finally, section 6 presents conclusions.

2. OVERALL STRUCTURE

In this approach, the range and vision sensors are used for reliable table detection. A range sensor such as a laser scanner or an IR scanner detects the 2D environment, and in this case, only the table legs can be sensed. Even though a range sensor can offer only 2D information, it can provide more accurate and stable data than a vision sensor. Therefore, good estimation of leg positions can be obtained using range sensor data.

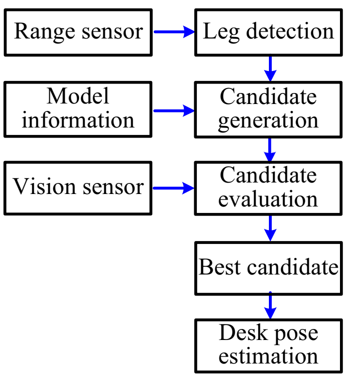

Fig.2 Overall structure of proposed scheme.

After leg detection, the candidates for table pose should be generated. In this case, the model information is required to generate the candidates which are similar to the real table. Without the model, many candidates should be generated and evaluated to find the real table pose, but with the aid of the model information, the accurate and small candidates can be made.

All candidates should be evaluated to choose the best one. The surface of a table is the most important part that has relatively plenty of features used to compare the candidates with the real table. A vision sensor is suitable for performing this task. After calculating the similarity between each candidate and the vision sensor data, the best candidate is chosen and the table pose is estimated. Figure 2 shows the overall structure of the scheme.

3. LEG DETECTION AND CANDIDATE GENERATION USING RANGE SENSOR

3.1 Leg detection with range sensor data

A leg of a common table can be modeled as a pole. To detect a pole reliably using a range sensor is a very difficult task, especially in the case that a robot uses an IR scanner which is far less accurate than a laser scanner. In this research, however, it is assumed that a laser scanner is used and the pole is reliably detected. One of the popular methods to detect a pole is the Hough transform that is widely used in mage processing. Originally, the Hough transform solves the clustering problem automatically, but in this research the target to detect is a pole which is apart from other objects and isolated. In this case, a simple clustering method can be used, which makes the feature extraction fast.

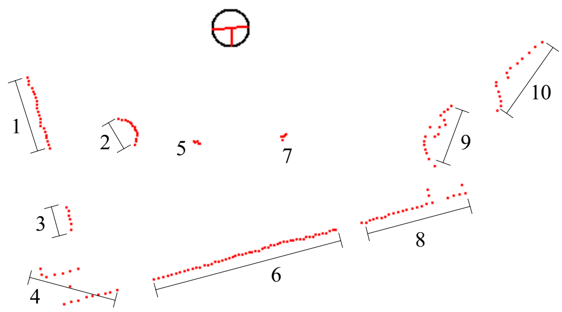

Fig. 3 Range sensor data in real environment and clustered groups. Each number is a group number.

Figure 3 shows the range sensor data in the real environment including a table. The sensed points for the table legs, group 5 and 7 in Fig. 3, are apart from the other sensed points. The environment is sensed with a resolution of 1° from -90° to 90° by the SICK laser scanner. 181 sensed points are expressed as

$$R_t = \{r_{i,t}\} \ (i = 1,\dots,181) \tag{1}$$

where $r_{i,t}$ is the range reading of the ith sensing point at time t. The clustering criterion using a Euclidian distance is defined as

$$\left| r_{i,t} - r_{i+1,t} \right| > d \tag{2}$$

where d is the threshold above which two objects are considered apart from each other. When the inequality (2) is satisfied, a new group starting from the $(i+1)$th sensing point is created.

After clustering, the Hough transform for detection of circular arcs can be applied [4]. This method gives the radius and center position of the circle. Figure 4 shows the result of detection. Using this method, a robot can detect the circular poles such as table legs, but is not able to distinguish the table legs from other detected arcs. As shown in Fig. 4, not only the table legs but also human legs or cylindrical trash cans be detected. The numbers in Fig. 4 represent the distances (in mm) from the robot to the center of the detected poles.

