1

Intelligent Unmanned Store Service Robot "Part Timer"

Ho Seok Ahn*, In-Kyu Sa**, Young Min Baek* and Jin Young Choi*
Seoul National University, Samsung Electronics Co.***
Republic of Korea

1. Introduction

In the 21st century, life for humans being has become busy to the extent that they strive for a comfortable and easy life. Service robots can provide this comfort to humans by doing all their difficult and dirty work. In order to provide this service, robots should be able to work in the same environment as humans. Service robots should have intelligent abilities such as greeting, conversation with humans, moving while avoiding obstacles, grabbing objects, etc (Sakai K. et al., 2005). However, service robots can just walk or run slowly (Riezenman, M.J., 2002), recognize little behavior (Waldherr S., 1998), and work in simple or limited environment until now. Therefore it is essential for service robots to have communication and recognition abilities that can be used in any kind of situation to help humans in a real environment. Lots of studies have focused on these abilities of service robots and many system architectures for intelligent service robots are introduced.
Mumolo has proposed the algorithm based on Prosodic model that processes vocal interaction using natural language (Mumolo E. et al., 2001). Kleinehagenbrock has introduced the system architecture of intelligent robots based on agent for humans robot interaction and have applied it to real robot (Kleinehagenbrock M. et al., 2004). Hyung-Min Koo has proposed new software architecture that changes its functions and compositions according to situation (Hyung-Min Koo et al., 2005). Kanda has proposed the system architecture that performs more than 100 behaviors with simple voice (T. Kanda et al., 2002). Mikio Nakano has proposed the two-layer model for behavior and dialogue planning in conversational service robots (Nakano, M. et al., 2005). Gluer has proposed the system architecture for context based exception handling in the case of an unexpected situation (Gluer D., 2000).
The expectation of the service robot market is getting bigger and lots of researchers are interested in developing intelligent service robots recently (Sakai K. et al., 2005). ApriAlpha is home automation robots for intelligent home and used for communication with humans (Yoshimi T. et al., 2004). Besides there are lots of examples of home automation robots; ISSAC (Dong To Nguyen et al., 2005), ETRO (Jeonghye Han et al., 2005), SHR100 (Moonzoo Kim et al., 2005), MARY (Taipalus, T. Et al., 2005), PBMoRo (Ho Seok Ahn et al., 2006), PaPeRo (Sato M. Et al., 2006). The most popular robots in the market are cleaning robots. The examples are home cleaning robots such as Roomba (Jones, J.L., 2006), Roboking (Sewan Kim, 2004), office cleaning robots such as DAVID (Prassler E. et al., 1997), glass-wall

cleaning robots such as Cleaner 3 (Houxiang Zhang et al., 2006). Besides there are various kinds of service robots; welfare robots such as ROMAN (Hanebeck U.D. et al., 1997), guidance robots (Koide Y. et al., 2004), entertainment robots such as AIBO (Fujita M., 2004), therapy robots such as PARO (Shibata T., 2004).

We have designed modular system architecture for intelligent service robot and developed intelligent unmanned store service robot 'Part Timer'. It works in store environment instead of humans and manages the store without any humans' intervention. It recognizes humans and makes conversation with customers who order some products. It can move autonomously, grab some products, and deliver them to customers. It finds suitable information from the internet and gives search results to humans using voice synthesis. Administrator can monitor the status of the store anywhere and anytime by connecting Part Timer or the store server which has most of the information of the store. As Part Timer has lots of abilities like this, it is required that system architecture need to be easy to add or remove functional engine modules. This system architecture is useful for service robots which have complex functions.

2. Part timer

2.1 Overall description

Part Timer is an intelligent service robot specified for unmanned store developed by Perception and Intelligent Lab.(PIL) at Seoul National University and Software Membership at Samsung Electronics Co. It is 100cm tall and weighs 30kgs. The frame of the robot is composed of acryl pannel and aluminum poles for light weight. A lithium-polymer battery is used and it supplies for about three hours without external power source. One embedded computer which uses Intel pentium 4 processor is located inside the robot and Microsoft Embedded Windows XP is used for operating system. Main system accesses the internet using wireless communication and controls most of the appliances in the store using Bluetooth and wireless LAN. It has two wheels for moving and uses ten ultrasonic sensors for measuring distance to objects. It has one camera for image processing and one microphone for communicating with humans by recognizing voice. It has a manipulator for grasping some objects. The manipulator has five degrees of freedon (DOF) and located at the front of the robot. All data is stored to the store server. Table 1 shows the overall specifications of Part Timer.

Height	100cm
Weight	30kg
Frame	Acryl pannel & aluminum poles
Power	Lithium-polymer
Operating System	Microsoft Embedded Windows XP
Communication	Wireless LAN, Bouetooth
Camera	Logitech Quickcam pro 4000 x 2
Moving speed	0 ~ 2m/s
Manipulator	DC motor x 6
Sensor	Ultrasonic x 11 (10 for SLAM, 1 for Manipulator) Gas sensor x 1

Table 1. The overall specifications of Part Timer

It recognizes humans by face recognition engine. It can read characters written in Korean, English and Japanese. It can recognize some objects and grab them. It communicates with humans by recognizing voice using STT(Speech To Text), TTS(Text To Speech) and reacts using conversation engine. When users ask something, it finds the suitable information by surfing the internet autonomously such as weather, news. It also plays music and video through the internet. It has navigation capability and moves to its destination by avoiding obstacles and mapping automatically. It measures the distance between robot and obstacle using ten ultrasonic sensors and calculates the trajectory and robot velocity. Using these capabilities, it receives orders, sells and delivers products. It controls several the electrical appliances, blind curtains, windows etc.

It is controlled remotely as well as directly. Humans manages Part Timer by PDA and mobile phone. Humans can monitor the status of the store and use every functinos which are provided as local service. For example, humans can open the door and close the window in the store remotely. It sends the streaming camera image from the robot to PDA and store server. If an intruder comes in the store and destorys the robot, it is possible to arrest him using store server information, because every status of the store and streaming image are stored in the store server. Fig. 1 shows the functions of Part Timer according to location. Fig 2 shows the environment of the unmanned store service robot Part Timer.

Fig. 3 shows some examples of using Part Timer. In the case of the administration of the store, humans makes schedule of Part Timer remotely as well as directly. It is possible to send some commands to Part Timer and to control it to move anywhere. In the case of guest, when guest meet Part Timer, he just behave same as in a normal case. Part Timer recognizes who the guest is, when the guest visits again, which products guest bought, etc. Part Timer makes conversation with the guest using these data. Part Timer gets the order and sells it. If guest asks something such as the weather of today, the nearest bank form here, Part Timer searches suitable information from the internet and gives the answer to the guest.

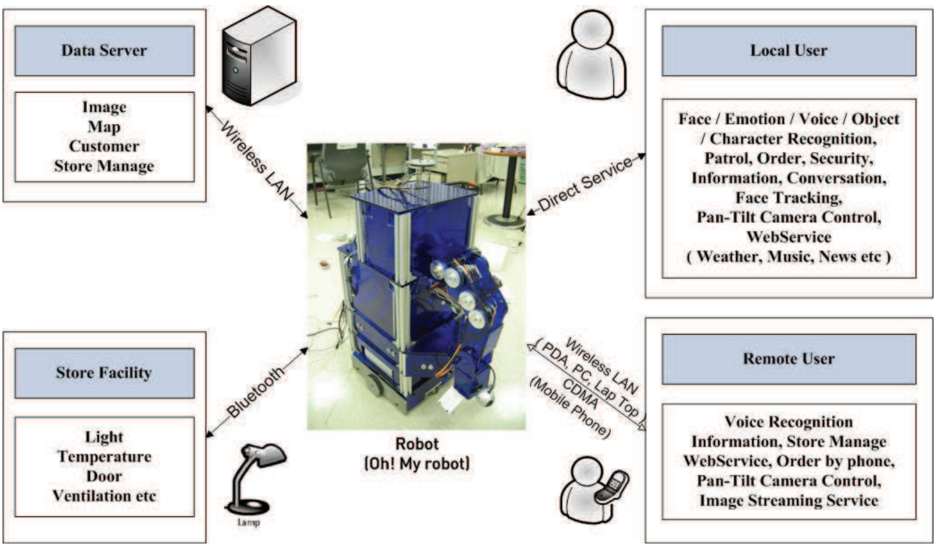

Fig. 1. The functions of Part Timer according to location.

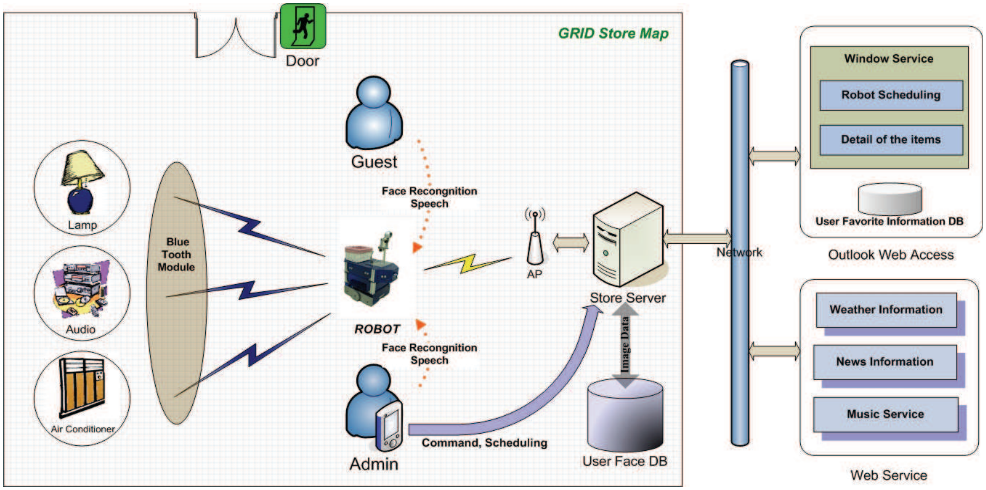

Fig. 2. The environment of the unmanned store service robot Part Timer.

