Research Article · DOI: 10.2478/pjbr-2013-0013 JBR · 4(2) · 2013 · 63-72

Cooking Behavior with Handling General Cooking Tools based on a System Integration for a Life-sized Humanoid Robot

Yoshiaki Watanabe[1]*, Kotaro Nagahama[2], Kimitoshi Yamazaki[3],†, Kei Okada[2], Masayuki Inaba[2]

1 Graduate School of Interdisciplinary Information Studies, The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan

2 Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan

3 Faculty of Engineering, Shinshu University, 4-18-1, Wakasato, Nagano, Nagano, Japan

Received 23-03-2013

Accepted 19-09-2013

Abstract

This paper describes a system integration for a life-sized robot working at a kitchen. On cooking tasks, there should be various tools and foods, and cooking table may have reflective surface with blots and scratch. Recognition functions should be robust to noises derived from them. As other problems, cooking behaviors impose motion sequences by using whole body of the robot. For instance, while cutting a vegetable, the robot has to hold one hand against the vegetable even if another hand with a knife should be moved for the cutting. This motion requires to consider full articulation of the robot simultaneously. That is, we have difficulties against both recognition and motion generation. In this paper we propose recognition functions that are to detect kitchen tools such as containers and cutting boards. These functions are improved to overcome the influence of reflective surface, and combination shape model with task knowledge is also proposed. On the other hand, we pointed out the importance of the use of torso joints while dual arm manipulation. Our approach enables the robot to keep manipulability of both arms and viewing field of a head. Based on these products, we also introduce an integrated system incorporating recognition modules and motion generation modules. The effectiveness of the system was proven through some cooking applications.

Keywords

Life-sized robot · system integration · cooking

1. Introduction

In these days, there are growing concerns about the demand to robots that work in daily environment. In fact, some robots doing a specific task have been marketed. For instance, cleaning robots such as roomba [1] are able to remove dusts on floors automatically. On the other hand, versatile robots doing various daily tasks have begun to be required. For the robots, recognition and manipulation functions that are available at various daily situations are needed.

In this paper, we focus on cooking that is one of important tasks in everyday living. Because cooking food is a behavior that can be broken down a set of procedural and parallel actions, automated machines have a potential to help it. One of the assisting methods is to reserve and to provide information that is needed for knowing step-by-step recipe. Cooking Navi [2] and CounterActive [3] performed cooking assistance by showing recipes on a display. In addition to display, CounterIntelligence [4] and Future of the Kitchen [5] used sensors attached at kitchen. They were used to provide recipe information on the moment, or to store recipes on site. As an application to robotics, Nakauchi et al. [6] introduced a robot that instructed a user about recipes by presuming his cooking behavior.

On the other hand, we can find intelligent autonomous robots that performed kitchen tasks alternative to human. Bollini et al. [7] developed a robot system that made cookies based on automatic recipe construction and action planning. Srinviasa et al. [8] developed a mobile manipulator *HERB*. They demonstrated a fetch-and-carry task that the robot removed an object from a refrigerator, and handed the object on a person. As other applications, Okonomiyaki [9] and Ramen [10] were targeted.

From above researches, it would appear that cooking assistance can be achieved by means of two approaches: (i) sensors and displays put in environment are used to assist human behaviors, (ii) independent intelligent robots perform cooking tasks on behalf of human. In this research, we tried second approach. Cooking food is one of heavy routines for daily living because it includes many manipulations of foods and tools while long working hours. If autonomous robots perform a part of the cooking, it will be useful.

There are a plenty of issues when we make a dual-arm robot do a cooking. For instance, the robot has to operate many types of foods that the shapes of them are not uniform among same species. Some of cooking tools such as knives have simple texture and reflective body. Recognition modules that can cope with these issues are required to the robot. As another issue, the variety of the dual-arm motions according to manipulation targets, tools, and cooking situation, should be considered. Our goal is to show the behavior of cooking by an au-

*E-mail: {watanabe, nagahama, k-okada, inaba}@jsk.t.u-tokyo.ac.jp
†E-mail: yamazaki@shinshu-u.ac.jp

tonomous robot using an integrated system coping with above issues. This paper shows the experimental results of basic cooking behaviors; cutting, peeling, transferring. An experiment about consecutive behaviors of making a salad is also reported.
This paper is organized as follows: section 2 denotes our concept and approaches. Section 3 explains about recognition and manipulation modules for cooking by a dual-arm robot. Section 4 presents some experiments using a real robot, and section 5 concludes this paper.