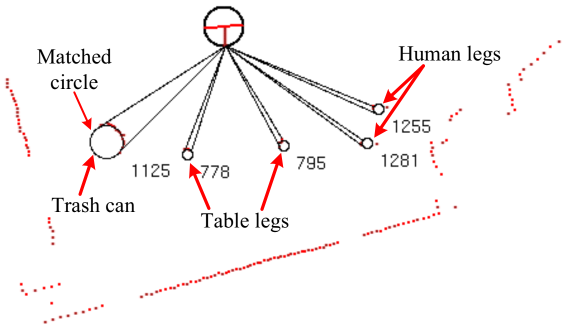

Fig. 4 Detection of arcs based on Hough transform.

3.2 Table model information

If a robot has no information on the table, the whole area in which a table can be located should be searched. In this case, it is difficult for a robot to detect a table in real-time due to high computational complexity. To overcome this difficulty, the proposed method uses the minimum information on the table; the lengths of two axes for a rectangular table and the number of legs. Figure 5 shows examples of the model information which was used in the experiments.

Another advantage of the use of the model information is that a robot can detect a table although the whole table is not within the field of view (FOV) of a vision sensor. The FOV of a common camera is rather small, for example 58° for the camera used in this experiment, which corresponds to one third of 180° of a range scanner. If the robot has no model information, the detection ability is restricted to the case when the entire table is placed within the FOV of the camera. Therefore, the model information is useful for reliable and practical table detection.

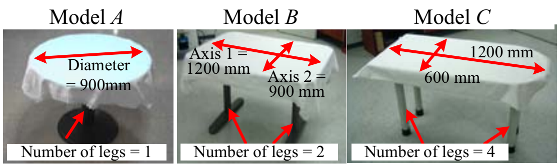

Fig. 5 Examples of model information.

3.3 Candidate generation with detected legs

The candidates can be generated using both the model information and some poles selected among all detected poles. This task is executed by the candidate generator of each model. Model A with only one leg in Fig. 5 has a very simple candidate generator. As shown in Fig. 6, it chooses one pole among all detected poles whose center positions are already found by the Hough transform. Then, it generates a candidate for every chosen pole. It is possible that there exist other poles within the generated candidate because human legs or other objects can be detected as a pole. The poses of all candidates written in the robot coordinates are stored in memory.

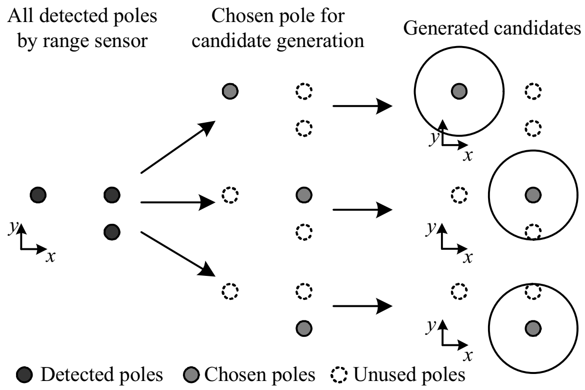

Fig. 6 Example of candidate generation of model A.