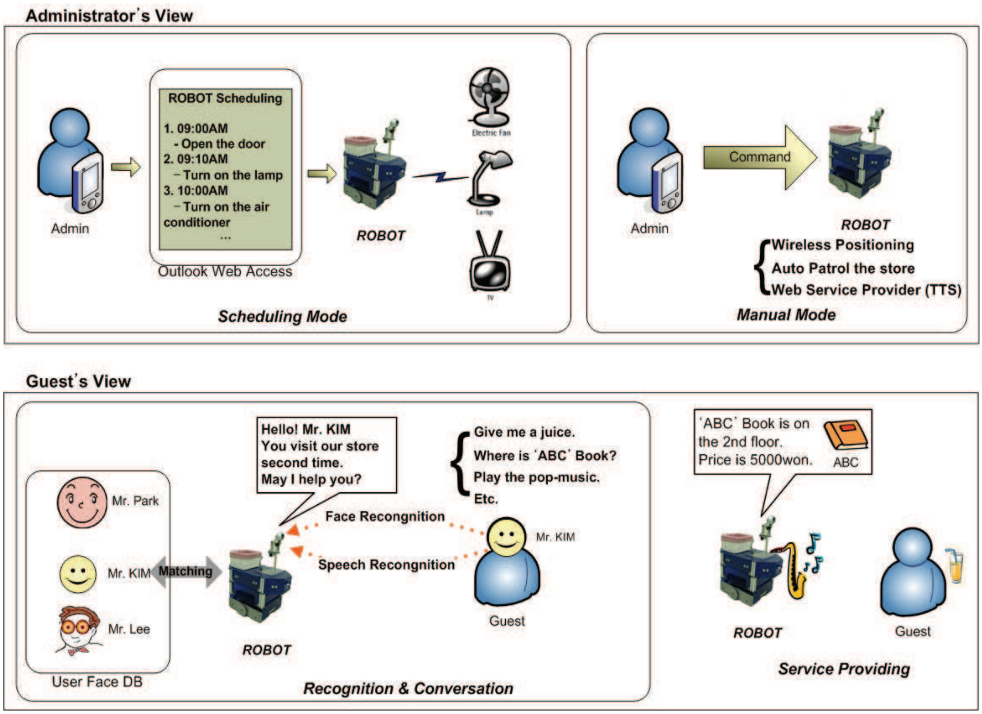

Fig. 3. Some examples of using Part Timer.

2.2 System architecture

Notion for software development is divided into structured method and object oriented method (Erich Gamma et al., 1994). Although object oriented method costs a lot for developing, it is more favourable method because of low costs of maintaining, re-using, and revising. Especially in the case of developing intelligent service robots, this method is efficient as lots of researchers work together and it is required to upgrade the robot continuously. Part Timer has lots of functions such as image processing, voice recognition and navigation which are required capabilities for intelligent service robots. Part Timer uses modular system architecture using object oriented method for connecting each functions efficiently. Since it can be different that which intelligent functional engine is needed according to the purpose of the robots. Therefore system architecture for intelligent service robot should allow addition or removal of intelligent functional engine easily.

Fig. 4 shows the system architecture of Part Timer. The system architecture consists of four layers; User Interface layer (UIL), Behavior Scheduling Layer (BSL), Intelligent Module Layer (IML), and Hardware LayerHWL). UIL is the layer for communication with humans. This layer takes part in Humans-Robot Interaction. UIL has some devices for interaction with humans such as touch screen and speaker. UIL offers various kinds of devices for efficiency such as PDA and mobile phone. User can get some information by these devices. These devices can be changed according to the purpose of the robots. Part Timer includes various behaviors such as conversation, delivering and gesture in UIL.

BSL is the core layer that connects each intelligent functional engine organically and decides how to act. Connections are different according to robots and their purpose. Main system in BSL schedules whole works of robot. IML is the layer for each intelligent functional engine. HWL is the layer that contains hardware for each IML. IML and HWL consists system module of each function. As each intelligent algorithm requires some equipments for performing functions, suitable equipments should be added when some algorithms are added. Hence, these two layers organize system modules and operate together.

The system architecture of Part Timer is reconfigurable architecture connecting each intelligent functional engine efficiently. If each intelligent functional engine is regulated to specific rule, it is easy to add or remove it and saves cost for developing and maintaining. As each engine is able to choose suitable algorithms according to situation, it can have better processing results. For these reasons, we have designed Evolvable and Reconfigurable Intelligent (ERI) architecture (Jin Hee Na et al., 2005). We have defined intelligent functional engine as Intelligent Macro Core (IMC) and some IMCs are merged to one module performing one function. ERI architecture has hierarchical structure; each module has upper IMCs and lower IMCs. Upper IMC performs the job using the results from each lower IMC which performs each function. Connection manager chooses and connects suitable module according to situation.

Fig. 6 shows the functional architecture of Part Timer using ERI architecture. Behavior manager is in BSL and performs some jobs; task planning, behavior scheduling, communication, and bridge of modules. Behavior manager has lots of modules which is intelligent functional engine in IML. Part Timer has six modules; SLAM module, vision module, arm module, voice module, web service module, and facility control module. Each module is developed based on ERI architecture. Fig. 7 shows the structure of each module using ERI architecture. Each module has some IMCs which have intelligent functional algorithms.

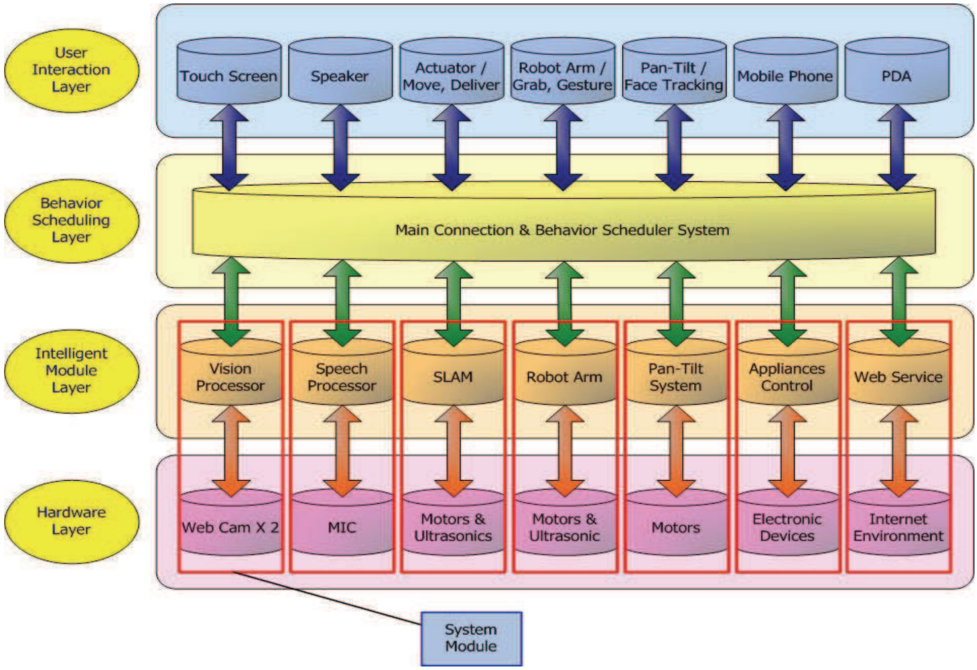

Fig. 4. The system architecture of Part Timer.

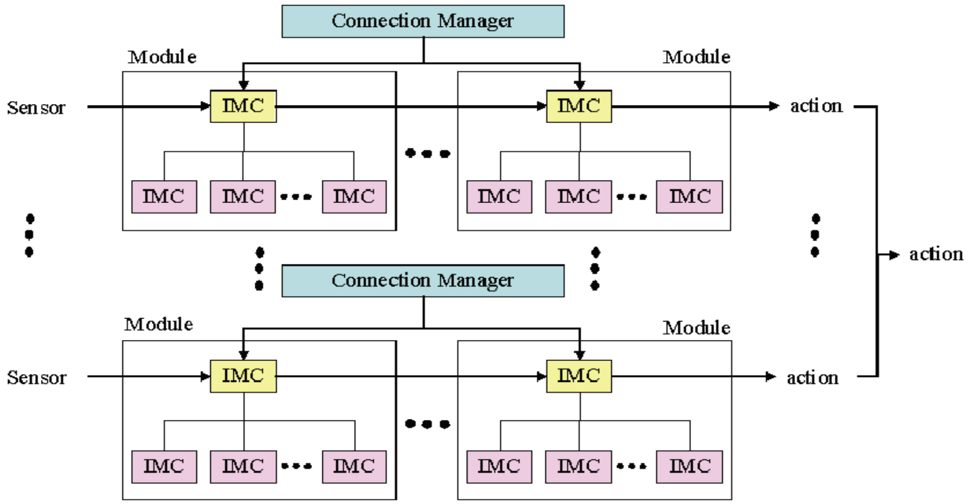

Fig. 5. The system architecture of Part Timer.

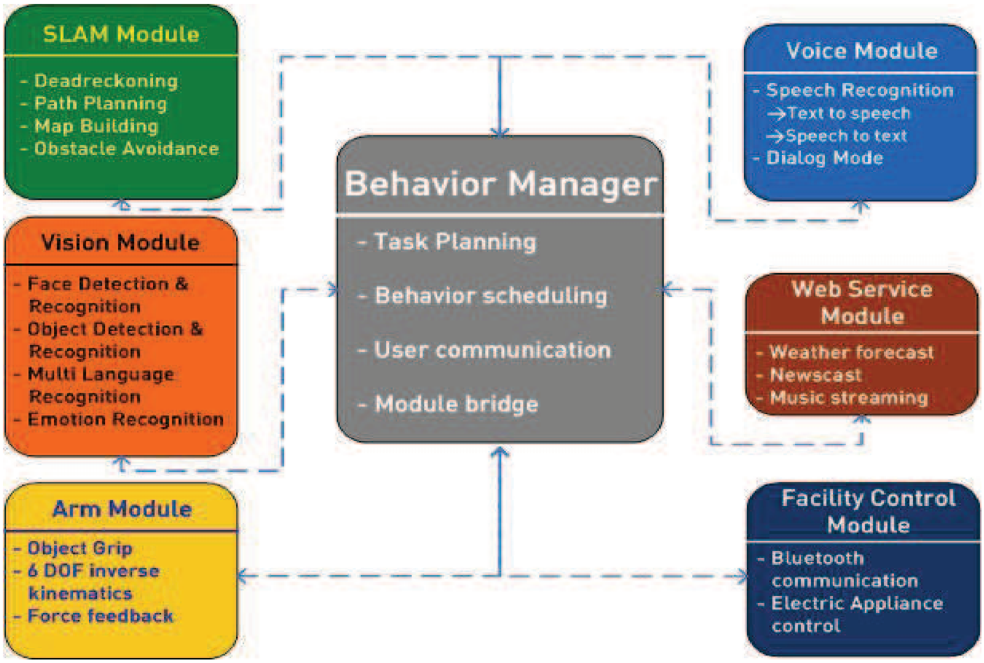

Fig. 6. The functional architecture of Part Timer using ERI architecture.