2. Concepts and approaches for designing cooking behaviors by a lifesized humanoid robot

Many types of robots that are for daily living assistances including cooking have been proposed. Kosuge [11] developed a single-arm robot that put a dish in a kitchen cabinet automatically. The robot had an ability to grasp plates with various types of shape, even if several plates were stacked on a table. Igarashi et al. [12] used multiple small robots, and they succeeded to make a dish based on a coordination system for them. The results of these researches definitely provided meaningful assistance. Dextrous manipulation was done by the former robot. On the other hand, the latter robot had wide working space. However, it was difficult for these robots to perform complex cooking behavior, for instance, they did not perform cutting vegetables because of the lack of two or more end-effectors contacting to foods and cooking tools. In this research, we use a life-sized humanoid robot to clear away such restriction. It has possibilities to have both dextrous manipulation skill and wide working space.
Recently, humanoid robots have become a research platform for daily living support. Asfour et al. [13] developed multi-degree of freedom robot *ARMAR* that worked at human's living space. They achieved fetch and carry tasks based on sensor information processing and action sequence planning. For object recognition, the robot equipped image-based recognition functions that used 3D shape model as object knowledge. However, there are still inevitable knowledge such as global object position and grasping method of the object. These knowledge did not systemized in their research.
Justin developed at DLR was a dual-arm mobile manipulator. Its sophisticated hardware and motion control technique were proven by tasks of making a tea [14] and a coffee [15]. *Twendy-one* that was developed by Sugano et al. [16] was composed of flexible joints and could perform dextrous manipulation such as handling tongs to pinch a small object. These robots succeeded to prove their dexterity performance. On the other hand, because their main focuses did not recognition and motion planning, difficulties about them were not considered. For instance, when a robot cut a piece of vegetable by using a knife, both arm and torso should be considered simultaneously in the calculation of inverse kinematics, because one problem of dual arm manipulation is narrow movement range. As other things, failure detection and recovery are very important to sure working.
Based on these discussions, we focus on both recognition and motion generation that are needed for daily cooking action. On that basis, a system integration for a life-sized humanoid robot is proposed. The system concept will be explained at next section.

2.1. A concept of designing recognition modules

A concept of designing recognition modules is as follows:

- Our integrated system has a set of knowledge (we call them 'task knowledge') that are necessary to complete given tasks even if a robot targets various types and shapes of foods. The inputs of recognition or manipulation modules are the knowledge and actual sensor data.

The task knowledge is necessary information on recognizing and handling objects. It includes the following information.

- Features of each manipulated object (e.g. stiffness)
- Grasping method (e.g. grasp position and direction embedded on an object model)
- Existence region (a list of places that a robot has already found a target object or has been taught where the target exists).

Based on the concept, we can get following two practical points for designing behaviors in daily environments.

I. **Ease of designing recognition module with reusability, and their combination.**
It is difficult to perform completely accurate recognition because of various types of errors. In the case that an error hinder an original behavior, a robot may displace its planned motion to cancel out the error. On the other hand, if a big error prevents to continue originally planned motion, a new behavior to recover from the condition should be re-planned. Anyway, it is needed that the robot can recognize the level of errors. By preparing a set of small recognition modules that refer task knowledge to detect the level, it is possible to design error detection and recovery routine by combining the modules.

Also, combination with several sensors is important to enhance one recognition purpose. For instance, force sensor on arms enables the robot to know manipulation condition even when it is difficult to observe by a camera looking down the manipulation.

II. **Reduction of the recognition error for identifying object position**
Most of previous researches on object recognition by using images were based on local texture or color. That is, the knowledge about each object is used for detect the actual position of the object.

In many manipulation tasks, because we are able to give rough position of the target object to the robot, what the robot should perform is to narrow down the existence range. For this reason, rough position information is included in task knowledge as 'search area', and it is used to decrease possibility of false recognition.

2.2. A concept of designing manipulation modules

There are two major policies for our manipulation modules.

- To find a feasible manipulation pose for whole body, the calculation of inverse kinematics is alternated between two arms.
- The amount of torso motion is repressed as much as possible while calculating the inverse kinematics.