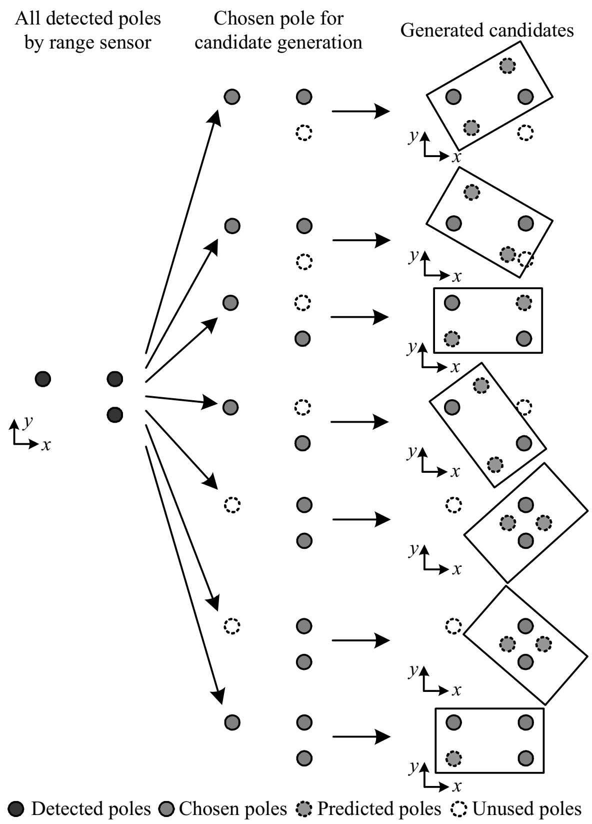

Fig. 7 Example of candidate generation of model C.

Table 1. Examples of candidate generation rules of some models.

	Model *A* No. of legs: 1	Model *B* No. of legs: 2	Model *D* No. of legs: 3	Model *C* No. of legs: 4
No. of considered poles: 1		Too many possible candidates Meaningless	Too many possible candidates Meaningless	Too many possible candidates Meaningless
No. of considered poles: 2	Already considered above			
No. of considered poles: 3	Already considered above	Already considered above		
No. of considered poles: 4	Already considered above	Already considered above	Already considered above	

Detected poles Predicted poles

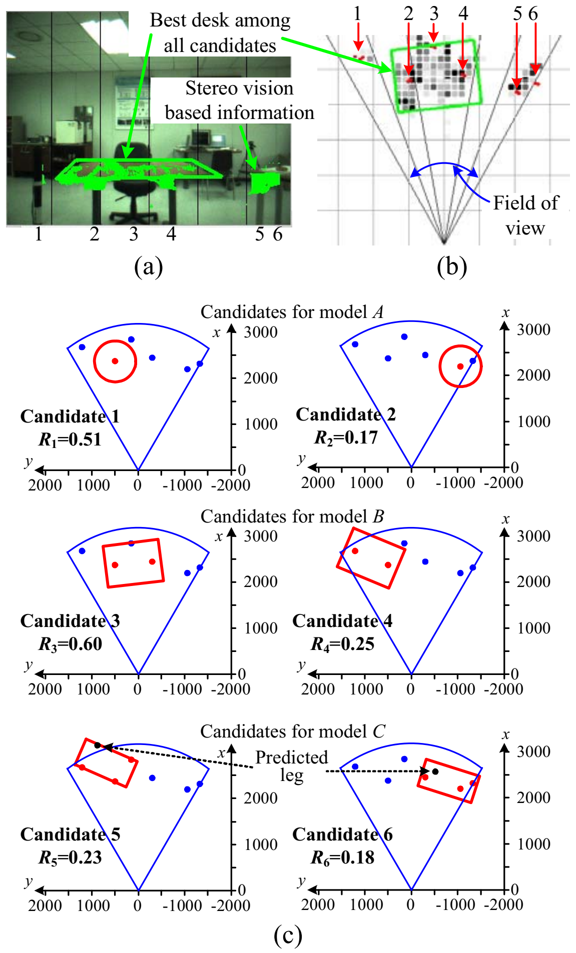

Fig. 8 Examples of some candidates in real environment using model *A*, *B* and *C*; (a) visual image and some stereo vision based information, (b) positions of 6 detected poles, and (c) some of candidates generated using model information and positions of poles.

Figure 7 shows an example of candidate generation of model *C* in Fig. 5. Since the model information has only the number of legs and the table size, many combinations of two or more poles can be used to estimate the pose of model *C*. In Fig. 7, seven candidates are possible using 3 detected poles. If more poles are detected, far more candidates will be generated. For example, when 4 poles are detected, not only all types of candidates as shown in Fig. 7, but also candidates in consideration of 4 poles in Table 1 must be generated. The poses of all candidates relative to the robot coordinate frame are easily calculated using geometry since the positions of the detected poles and the model shapes are known.