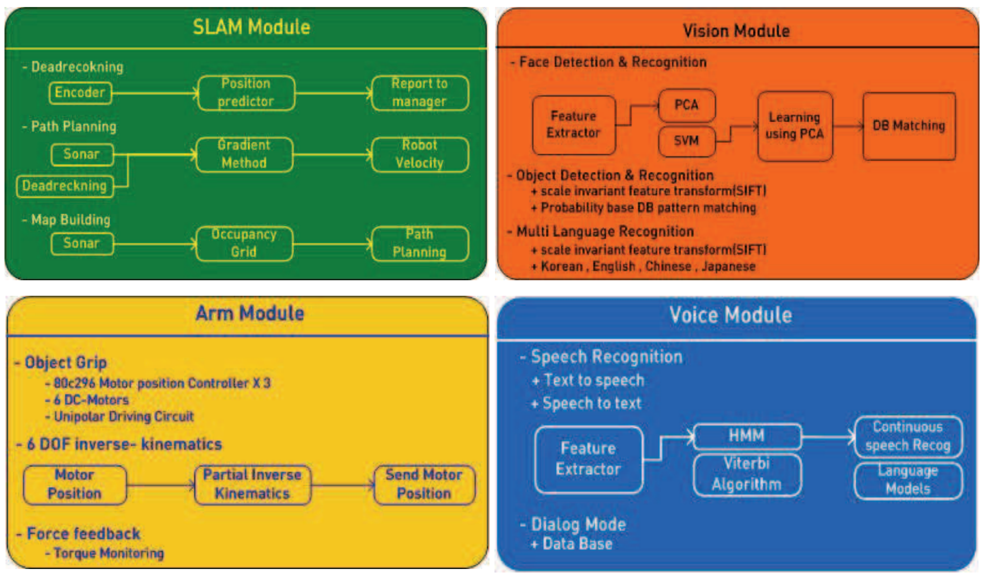

Fig. 7. The structure of each module using ERI architecture.

2.3 Mechanical design

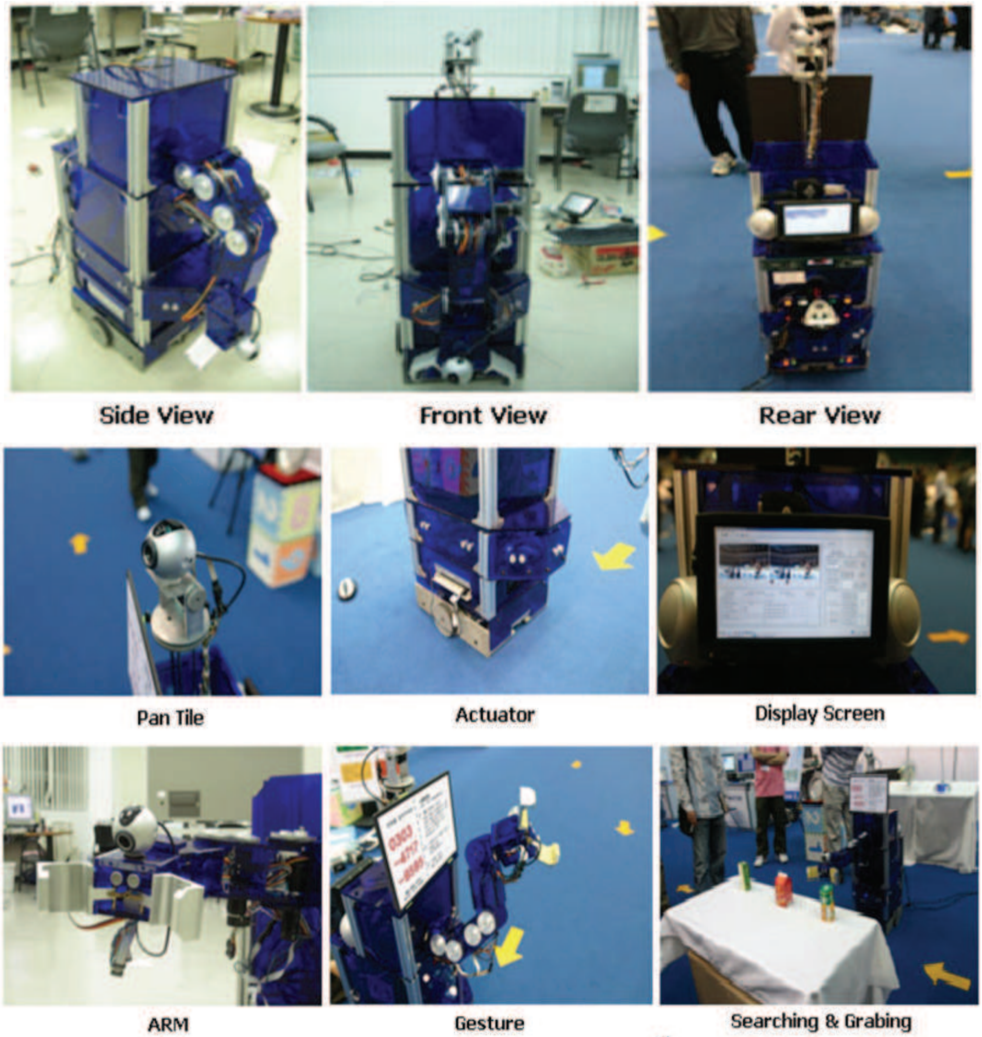

Fig. 8. The mechanical design of Part Timer.

3. Navigation

Navigation of Mobile Robots is a broad topic, covering a large spectrum of different technologies and applications. Briefly, navigation is designated moving technique from current pose to desired pose. It is remarkably easy to handle for humans, but not for robot. It is difficult that a robot recognizes its surroundings to get a current pose and to avoid an obstacles. So this book covers kinematics, motion generator, path planning and map building on mobile robot.

3.1 Kinematics on mobile robot (Seung-Min Baek, 2001).

V_R and V_L denote the velocity of wheels. Hence the linear velocity is directly proportional to the radius of a wheel. We can obtain the lineal velocity (V_C) and the angular velocity (ω_C) of the robot from the velocity of wheels. Fig. 9 shows kinematics on mobile robot.

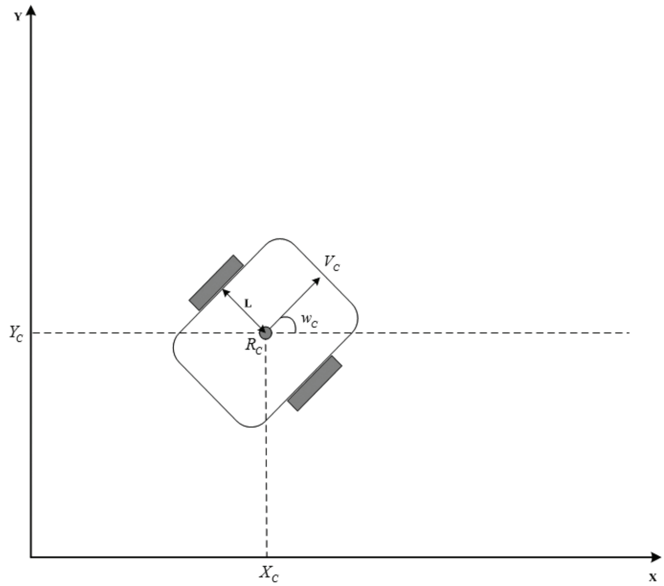

Fig. 9. Kinematics on mobile robot.

The mobile robot must recognize its pose to move freely in the room. It is required to generate proper velocity of robot for reaching the desired pose. It is called ‘motion-generator’. We propose a method which generates trapezoidal velocity profile from reactive trace generator. As shown in Fig. 10, the pose of the robot is selected randomly and the desired pose is fixed.

State vector of robot ($S_r(t)$) and Goal Vector ($G(t)$) are described as follows:

$$\begin{aligned} S_r(t) &= [x_r(t) \quad y_r(t) \quad \theta_r(t) \quad V_r(t) \quad \omega_r(t) \quad]^T \\ G(t) &= [x_g(t) \quad y_g(t)]^T \end{aligned} \tag{1}$$

where $x_r(t)$, $y_r(t)$, $\theta_r(t)$ mean robot pose in Eq. (1). $V_r(t)$ denotes linear velocity and $\omega_r(t)$ denotes angular velocity respectively in Eq. (1). Hence motion generator makes next period $V_r(t+1)$ and $\omega_r(t+1)$ by means of current pose in Eq. (2). They can be described as following:

$$\begin{aligned} V_r(t+1) &= \Pr[\ V_r(t) + \delta_t \nabla_v (t+1) \ , \ where \ |\nabla_v (t+1)| \le |\nabla_{vMAX}| \\ \omega_r(t+1) &= \Pr[\ \omega_r(t) + \delta_t \nabla_\omega (t+1) \ , \ where \ |\nabla_\omega (t+1)| \le |\nabla_{\omega MAX}| \end{aligned} \tag{2}$$

Next step is to decide $\nabla_v(t+1)$ and $\nabla_\omega(t+1)$ reasonably by considering $S_r(t)$ and $G(t)$.