There were many of researches coping with dual arm manipulation [15] [17]. In most cases, a static object was grasped by one arm while the other arm has already held it. The problem to the manipulation

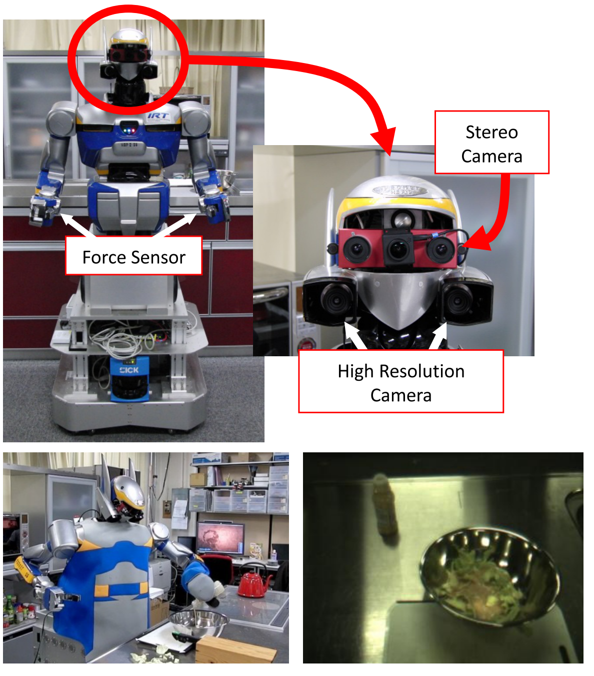

Figure 1. A life-sized robot working for making a salad

is that the solution space of inverse kinematics is significantly reduced comparing with the case of a single arm. In the case of humanoid robot, torso link is connected to both left and right arms. So if the robot rotates a torso for stretching one end-effector, the other end-effector may not have a solution to reach its target pose. To ensure big solution space, torso link should be used, but a small amount of motion is permitted in our calculation.

In this research, we target three cooking behavior; cutting, peeling and transferring. Vegetables such as cucumber, cabbage and potato are used for proof experiments. Recognition and manipulation functions that are designed along above concepts are integrated into a system for a humanoid robot.

Fig. 1 shows a humanoid robot to be used. It has an omnidirectional wheelbase, 2-DOF torso, and 7-DOF dual arms. Force sensors are embedded on the wrists. The type of end-effector is a jaw gripper. One stereo camera and two high resolution cameras are equipped on the head.

3. Integrated system for life-sized humanoid robots to cook

In this section, we describe the recognition and manipulation modules for the system under the concepts written in section 2.

3.1. Architecture of the integrated system

Fig. 2 shows the overall architecture of the system. Each module to recognize vegetables or tools is developed with respect to each target object and purpose. Their inputs are sensor values and knowledge for the tasks, which are described in the bottom section of Fig. 2. The output of each recognition module is the position of the target. The manipulation modules use the outputs of the recognition modules to modify the trajectory for the motions of the arms, and execute it. However some kinds of manipulation modules, such as force feedback controller for peeling tasks, use the sensor values directly. Additionally, the system has modules to evaluate the result of each operation and to execute a trajectory to fix the situation. If the result of the operation is not desirable one, the module changes an undesirable situation to the correct one. For example, if the robot finds that it failed in cutting a vegetable, the robot tries a cutting motion once more. If the robot detects a peeled coat on a vegetable, the robot shakes the vegetable to remove the coat. The details of the modules for recognition and manipulation are described below.

3.2. Recognition of tools and foods in a kitchen

Methods to recognize tools and vegetables with a stereo camera are described in this section. In kitchen environments, it is difficult to detect objects and to calculate the accurate position of them because of the reflected light from metal objects and the occlusion by blots.

3.2.1. Recognition of a cutting board using a 3-D shape model

This part describes a method to detect a cutting board. In general, foods like vegetables are placed on a cutting board when they are processed. Therefore the area of a cutting board can be used for the search area to find the foods. Normally the shape of a cutting board does not change, and there are not so many types of cutting boards in each family. Therefore, the 3-D shape model of a cutting board is useful for the template to search it.

Many of methods for object recognition employ a matching between image features and 3D shape model [18] [19]. In our case, almost all of our target objects do not have rich texture, and around environment includes reflection and scratch. From these reasons, image edges are used as image feature to be matched with 3D shape model. because such information is relatively robust to noise. In addition, for increasing robustness, particle filter is applied [17].