Table 1 shows the candidate generation rules of some models. It is easily obtained using geometry. The detailed generation rules of model *A* and *C* were described above. If a robot wants to detect a new table such as model *D* which is a circular table with 3 legs, only candidate generation rules for the new model are required. All poses generated by these rules expressed in the robot coordinate frame are stored in memory and their reliability will be calculated in the evaluation stage. In a certain case, too many candidates may be generated. For example, when only one detected pole is used for candidate generation of model *B* which has two legs, the number of candidates is almost infinite in consideration of the orientation. In this case, the candidate generation is meaningless.

Figure 8 shows some candidates generated in the real environment. Model *A*, *B*, and *C* in table 1 are used and there are 6 poles which can be detected as a table leg in Fig. 8(a). The positions of the detected poles are shown in Fig. 8(b). From these poles, many candidates can be generated, and Fig. 8(c) shows some of them. In the figure, *R* means the reliability of each candidate, which will be described in section 4. In this case, candidate 3 is chosen as the table pose because it has the highest reliability.

4. VISION-BASED CANDIDATE EVALUATION

Every candidate should be evaluated to determine how similar it is to the real table. In this research, the stereo vision based information is used to compare the candidates with the sensed environment because both the candidate poses and the stereo depth are 3D information. The simple and intuitive approach to calculating the reliability of the candidate is to compare the sizes of two areas; the size of the candidate and the area where the stereo vision information can be obtained on the inside of that candidate region.

First, the environment is divided into a discrete 2D grid of cells with uniform size, 100mm by 100mm in this research. The initial values of all cells are *empty*. Then the stereo vision based information between 400mm and 800mm in height from the ground is projected on the grid. The cells on which the stereo vision based information is projected change their values to '*occupied*.' This process is done repeatedly at every evaluation stage. In Fig. 9, the gray cells represent the occupied cells. A darker cell has more 3D points within its boundary than a lighter cell. In this research, however, only the number of occupied cells is important (not the gray scale of the cell).

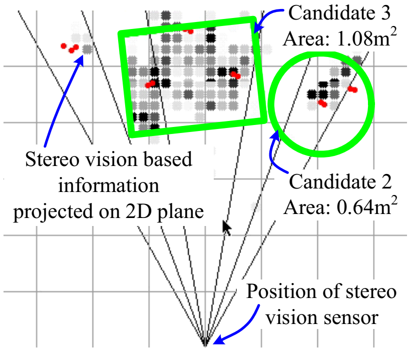

Fig. 9 Evaluation using candidate size and stereo vision based information.

These cells are used to evaluate the reliability of the candidates. If many cells within the area of one candidate have the occupied values, the reliability of that candidate is high as shown in candidate 3 in Fig. 9. The area of this candidate is 1.08m^2, which corresponds to about 108 cells. The number of the occupied cells is 65. In this research, it is assumed that the candidate is reliable if more than half of all cells of the candidate are occupied. Therefore, it is considered that the table pose is similar to this candidate. On the other hand, the area of candidate 2 is 0.64m^2 and it consists of about 64 cells. However, the number of the occupied cells within the area of this candidate is only 11, which means the reliability of this candidate is low. The reliability of a candidate is defined as

$$R_i = \frac{N_i}{A_i / 0.01} \quad (i = 1, ..., n) \tag{3}$$

where n is the total number of all candidates, R_i is the reliability of the ith candidate, N_i is the number of the cells within the area of the ith candidate and A_i is the area of the ith candidate. Note that A_i is divided by 0.01 since the area of one cell is 0.01m^2. After calculating the reliabilities of all candidates, the best candidate can be chosen to estimate the pose of a table. In Fig. 8, R shows the reliability of each candidate calculated by equation (3). Because the highest value is R_3 = 0.60, candidate 3 is chosen as the pose of the detected table.