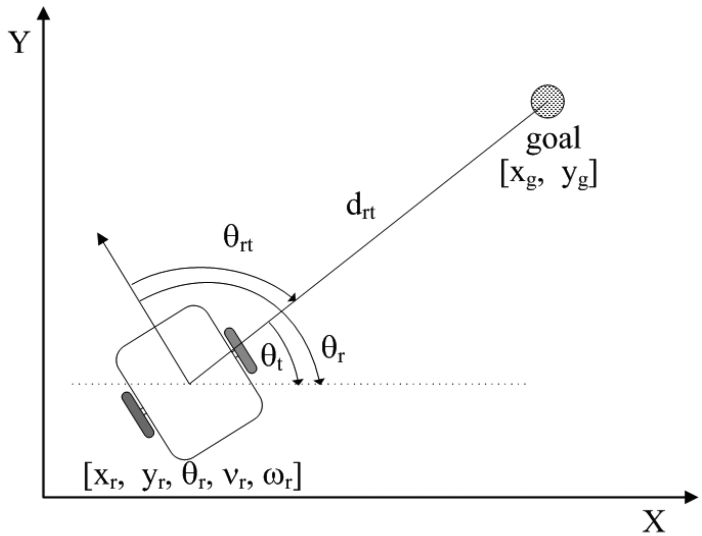

Fig. 10. The motion generator of Part Timer.

3.2 The motion generator

To determine the linear velocity of the robot, we should know the distance from robot to desired pose at first. And the velocity of the robot should be changed by acceleration section and deceleration section. θ_{rt} denotes the difference of angle between θ_r and θ_t in Fig. 10. θ_t denotes the absolute angle of the robot and target respectively. d_{rt}, the distance between the robot and the goal, can be calculated by using max velocity and max acceleration.

$$\begin{aligned} \theta_{rt} &= \theta_r - \theta_t \\ d_{dec} &= d_{end} - d_1 \\ &= \frac{V^2{}_{MAX}}{2\Delta_{VMAX}} \\ V_s &= \begin{cases} \sqrt{2 \cdot d_{rt} \cdot \Delta_{vMAX}} \ , \ if \ d_{rt} < d_{dec} \\ V_{MAX} \ , \ otherwise \end{cases} \\ V_s &= standard \ \ linear \ \ velocity \end{aligned} \tag{3}$$

$$\Delta_v(t+1) = \begin{cases} \Delta_{vMAX}, & if \ V_e(t) > \delta_t \cdot \Delta_{vMAX} \\ -\Delta_{vMAX}, & if \ V_e(t) < -\delta_t \cdot \Delta_{vMAX} \\ V_e(t) & , otherwise \end{cases} \tag{4}$$
$$where, \quad V_e(t) = V_s(t) - V_r(t)$$

$$\begin{aligned} &V_r(t+1) = \Pr[V_r(t) + \delta_t \Delta_v(t+1)] \ , only \ |\Delta_v(t+1)| \le |\Delta_{vMAX}| \\ &\Delta_v(t) = linear \ acceleration \ \ when \ \ time(t) \end{aligned} \tag{5}$$

Where $V_r(t+1)$ represents the next linear velocity in Eq. (5). We can define next angular velocity same as linear velocity.

$$\theta_{dec+} = \frac{\omega^2{}_{MAX}}{2\Delta_{\omega MAX}}$$

$$\theta_{dec-} = -\frac{\omega^2{}_{MAX}}{2\Delta_{\omega MAX}}$$

$$\omega_s = \begin{cases} -\omega_{MAX} & , \ if \ \theta_{rt} < \theta_{dec-} \\ -\sqrt{2\cdot|\theta_{rt}|\cdot\Delta_{\omega MAX}} & , \ if \ \theta_{dec-} \le \theta_{rt} < 0 \\ \sqrt{2\cdot|\theta_{rt}|\cdot\Delta_{\omega MAX}} & , \ if \ 0 \le \theta_{rt} < \theta_{dec+} \\ \omega_{MAX} & , \ otherwise \end{cases} \tag{6}$$

$$\Delta_\omega(t+1) = \begin{cases} +\Delta_{\omega MAX}, & if \ \omega_e(t) > \delta_t \cdot \Delta_{\omega MAX} \\ -\Delta_{\omega MAX}, & if \ \omega_e(t) < -\delta_t \cdot \Delta_{\omega MAX} \\ \omega_e(t) & , otherwise \end{cases}$$

$$where, \quad \omega_e(t) = \omega_s(t) - \omega_r(t)$$

$$\omega_r(t+1) = \Pr[\omega_r(t) + \delta_t \Delta_\omega(t+1)], \ only \ |\Delta_\omega(t+1)| \le |\Delta_{\omega MAX}| \\ \Delta_\omega(t) = angular \ acceleration \ when \ time(t) \tag{7}$$

It is important to decide the motion of the robot using Eq. (5) and Eq. (7). Before applying the motion generator to real robot, we tested the algorithm using simulator as shown in Fig. 11.

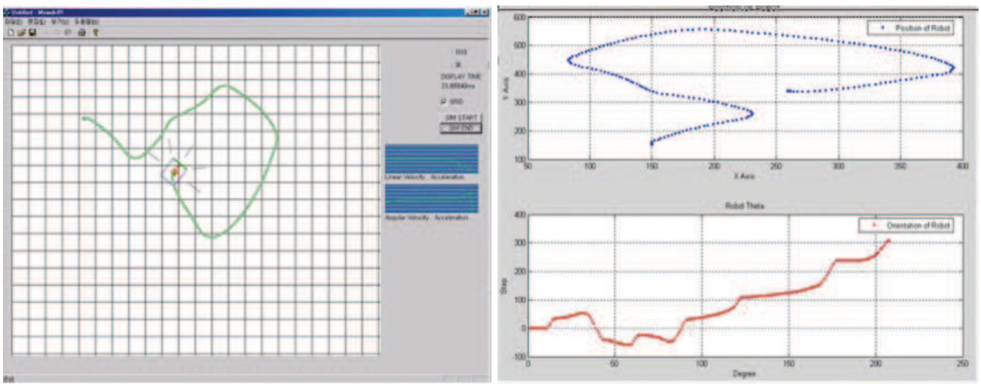

Fig. 11. The experimental results of motion generator simulation.

Right upper graph means the trajectory of the robot and right lower graph means the angular velocity of the robot. Fig. 12 shows simulation of motion generator with obstacle avoidance which is based on rules. You can download the source code of the program from our website (www.andyhouse.net).

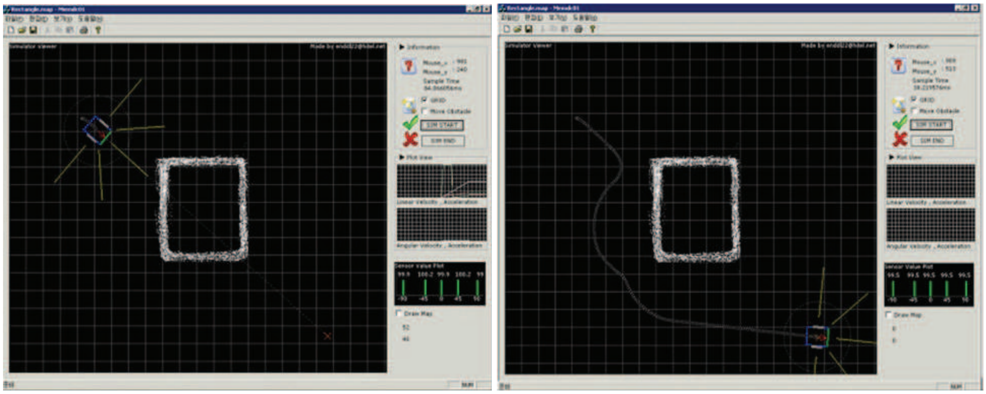

Fig. 12. Obstacle avoidance based on rules.

The greatest merit of obstacle avoidance based on rules is that it can generate avoidance angle without powerful computing ability. Using IF-Else sentence, we can develop the obstacle avoidance function easily. The disadvantage of obstacle avoidance based on rules is that robot may be trapped in loop. Fig. 13 shows the trapped situation. We call this as 'Local minima'. Many methods are used to overcome it. First, the robot can generate a virtual goal when it detects a trapped situation has occurred. Next step is path planning such as the A* and the gradient-method. These algorithms help to find the shortest path from global map which is the result of the map-building.

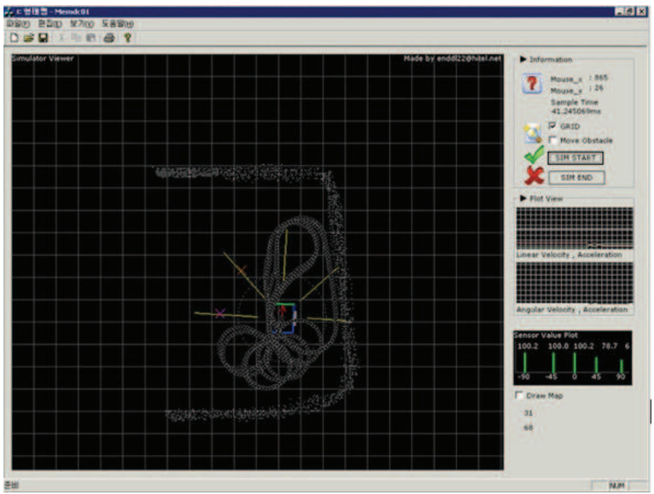

Fig. 13. Trapped situation.

3.2 Path planning and map-building method.

In this book, we use the gradient method which is based on grid-map to find shortest path. The gradient method is a kind of wavefront algorithm. The wavefront algorithm scans grid map through breadth first algorithm. It can be finished when robots reach the goal. If you want to know further, visit following sites:

Wavefront algorithm:

http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/16311/www/current/labs/lab05/

Gradient method (Kurt Konolige, 2000):

http://vision.cis.udel.edu/mrp/readings/gradient.pdf

The gradient method is fairly simple and powerful algorithm in path planning. We have developed the simulator to verify the gradient method we implemented. Fig. 14 shows the results of the path planning simulation.

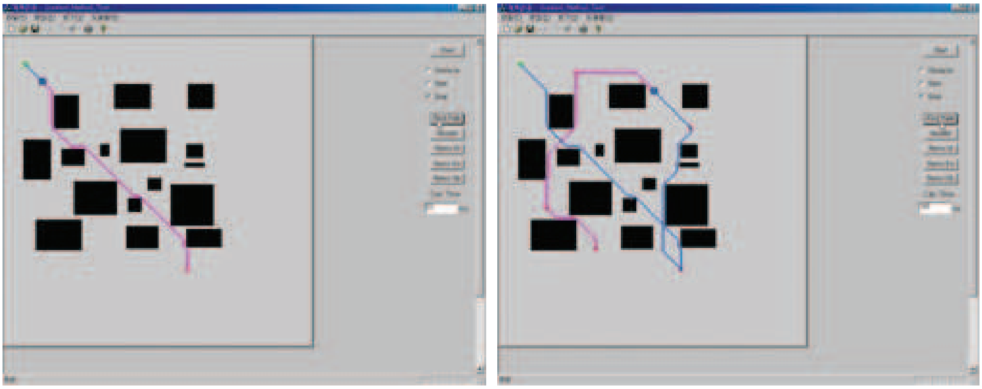

Fig. 14. The results of the path planning simulation

It took about 30ms to find the path in 500 x 500 grid map which is 25m x 25m. We have tested the experiment of path planning using real robot with the same algorithm we have tested in this simulator. It shows a precise motion to avoid several obstacles.