Eq. 1 shows a formula to calculate a likelihood of the particle $\mathbf{x}_t$ at time t. The center figure of Fig. 3 shows a result of detecting a cutting board. The particle with the highest likelihood is drawn with the thick green line. The equation for updating the filter is as follows:

$$p(\mathbf{x}_t|\mathbf{z}_t) = exp\frac{(D_{edge}(E^{2D}, E^{2D}_{ref}))^2}{2\sigma^2_{edge}}. \tag{1}$$

The 3D pose of the model is represented by hypothesis $\mathbf{x}_t$ that is evaluated using sensor data $\mathbf{z}_t$ (in this process, image edges). E^{2D} is a set of edge segments extracted from an input image, and E^{2D}_{ref} is a set of reference edges in image into which a 3D shape model is projected. σ_{edge} is a pre-defined constant value. D_{edge} is a set of functions each of which searches one edge segment which is the closest to one ref-

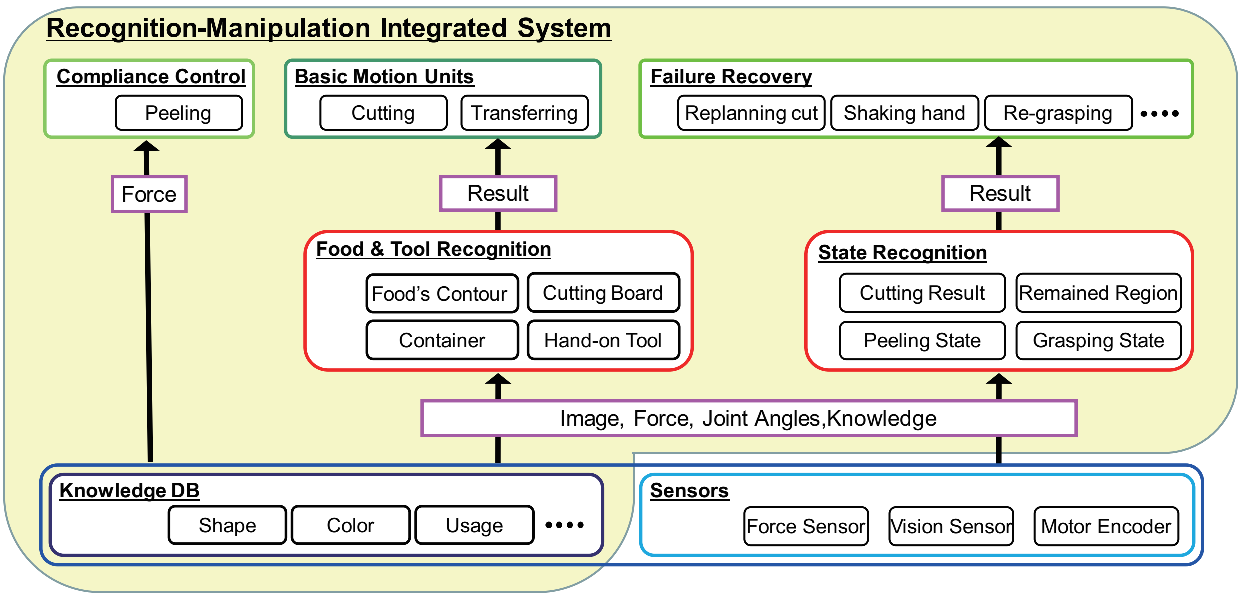

Figure 2. A cooking system integrating recognition and motion generation

erence edge and calculate the differences of them. That is,

$$D_{edge}(E^{2D}, E^{2D}_{ref}) = \frac{\sum_{N_d} value(e^m_{ref}, e^m)}{N_d}, \tag{2}$$

where $value(e_1, e_2)$ is a function that calculates correspondence between e_1 and e_2 from the difference of distance and angle. N_d is the number of pairs that have a certain level of similarity. If some reference edges could not correspond to any of input edges because of an input image having high reflected illumination, above method makes it possible to detect a target object by using other edges .