5. EXPERIMENTS IN REAL ENVIRONMENT

By the proposed scheme, the tables on which a robot has information can be detected in the real environment. Figure 10 shows the experimental results. Only candidates for model B are drawn in Fig. 10 because the image would be very unclear if all candidates were drawn together. In Fig. 10(a), only two legs are detected and one candidate for model B is generated and its reliability is computed as 0.50. This candidate is chosen as the pose of a table and is projected on the right image of Fig. 10(a). In Fig. 10(b), one additional pole is detected and two candidates are generated. By candidate evaluation, the same candidate is chosen as the best one. In Fig. 10(c) and (d), many candidates are generated because a chair is added and human legs are also detected. From (a) to (d), however, one candidate whose pose is similar to the pose of a real table is chosen. The value of the reliability varies little because the stereo vision based information continues to change little.

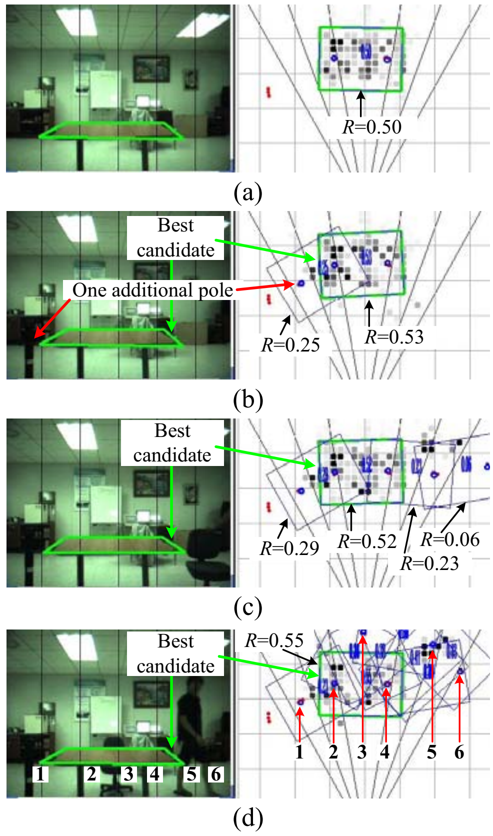

Fig. 10 Evaluation using candidate size and stereo vision based information.

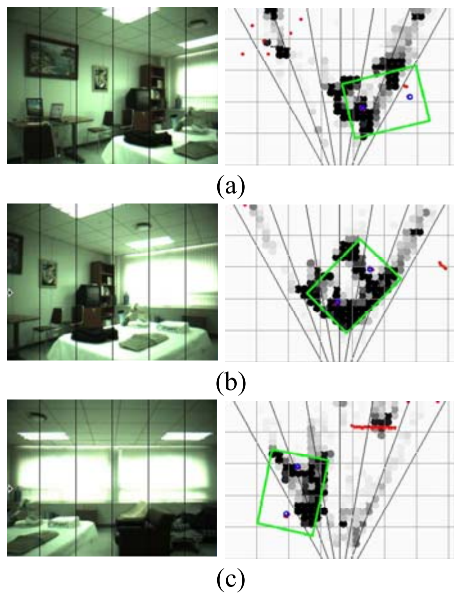

Fig. 11 Detection of a table which is not fully seen.

In Fig. 11, a robot turns clockwise. If a robot were able to detect a table only in the case when the whole table surface was placed within the FOV, as shown in Fig. 11(b), the table detection ability could be limited and might be practically useless. However, the candidate pose can be estimated with the aid of both a range sensor and the model information even though a table is not fully seen within the FOV of a vision sensor in Fig. 11(a) and (c). In these cases, one third of the candidate shape is located on the outside of the FOV, but the table pose can be estimated because the reliability of the candidate is higher than the threshold.