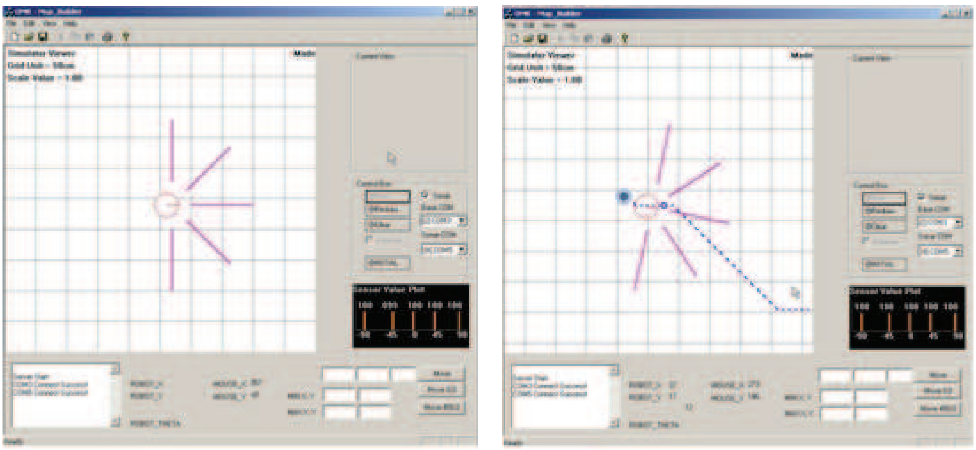

Fig. 15. The experimental results of navigation from start position to goal position.

The reason why we select the gradient method is speed. This algorithm reduces calculation time by stopping if it finds the goal position. Moreover, it can lessen the consumption by linear programming, adds the path which was newly calculated to the existing path. Finally, it can solve local minima problem which most of the path planning algorithms have. However it takes the time proportional to scale of the map. And the map must be stored in physical memory. Recently the power of computer is getting more and more powerful. Hence the path can be calculated in real time even up to 500 x 500 scaled map.

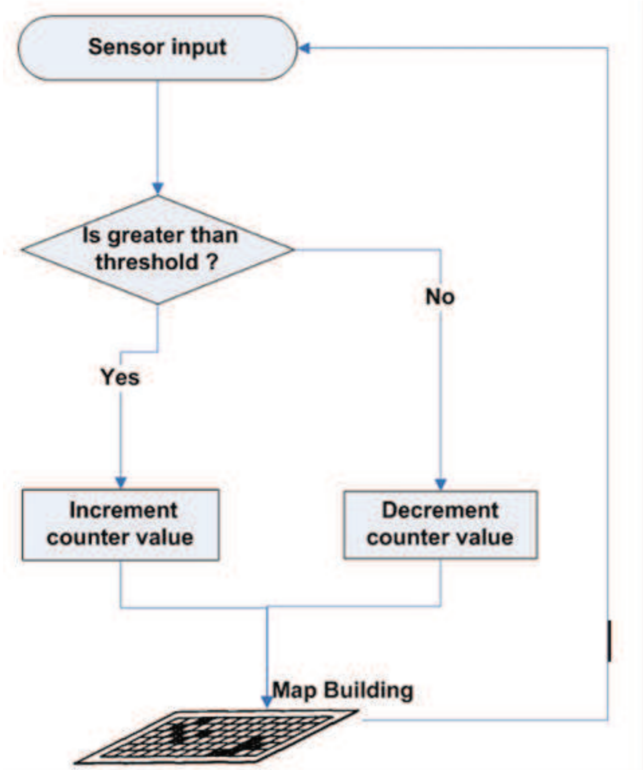

Fig. 16. Simple map-building process.

Part Timer uses ultrasonic sensors to make map. The characteristic of ultrasonic sensor is spreading. Hence it is limited by the shapes of surfaces and the density or consistency of the material. For example, foam on the surface of a fluid in a tank could distort a reading. The main issue is the updating of map. The map could be changed by an accumulated error while robot is moving or stationary. We couldn't compensate the dead-reckoning error because of limited ultrasonic sensors. Fig. 16 shows our simple map-building algorithm.
Fig. 17 shows the simple map-building on the real robot. The map building takes about 100ms. It means that the map building is executed up to 10 times in a second. Consequently we could embed the simple map-building into our real robot. It also describes the map building process while the robot is moving through a confusion area.

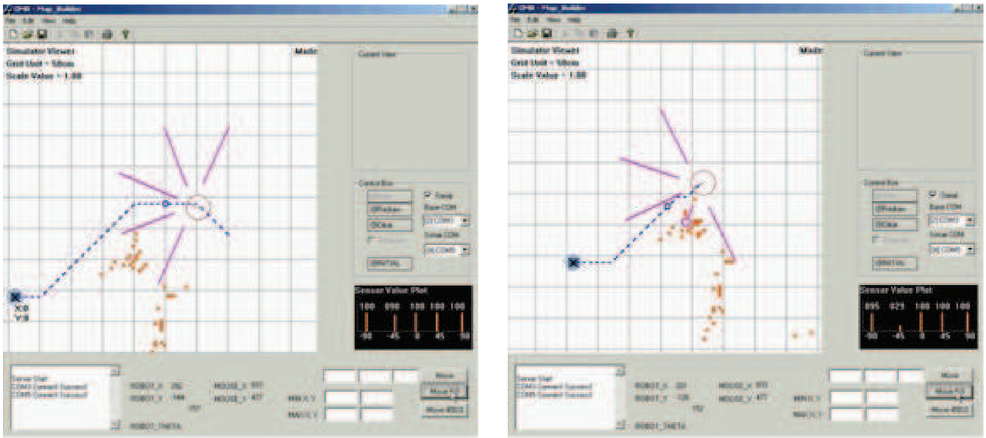

Fig. 17. simple map building experiment.

4. Manipulator

Manipulator is necessary to perform humans-like motion for robots. Especially, it requires grasping action for service robot as well. Recent service robots have manipulators even through they have no legs. It shows the importance of robot manipulators compared to moving actuators. Actually, the design of elaborate manipulator is not easy work. External factors like gravitation, collision with obstacles, and backlash of motors should be considered. Sensing scheme gives the robot surrounding information. In Part Timer system, 5-DoF Manipulator is designated with vision system that uses SIFT Algorithm for object recognition.

4.1 Considerations

There are several issues to implement vision-based grasping manipulation (Smith.C.E. et al., 1996). Two kinds of control systems; Open-Loop and Closed-Loop are mainly introduced. In case of the open-loop control, the vision system is used only to determine the object's position and orientation prior to a blind grasp. On the contrary, closed-loop control allows visual data to compensate for manipulator positioning inaccuracies and sensor noise. Typically, open-loop control requires more accurate calibration while closed-loop control requires faster vision system performance. For our system, we developed a high-performance object recognition algorithm choosing the close-loop control using SIFT (D.G.Lowe, 1999). It gives us more accurate results than open-loop systems because of compensating for noise data in real time manner. Although object's position may be changed during grasping time, closed-loop system would be able to track transition of target object. Because vision system requires high computational power, Pentium IV 2.4GHz is used for object recognition system.
For this, we placed a web-camera on the back of the hand. This placement allows camera to see wider scenes while grasping objects. We use a monocular camera that required no calibration and reduced hardware requirements. Unfortunately, monocular camera systems

cannot provide three-dimensional location of target object. That is why it is needed to recover depth through some different way. We have implemented approximation distance calculator by comparing camera image size to database image size.

The shape and complexity of a gripper is different from the object that the manipulator would try to grasp. We have designed the gripper that can grasp beverage cans to work in practical store services. The gripper has one degree of freedom so that it can work in two ways; closing, opening. It has one DC motor with a worm gear. Fig. 18 shows mechanical design of Part Timer's gripper.

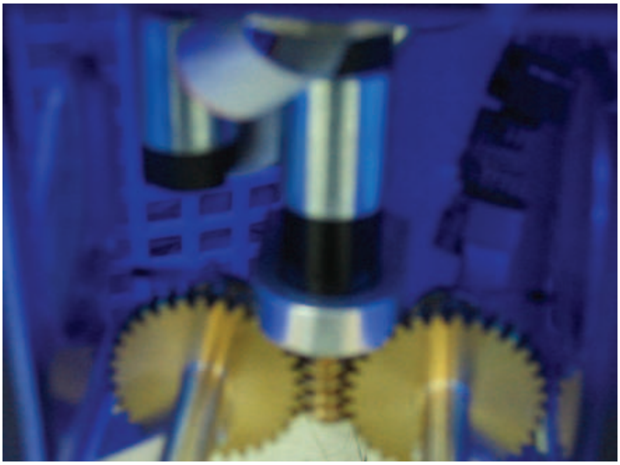

Fig. 18. Mechanical Design of Part Timer's Gripper.

Fig. 19 shows overall system architecture of the manipulator system. There are two main parts; vision system and control system. The vision system takes charge of the object recognition, and the control system is in charge of positioning the robot manipulator to the desired position. Therefore, the vision system includes the object recognition system and depth finder to calculate the 3-dimensional position in real time manner. In the control system, angle value of the each shaft will be calculated by inverse kinematics rule. Then, the manipulator positioning controls the motors to each angle value. These two separated systems are working independently from the main system to improve the performance.

The data flow of the whole manipulator is like below. First, the web-camera captures an image. The image coming from the web-camera goes to the object recognition system. The recognition system calculates a distance from the target object to the gripper. The method to find depth will be discussed in section 4.3. After that, the robot can get approximate coordinates (x, y, z) based on the pre-stored database information. In this time, the ultrasonic sensor is used to compensate errors because depth value may not guarantee accuracy. Next, the control system solves the inverse kinematics problem and moves motors to the desired position. The object recognition system receives the object position while the gripper is moving. Closed-loop system using the feedback of visual data is constituted in Part Timer System.