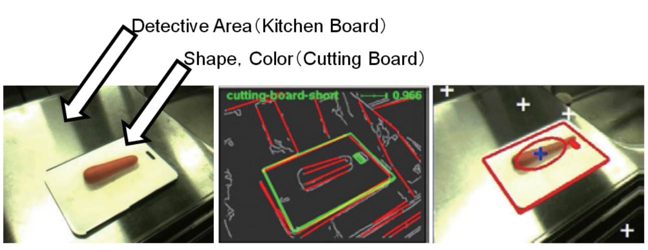

Figure 3. A cutting board and vegetable recognition based on the shape model

3.2.2. Vegetable detection based on color information

The purpose of a module to recognize target vegetables is to calculate the 3-D position and the length of the vegetable on cutting board. Because vegetables have variety of shapes and they are deformed while cooking, it is difficult to recognize them based on their shapes.

In general, cutting boards are painted with a single color. Therefore color information would be useful to detect vegetables on cutting boards. Image-based recognition intending to food have used color information [20] [21]. The main reasons are, (i) many vegetables have simple color, and (ii) because foods are processed (cutting, peeling, etc.), their shape will significantly be changed. We also apply color-based method.

The details of the module to recognize vegetables are as follows. The color of the cutting board is given in advance as a task knowledge. First, the areas with the different color from the color of cutting board are extracted. HSI color space is used in this processing. Second, the size of each extracted area is evaluated so that only the area whose size is appropriate for a vegetable is selected. The criterion for the size evaluation is as follows:

$$S_{threshold} < s < S_{cutting\_board}, \tag{3}$$

where s is the size of one of regions found on the cutting board. $S_{cutting\_board}$ denotes the area of the cutting board, and $S_{threshold}$ denotes the lower limit that is empirically defined. $S_{threshold}$ is needed to remove noise or too small vegetable pieces. After the filtering based on above equation, remaining regions are regarded to vegetable pieces. Finally, the 3-D positions of the regions are estimated using stereo calculation. Besides, the length of the area is estimated by means of PCA [22]. The right figure of Fig. 3 shows the result of this processing.

3.2.3. Recognition of containers using their circle edges

This section describes a module to recognize a container, which is used in cooking tasks. Many of containers, for example dishes and bowls, have circular edges on their top. Therefore we design a module to detect containers using their circle edges. Not only the containers but also pans and pots have circle edges. This indicates that our module could be also used to detect them.

Fig. 4 shows a result of detecting containers. The inputs of this module are, an image captured from a camera mounted on a robot, the coordinate system and the intrinsic parameters of the camera, the height of a kitchen board, and the shape of the target container which includes its radius and height.
The procedure of the recognition is as follows. First, the input image is transformed so that both of the vertical axis and the horizontal axis of the image become parallel to the plane which passing through the top edge of the container. This transformation is achieved by perspective transformation procedure, which requires the height of the kitchen table and that of a container, and the intrinsic parameters of the camera. Next, circle detection by means of Hough transformation [23] is applied to an image that is the result of canny edge detection [24] to the image after perspective transformation. Red circles shown in Fig. 4, (4) are a result of the transformation, and they indicate candidates of container. Because Hough transformation can find circles from disjunct edges, it is suitable for the kitchen environments where a part of the container can be occluded or be in too shiny as is referred on the first paragraph on section 3.2.
On the other hand, the consistency of the circles should be checked because they may include wrong candidates that cause of background edges. Another edge operator that is robust to disjunction of edges is applied to the evaluation. The operator is gabor filter [25], whose equation is as follows:

$$F(\mathrm{x}) = \frac{1}{\sqrt{2\pi}\sigma_x\sigma_y} e^{a} \cos(2\pi f x_\theta + p), \tag{4}$$

where

$$\begin{aligned} a &= -\frac{1}{2}\left(\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right), \\ x_\theta &= \quad (x-u_x)\cos\theta + (y-u_y)\sin\theta, \\ y_\theta &= -(x-u_x)\sin\theta + (y-u_y)\cos\theta. \end{aligned} \tag{5}$$

x and y denote the position of the pixel before the filtering, and u_x and u_y are the center point of gauss distribution. f denotes the frequency property of the filter. σ_x and σ_y are covariance value, and θ indicates filter direction. p indicates a phase, in this paper it is set to $\pi/2$ for the purpose of edge detection. Fig. 4, (2) is a summation of a set of filtering results that uses different θ values.
Finally, detected circles by means of Hough transformation are evaluated. Logarithmic polar coordinate conversion [26] is applied to the gabor filtering result. The center position in the conversion is the center of each circle. Fig. 4, (3) shows an example. The horizontal of the image indicates a radius direction and the vertical does angular direction. Proceeding from right to left in the horizontal axis indicates that the radius of the circle is increased. If a determinate vertical line is found in this image, the candidate circle is regarded as the top of a container. To confirm it, a vertical line is slid from leftmost side of the image, and the sum of the brightness of pixels that is on the line is calculated. If there is a line whose sum of brightness is greater than pre-defined threshold, and its position is near to radius calculated from hough transform, the circle is regarded as the edge of an actual container.
Purple line in Fig. 4, (4) shows selected circles.