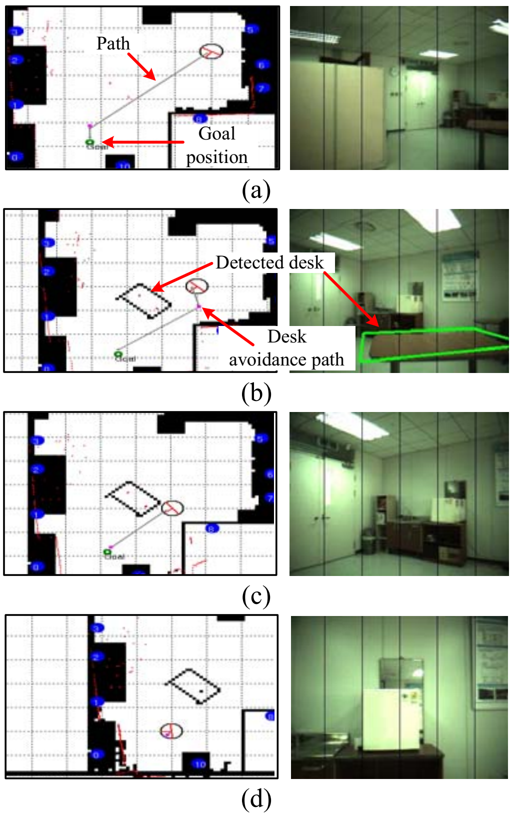

Fig. 12 Experiments on table detection and avoidance in real environment. Left figures show map and path, and right figures show images obtained from vision sensor.

The detected table can be exploited for the various purposes, such as map building, localization, obstacle avoidance, and so on. Figure 12 is an example of table avoidance. In Fig. 12(a), the initial path to the refrigerator is planned. During movement to the refrigerator, the robot detects a table using the proposed method and maps it on the grid map in Fig. 12(b). The avoidance path is generated by the gradient method [5], one of the popular path planning methods, and then the robot avoids it as shown in Fig. 12(c) and safely arrives at the goal position.

6. CONCLUSIONS

In this paper, a simple but practically useful scheme to detect a table was proposed. The possible table legs are detected by a range sensor, and many candidates are generated. Then, the similarity between the model and each candidate is evaluated and the best candidate is chosen as a table pose. This table detection scheme was validated by a series of experiments. From this research, the following conclusions have been drawn.

1. In the real environment, many objects such as table legs, human legs, chair legs, and other poles can be recognized as a table leg by a range sensor. Therefore, many candidates for the table pose should be generated with all detected poles.
2. To choose the best one among many candidates, a vision sensor can be used. The comparison between the stereo vision based information and each candidate for the table pose in 3D space can serve as a good evaluation tool.
3. The accurate candidates for table pose can be generated using the 3D model information that is given in the database even if a small part of the entire table is projected on the outside of the image due to the small FOV (Field Of View) of the vision sensor.

Because the stereo vision information is sensitive to the noise and illumination condition, the table detection method which uses the only vision information may not be enough in the real environment. Currently, the research on table detection using not only the stereo vision based information but also other features is under way.

ACKNOWLEDGEMENT

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

REFERENCES

[1] D. S. Kim and R. Nevatia, "A Method for Recognition and Localization of Generic Objects for Indoor Navigation," *Proceeding of the Second IEEE Workshop on Applications of Computer Vision*, Vol. 2, pp. 1069-1076, 1994.
[2] H. Takeda, A. Ueno, M. Saji, T. Nakano and K. Miyamoto, "A Robot Recognizing Everyday Objects," *Proceeding of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2, pp. 1107-1112, 2000.
[3] C. Rasche, *The Making of a Neuromorphic Visual System*, Springer, 2005.
[4] P. Kierkegaard, "A Method for Detection of Circular Arcs Based on the Hough Transform," *Trans. on Machine Vision and Applications*, Vol. 5, No. 4, pp 249-263, 1992.
[5] K. Konolige, "A Gradient Method for Realtime Robot Control," *Proceeding of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 639-646, 2000.