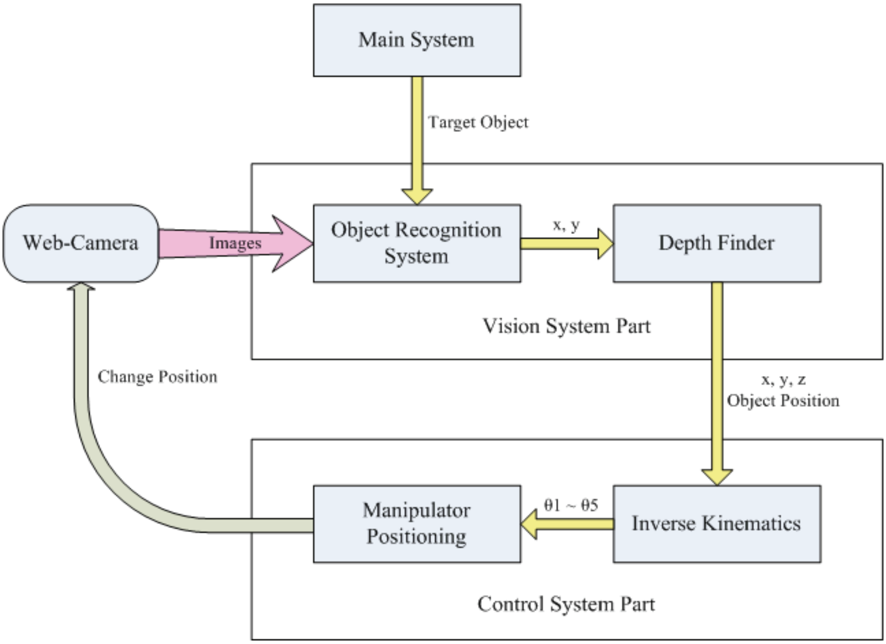

Fig. 19. Overall Manipulator System Architecture.

4.2 Design of robot manipulator

We have designed a 5-DoF robot manipulator to grasp objects. There are two motors on the shoulder (Roll, Pitch), one motor on the elbow (Pitch), and two motors on the wrist (Roll, Pitch). The end effecter also has one motor to drive the worm gear that controls the gripper.
There is one manipulator and we put it in front of the robot for the efficiency of the entire robot system. As each shaft has boundary in some area, entire workspace is quite small. To fulfill the delivery service, the robot has moving parts. Moving parts can work concurrently with manipulator through main scheduling system.

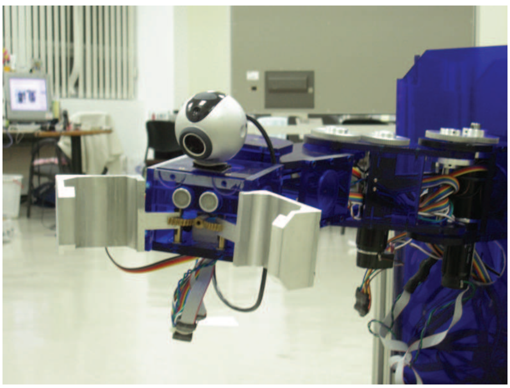

Fig. 20. The gripper with a camera and an ultrasonic sensor.

To grasp objects, the robot should interact with its environment. For example, it should perceive where the desired object is. If there is a camera on the robot on same height as humans eyes are, the robot cannot recognize objects which are far from the robot. General web camera specification is insufficient to see far objects. We put the web-camera on the back of the hand so that it can see the object closer and move its position during searching objects. Even if there are no objects on the camera screen, the robot can try to find the object by locating its end effecter to another position. Placing the camera on the hand is more useful than placing it on the head. One problem occurs if one camera is used. It is that the distance from camera to object is hard to calculate. Therefore, vision system roughly estimates the distance, and compensates the distance by using ultrasonic sensors. Fig. 20 shows the robot arm with a web-camera and an ultrasonic sensor.

4.3 Object recognition system

We use Scale-Invariant Feature Transform to recognize objects (D.G.Lowe, 1999). SIFT uses local features of input image, so it is robustness to scale, rotation and change of illuminations. Closed-loop vision system needs not only robust but also speed. Therefore, we have implemented basic SIFT algorithm and customized it for our robot system for speed up. Fig. 21 shows example of our object recognition system.

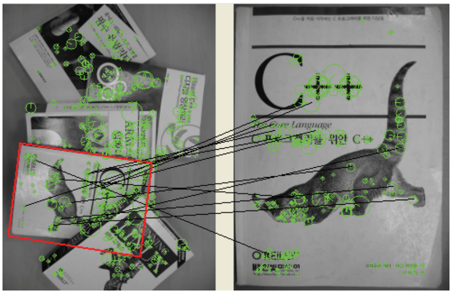

Fig. 21. Example of object recognition.

Unfortunately, the robot has just one camera on the hand, so it is not able to estimate exact distance like when using stereo vision. Therefore, more specific distance for the object database is required to calculate the distance using only one camera. When we make the object database, the robot should know the distance from the object to the camera. Then we calculate the distance comparing the area of object and the size of the database. The size of the object is inversely proportional to the square of the distance. Fig. 22 shows the relationship between the area size and the distance.

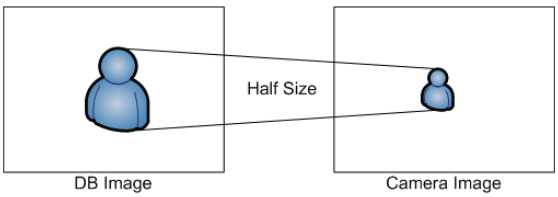

Fig. 22. The relationship between the area and the distance.

We assume that if the object is half the size of the database image, the area of the object will be quarter of the database image. The result of this equation is not correct, but we can adapt the equation to our system to get the distance roughly. Using this law, the robot can calculate the ratio between the area and the distance. The equations are,

$$d_a : d_b \cong s_a : \sqrt{s_b} \tag{8}$$

$$d_b \cong \frac{d_a \times \sqrt{s_b}}{s_a} \tag{9}$$

Variable d indicates the distance, and s indicates the size(or area). Small a and b indicate input image and database image respectively. Eq. (8) shows the relationship between distance and area. According to the relation, Eq. (9) shows how we get the approximate distance from the difference of area.

We use SIFT transformation matrix to locate the position of the object in the scene. We can get the transformation matrix if there are three matching points at least (D.G.Lowe, 1999). The transformation matrix indicates the object's location and its orientation. Then the manipulator control system moves motors to locate the end effecter at the center position of the object. However, there are some errors about 3~4cm within work space because of the object shape and database error. Even if very small error occurs, the manipulator has many chances to fail to grasp objects. That is why we use ultrasonic sensors, SRF-04 (Robot Electronics Ltd.,), to compensate errors. The robot computes the interval by measuring the ultrasonic returning time. This sensor fusion scheme reduces most failure ratio.

4.4 Flow chart

Even if manipulator system is accurate and robust, it may not be possible to grasp the object, only using the manipulator system. It requires that the integration of entire robot system. We present the overall robot system and flowchart for grasping objects. Grasping strategy plays an important role in system integration. We assumed some cases and started to integrate systems based on a scenario. In practical environment, there are many exceptional cases that we could not imagine. We though that the main purpose of the system integration is to solve the problems that we faced. Fig. 23 shows the flowchart of grasp processing.

First, the robot goes to the pre-defined position where the desired object is near by. Here, we assumed that the robot knows approximately the place where the object is located. After moving, the robot searches the object by using its manipulator. If the robot finds the desired object, it moves to the location of the object in the workspace. That is why the scanning process is necessary as the web camera is able to search further range than the manipulation workspace. The moving part of the robot is using different computing resources, so we can process the main scheduler and the object recognition in parallel. Fig. 24 shows the movement of the robot when the object is outside of the workspace.

After that, the robot moves the manipulator, so that the object is at the center of the camera by solving inverse kinematics problem. In this time, the image data will be captured and continually used for vision processing. If the object is in the workspace, the robot holds out its manipulator while the ultrasonic sensor is checking whether the robot can grasp the object or not. If the robot decides that it is enough distance to grasp the object, the gripper would be closed to grasp. Using the processing described above, the robot can grasp the object.

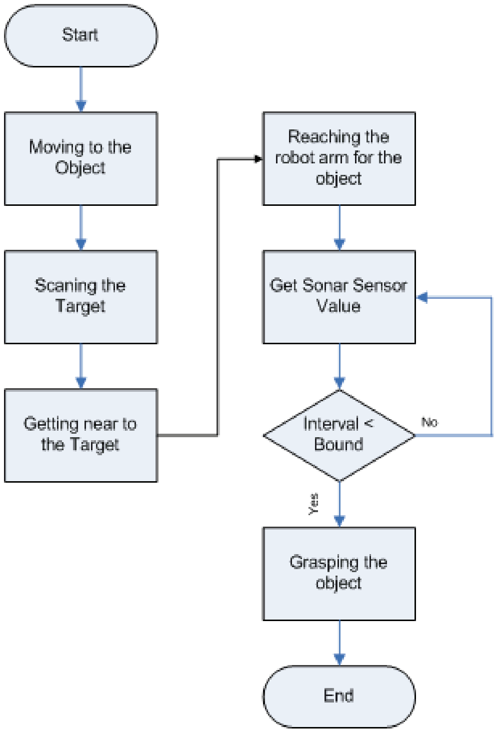

Fig. 23. The flowchart of grasping processing.

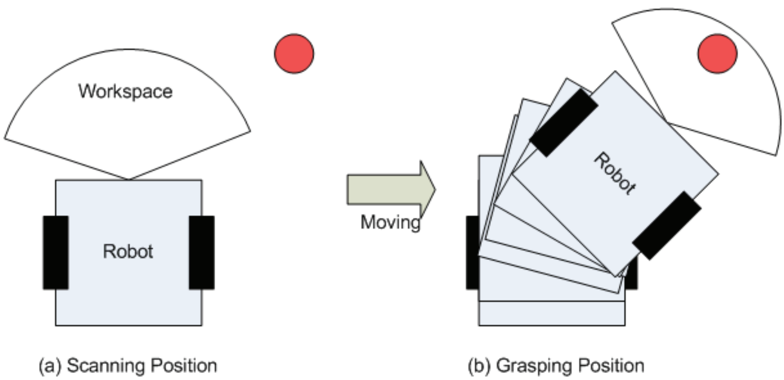

Fig. 24. Movement of the robot after scanning objects.