3.3. Motion generation for a cooking task

This section describes modules that are commonly used for motion generation during cooking tasks.

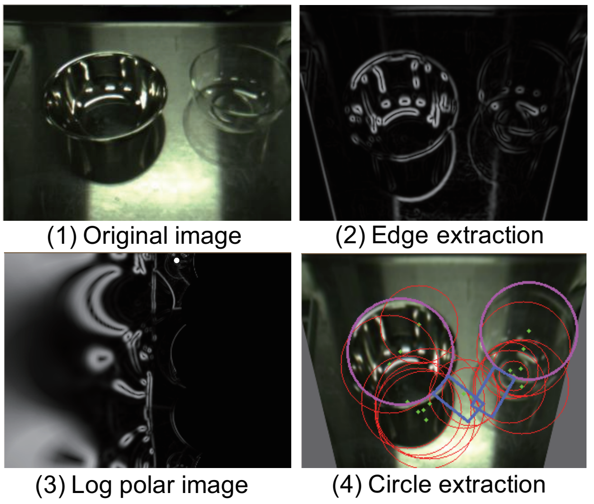

Figure 4. Round container recognition with Hough transformation

3.3.1. Method to calculate the posture of a humanoid robot

In general, to utilize a daily tool, a robot has to move one arm to operate the tool while it has to use the other arm for another important role (e.g. to fix a target object). Moreover, it is important for the robot to watch a specific point (e.g. the position of a target food and the tip of the tool) at each motion. Therefore the method to calculate the posture of the robot should take into account above two points.
The equation for posture calculation is as follows:

$$\dot{\mathbf{q}} = \mathbf{J}_w^{\#}\dot{\mathbf{x}} + \lambda(\mathbf{I} - \mathbf{J}_w^{\#}\mathbf{J}^{\#t})\mathbf{y}, \tag{6}$$

where $\dot{\mathbf{q}}$ indicates the velocity and the angular velocity of an end-effector. $\mathbf{J}_w$ is a product of a jacobian $\mathbf{J}$ and a weight matrix $\mathbf{W}$. $\mathbf{J}^{\#}$ is SR-Inverse [27] calculated using jacobian $\mathbf{J}$ as follows:

$$\mathbf{J}^{\#} = \mathbf{J}^t(k\mathbf{I} + \mathbf{J}\mathbf{J}^t)^{-1}, \tag{7}$$

where k is a constant that is for adjusting sensitivity to singular pose. $\mathbf{I}$ is an unit matrix.
$\mathbf{y}$ in Eq. (6) is an evaluation function which contributes to avoid the collision between the links of the robot, and to use the middle range of each of the joints. $\mathbf{y}$ is defined by a method proposed in [28],

3.3.2. Motion generation giving importance to the movements of the torso

Joint angles of the torso are an important factor to design the posture of the robot, because they significantly effect on the positions of both of end-effectors. For instance, let us suppose that a robot is manipulating a tool like a knife while it is holding down a vegetable on a cutting board. If the robot moves the torso to optimize only the movable range of the end-effector with a knife, it would be difficult to keep the position

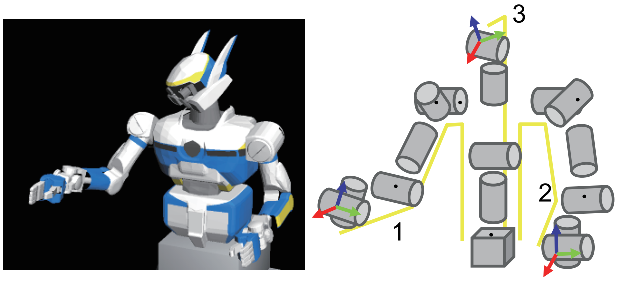

Figure 5. Kinematics link model of a humanoid robot

of the opposite end-effector, which is grasping a vegetable, because of the limitation of the joint angles. In addition, there is a possibility that a field of view of the camera on the robot cannot capture target objects. Therefore some kind of methodology to select the arm to be given priority, and to decide the joint angles of the torso in calculating inverse kinematics, is needed.