Fig. 25 presents the processing after the robot has found the desired object. First, the robot arm is in an initial state. If the robot receives a scanning command from the main scheduler, the object recognition system starts to work and the robot locates its manipulator to other position. If the robot finds the object, the manipulator will reach out. The ultrasonic sensor is used in this state. Reaching the robot manipulator, the ultrasonic sensor is checking the distance from the object. Finally, the gripper is closed.

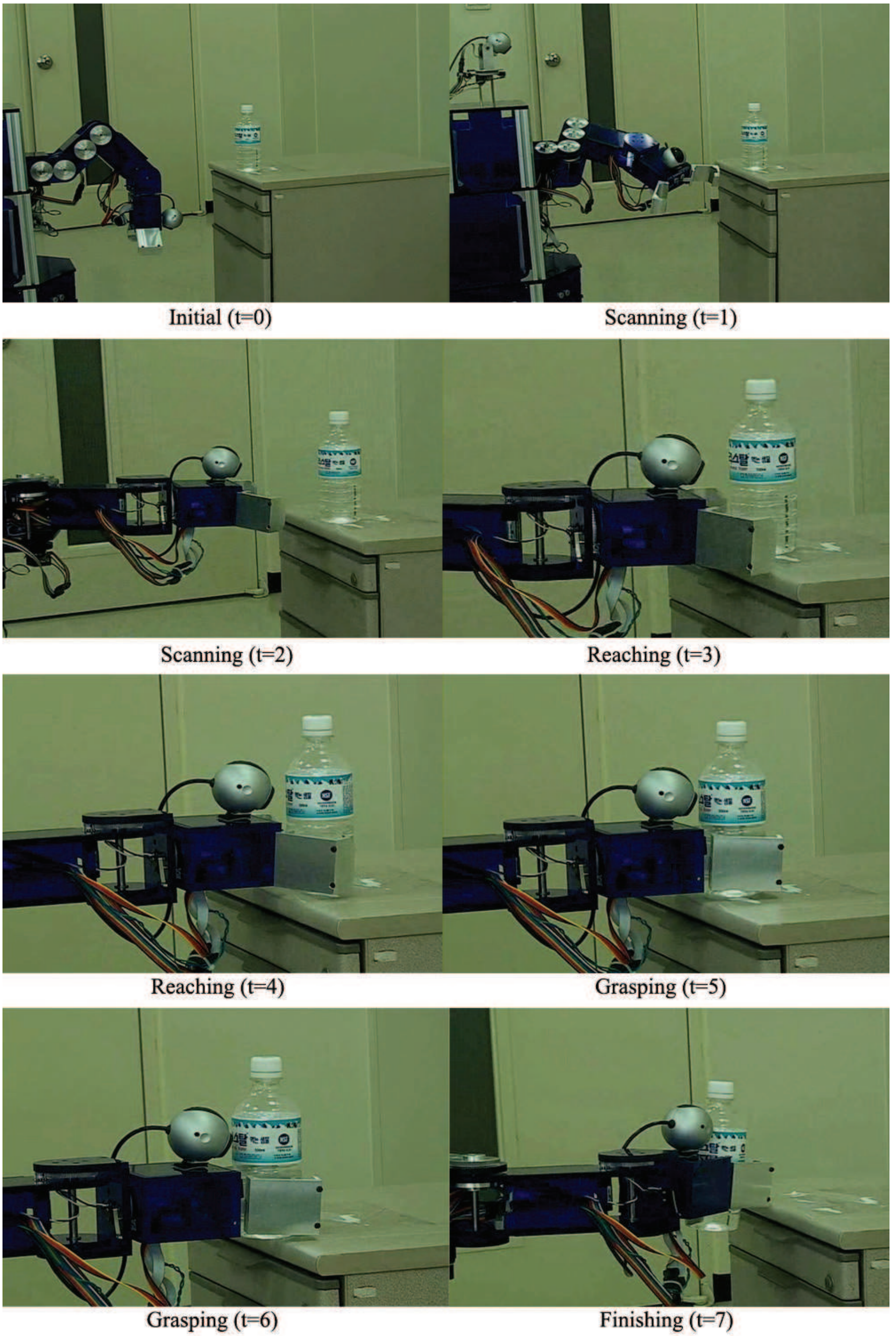

Fig. 25. Time progressing of the grasping action.

5. Face feature processing

The face feature engine consists of three parts; the face detection module, the face tracking module, and the face recognition module. In the face detection module, the final result is the nearest face image from the continuous camera images using CBCH algorithm. In the face tracking module, Part Timer tracks the detected face image using pan-tilt control system and fuzzy controller to make the movement smooth. In the face recognition module, it recognizes who the person is using CA-PCA. Fig. 26 shows the block diagram of face feature engine. The system captures the image from the camera and sends it to the face detection module to detect the nearest face image among the detected face images. And the face image is sent to the face tracking module and the face recognition module.

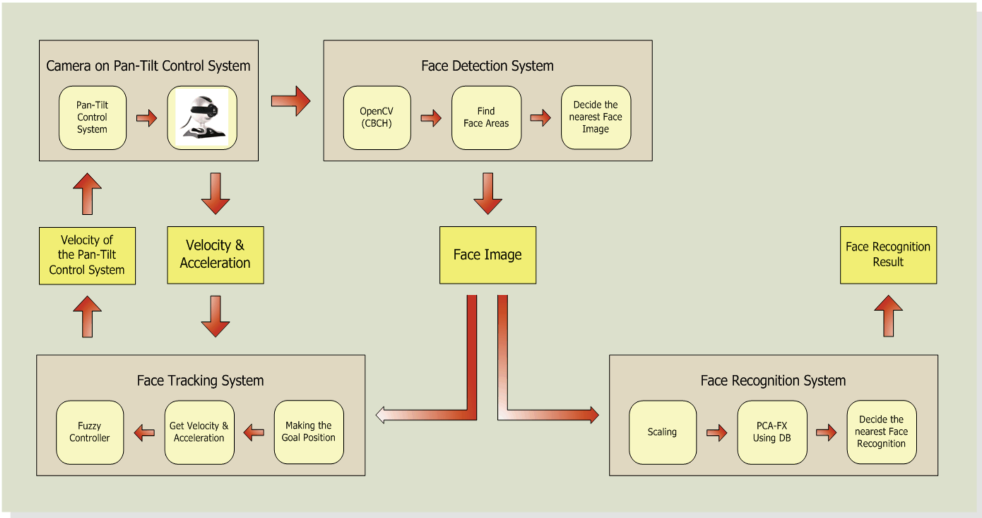

Fig. 26. The block diagram of face feature engine.

The face detection module uses facial feature invariant approach. This algorithm aims to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use these to locate faces. This method is designed mainly for face localization. It uses OpenCV library (Open Computer Vision Library) that is image processing library made by Intel Corporation. It not only has lots of image processing algorithms but also is optimized for Intel CPU so it shows fast execution speed. And they opened the sources about algorithms of OpenCV so we can amend the algorithms in our own way. To detect face, we use the face detection algorithm using CBCH (cascade of boosted classifiers working with Haar-like features) (Jos´e Barreto et al., 2004). The characteristic of the CBCH is fast detection speed, high precision and simple calculation of assorter. We use the adaboost algorithm to find fit compounding of Haar assorter and it extracts the fittest Haar assorter to detect face among all of the possible Haar assorter in order. Of course it shows the calculated result, the weight for each Haar assorter, because each one has different performance. And the extracted Haar assorters discriminate whether it is face area or not and distinguish whether it is face image or not by a majority decision. Fig. 27 shows the results of the face detection module.

Fig. 27. The results of the face detection module. In case there is just one person (left), there are three persons (right).

The face tracking module uses the fuzzy controller to make the movement of pan-tilt control system stable. Generally, the fuzzy controller is used to compensate real time operation of the output about its input. And it is also used by the system which is impossible to model mathematically. We use the velocity and the acceleration of the pan-tilt system for the input of the fuzzy controller and get the velocity of the pan-tilt system for the output. Table 2 shows the inputs and the output. We design the fuzzy rule like Fig. 28 and Fig. 29 is its graph. Fig. 28 means that if the face is far from center of camera image, move fast and if the face is near to center of camera image, move little.

	Pan (horizontality)	Tilt (verticality)
Input 1	Velocity (-90 ~ 90)	Velocity (-90 ~ 90)
Input 2	Acceleration (-180 ~ 180)	Acceleration (-180 ~ 180)
Output	Velocity of Pan (-50 ~ 50)	Velocity of Tilt (-50 ~ 50)

Table 2. The Input and output of the pan-tilt control system.

Pan-tilt		Acceleration				
		Left Fast	Left Slow	Zero	Right Slow	Right Fast
Velocity	Left Large	LL	LL	LL	L	Zero
	Left Small	LL	LL	L	Zero	R
	Zero	LL	L	Zero	R	RR
	Right Small	L	Zero	R	RR	RR
	Right Large	Zero	R	RR	RR	RR

Fig. 28. The fuzzy controller of pan-tilt system.

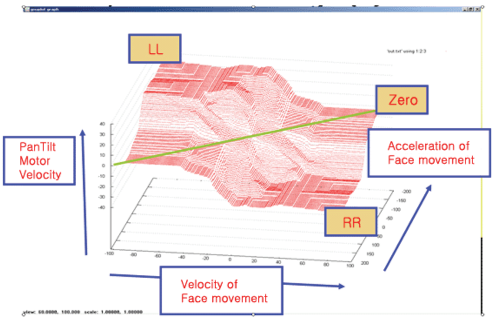

Fig. 29. The graph of Fig. 28.

The face recognition engine uses CA-PCA algorithm using both the input and the class information to extract features which results in better performance than conventional PCA (Myuong Soo Park et al., 2006). We built the facial database to train recognition module. Our facial database consists of 300 gray scale images of 10 individuals with 30 different images. Fig. 30 shows the results of face classification in Part Timer.

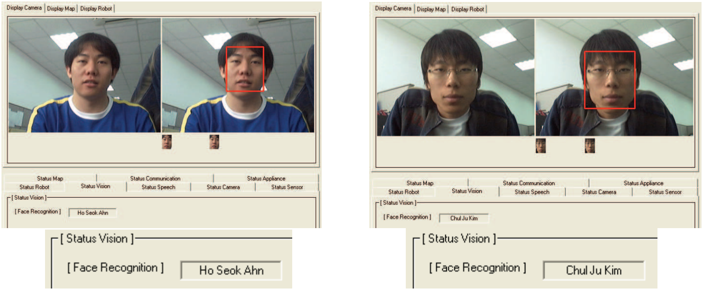

Fig. 30. The results of face classification in Part Timer.

6. Conclusion

Part Timer is unmanned store service robot and it has lots of intelligent functions; navigation, grabbing objects, gesture, communication with humans, recognizing face, object and character, surfing the internet, receiving calls, etc. It has a modular system architecture which uses intelligent macro core module for easy composition of whole robot. It offers remote management system for humans who are outside of the store. We have participated in many intelligent robot competitions and exhibitions to verify the performance. We won the first prize in many of these competitions; Korea Intelligent Robot Competition, Intelligent Robot Competition of Korea, Robot Grand Challenge, Intelligent Creative Robot Competition, Samsung Electronics Software Membership Exhibition, Intelligent Electronics Competition, Altera NIOS Embedded System Design Contest, IEEE RO-MAN Robot Design Competition, etc. Although Part Timer is unmanned store service robot, it can be used for office or home robots as well. As essential functions for service robot similar, it could be used for other purpose if the system architecture we introduced is used. For future work, we are applying the system architecture to multi robot system in which it is possible to cooperate with other robots.

8. References

Sakai K., Yasukawa Y., Murase Y., Kanda S. & Sawasaki N. (2005). Developing a service robot with communication abilities, *In Proceedings of the 2005 IEEE International Workshop on Robot and Humans Interactive Communication (ROMAN 2005)*, pp. 91-96.

Riezenman, M.J. (2002). Robots stand on own two feet, *Spectrum, IEEE*, Vol. 39, Issue 8, pp. 24-25.

Waldherr S., Thrun S. & Romero R. (1998). A neural-network based approach for recognition of pose and motion gestures on a mobile robot, *In Proceedings of Brazilian Symposium on Neural Networks*, pp. 79-84.

Mumolo E., Nolich M. & Vercelli G. (2001). Pro-active service robots in a health care framework: vocal interaction using natural language and prosody, *In Proceedings of* *the 2001 IEEE International Workshop on Robot and Humans Interactive Communication (ROMAN 2001)*, pp. 606-611.

Kleinehagenbrock M., Fritsch J. & Sagerer G. (2004). Supporting advanced interaction capabilities on a mobile robot with a flexible control system, *In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Vol. 4, pp. 3649-3655.

Hyung-Min Koo & In-Young Ko (2005). A Repository Framework for Self-Growing Robot Software, *12th Asia-Pacific Software Engineering Conference (APSEC '05)*.

T. Kanda, H. Ishiguro, M. Imai, T. Ono & K. Mase (2002). A constructive approach for developing interactive humansoid robots, *In Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, pp. 1265–1270.

Nakano, M. et. al. (2005). A two-layer model for behavior and dialogue planning in conversational service robots, *In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 3329-3335.

Gluer D. & Schmidt G. (2000). A new approach for context based exception handling in autonomous mobile service robots, *In Proceedings of the 2000 IEEE/RSJ International Conference on Robotics & Automation (ICRA 2000)*, Vol. 4, pp. 3272-3277.

Yoshimi T. et. al. (2004). Development of a concept model of a robotic information home appliance, ApriAlpha, *In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Vol. 1, pp. 205-211.

Dong To Nguyen, Sang-Rok Oh & Bum-Jae You (2005). A framework for Internet-based interaction of humanss, robots, and responsive environments using agent technology, *IEEE Transactions on Industrial Electronics*, Vol. 52, Issue 6, pp. 1521-1529.

Jeonghye Han, Jaeyeon Lee & Youngjo Cho (2005). Evolutionary role model and basic emotions of service robots originated from computers, *In Proceedings of the 2005 IEEE International Workshop on Robot and Humans Interactive Communication (ROMAN 2005)*, pp. 30-35.

Moonzoo Kim, Kyo Chul Kang & Hyoungki Lee (2005). Formal Verification of Robot Movements - a Case Study on Home Service Robot SHR100, *In Proceedings of the 2005 IEEE/RSJ International Conference on Robotics & Automation (ICRA 2005)*, pp. 4739-4744.

Taipalus, T. & Kazuhiro Kosuge (2005). Development of service robot for fetching objects in home environment, *In Proceedings of the 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 2005)*, pp. 451-456.

Ho Seok Ahn, In-kyu Sa & Jin Young Choi (2006). 3D Remote Home Viewer for Home Automation Using Intelligent Mobile Robot, *In Proceedings of the SICE-ICASE International Joint Conference 2006 (ICCAS2006)*, pp. 3011-3016.

Sato M., Sugiyama A. & Ohnaka S. (2006). Auditory System in a Personal Robot, PaPeRo, *In Proceedings of the International Conference on Consumer Electronics*, pp. 19-20.

Jones, J.L. (2006). Robots at the tipping point- the road to iRobot Roomba, *IEEE Robotics & Automation Magazine*, Vol. 13, Issue 1, pp. 76-78.

Sewan Kim (2004). Autonomous cleaning robot: Roboking system integration and overview, *In Proceedings of the 2004 IEEE/RSJ International Conference on Robotics & Automation (ICRA 2004)*, Vol. 5, pp. 4437-4441.

Prassler E., Stroulia E. & Strobel M. (1997). Office waste cleanup: an application for service robots, *In Proceedings of the 1997 IEEE/RSJ International Conference on Robotics & Automation (ICRA 1997)*, Vol. 3, pp. 1863-1868.

Houxiang Zhang, Jianwei Zhang, Guanghua Zong, Wei Wang & Rong Liu (2006). Sky Cleaner 3: a real pneumatic climbing robot for glass-wall cleaning, *IEEE Robotics & Automation Magazine*, Vol. 13, Issue 1, pp. 32-41.

Hanebeck U.D., Fischer C. & Schmidt G. (1997). ROMAN: a mobile robotic assistant for indoor service applications, *In Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1997)*, Vol. 2, pp. 518-525.

Koide Y., Kanda T., Sumi Y., Kogure K. & Ishiguro H. (2004). An approach to integrating an interactive guide robot with ubiquitous sensors, *In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Vol. 3, pp. 2500-2505.

Fujita M. (2004). On activating humans communications with pet-type robot AIBO, *Proceedings of the IEEE*, Vol. 92, Issue 11, pp. 1804-1813.

Shibata T. (2004). An overview of humans interactive robots for psychological enrichment, *Proceedings of the IEEE*, Vol. 92, Issue 11, pp. 1749-1758.

Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides (1994). *Design Patterns*, Addison Wesley

Jin Hee Na, Ho Seok Ahn, Myoung Soo Park & Jin Young Choi (2005). Development of Reconfigurable and Evolvable Architecture for Intelligence Implement. *Journal of Fuzzy Logic and Intelligent Systems*, Vol. 15, No. 6, pp. 35-39.

Konolige, K. (2000). A gradient method for realtime robot control, *In Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, Vol. 1, pp. 639-646

Rosenblatt J (1995). DAWN: A distributed Architecture for Mobile Navigation, *Spring Symposium on Lessons Learned for Implemented Software Architecture for Physical Agent*, pp. 167-178.

Roland Siegwart (2007). Simultaneous localization and odometry self calibration for mobile robot, *Autonomous Robots*. Vol. 22, pp. 75–85.

Seung-Min Baek (2001). Intelligent Hybrid Control of Mobile Robotics System. *The Graduate School of Sung Kyun Kwan University*.

Smith.C.E. & Papanikolopoulos.N.P(1996). Vision-Guided Robotic Grasping: Issues and Experiments. *In Proceedings of the 1996 IEEE/RSJ International Conference on Robotics & Automation (ICRA 2004)*, Vol. 4, pp. 3203-3208.

D.G.Lowe(1999). Object recognition from local scale-invariant features. *In Proceedings of the 1999 International Conference on Computer Vision (ICCV 1999)*, pp. 1150-1157.

D.G.Lowe(2004). Distinctive image features from scale invariant keypoints. *In Proceedings of the 2004 International Journal of Computer Vision (IJCV 2004)*, pp. 91-110.

Jos´e Barreto, Paulo Menezes & Jorge Dias (2004). Humans-Robot Interaction based on Haar-like Features and Eigenfaces. *In Proceedings of the 2004 IEEE/RSJ International Conference on Robotics & Automation (ICRA 2004)*, Vol. 2, pp. 1888-1893.

Park, M.S., Na, J.H. & Choi, J.Y. (2006). Feature extraction using class-augmented principal component analysis, *In Proceedings of the International Conference on Artificial Neural Networks)*, Vol. 4131, pp. 606-615.

Advances in Service Robotics

Edited by Ho Seok Ahn

ISBN 978-953-7619-02-2
Hard cover, 342 pages
Publisher InTech
Published online 01, July, 2008
Published in print edition July, 2008

This book consists of 18 chapters about current research results of service robots. Topics covered include various kinds of service robots, development environments, architectures of service robots, Human-Robot Interaction, networks of service robots and basic researches such as SLAM, sensor network, etc. This book has some examples of the research activities on Service Robotics going on around the globe, but many chapters in this book concern advanced research on this area and cover interesting topics. Therefore I hope that all who read this book will find lots of helpful information and be interested in Service Robotics. I am really appreciative of all authors who have invested a great deal of time to write such interesting and high quality chapters.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ho Seok Ahn, In-Kyu Sa, Young Min Baek and Jin Young Choi (2008). Intelligent Unmanned Store Service Robot "Part Timer", Advances in Service Robotics, Ho Seok Ahn (Ed.), ISBN: 978-953-7619-02-2, InTech, Available from: http://www.intechopen.com/books/advances_in_service_robotics/intelligent_unmanned_store_service_robot_part_timer

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821