One of the solution is to use appropriate weight matrix $\mathbf{W}$ shown in section 3.3.1. In the case of the robot used in this research, two torso joints (yaw and pitch) are the root of other sixteen joints. That is, the angles of the two joints significantly effect whole body motion. Setting small weight values to the two joints at $\mathbf{W}$ is effective against preventing large pose changes of whole body. In this research, the weight of the yaw and pitch joints of the torso are set as 0.3 and 0.2, respectively. On the other hand, the weights of the other joints are set to 1.0. These values were empirically defined through a set of simulation that achieves small torso motion. It also enables the robot to stably continue to observe vegetable pieces by cameras mounted on a robot head. Both before and after a grasping motion, the cameras had almost no motion because of small change of torso joints.

As another problem, when the target pose of one end-effector is set near to robot body, it is difficult to generate robot poses without collision. This situation is frequently found on the case when the second calculation of inverse kinematics for a remaining arm. In other words, after inverse kinematics for the other arm was already calculated. To overcome such situation, feasible angle search is performed. That is, torso joints are slightly moved and inverse kinematics is calculated again until joint angles for both arms are found.

3.3.3. Manipulation of a peeler using force feedback

The main problem using a peeler is that vegetables have uneven surfaces although a robot has to move a peeler along the surfaces of the vegetables while it is peeling them.

Our approach to move an end-effector with a peeler along the uneven surfaces is based on impedance control. The target coordinate system of the end-effector of the robot, $\mathbf{x}_d$, with an impedance controller [29] is as follows:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}(\mathbf{x} - \mathbf{x}_d) = \mathbf{P}(\mathbf{f}^{act} - \mathbf{f}^{ref}). \tag{8}$$

Let $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ be six-dimensional matrices, which show virtual inertia, virtual viscous, and virtual stiffness, respectively. All of them are positive definite diagonal matrices. $\mathbf{P}$, a six-dimensional projection matrix, controls the effects of the external force and momentum on the movements of the end-effector. $\mathbf{x}$ is the coordinate system of the end-effector of the robot. $\mathbf{f}^{act}$ and $\mathbf{f}^{ref}$ represent the measurement and target value of three-dimensional force and three-dimensional momentum vector.

$$\begin{aligned}
&\mathbf{x}(t) \\
&= [\mathbf{M} + dt\mathbf{D} + dt^2\mathbf{K}]^{-1} \\
&\quad [\mathbf{M}[2\mathbf{x}(t-dt) - \mathbf{x}(t-2dt)] \\
&\quad +dt\mathbf{D}\mathbf{x}(t-dt) + dt\mathbf{D}\mathbf{x}\{\mathbf{x}_d - \mathbf{x}_d(t-dt)\}] \\
&\quad +[\mathbf{M} + dt\mathbf{D} + dt^2\mathbf{K}]^{-1}[dt^2\mathbf{K}\mathbf{x}_d(t) + dt^2\mathbf{P}(\mathbf{f}^{act} - \mathbf{f}^{ref})]
\end{aligned} \tag{9}$$

4. Experiments

This section describes experiments of cooking behaviors by a robot. Experiments about three basic cooking behaviors (cutting, peeling, and transferring) whose targets were vegetables are reported. After that, making a salad that includes several cooking behaviors is introduced. At the beginning of each behavior, a target vegetable was placed on a cutting board, and the type of behavior was given by manual. Fig. 6 shows a list of task knowledge used for each task.

4.1. Cutting vegetables

There are various types of knife trajectories according to processing purpose. In this experiment, a trajectory that moves a knife vertically up and down for cutting firm vegetable was selected. Matters for defining the trajectory are the operational area of the knife and the pose of the cutting edge. That is, variables for defining the trajectory can be summerized by following three parameters:

1. the amount of movement in front-back direction of the cutting edge (Fig.7, x),
2. the maximum height of the knife (Fig.7, y),
3. the rotation angle of the knife (Fig.7, θ).

Of these, the knife angle θ was set to 0. y was automatically detected by the function of a cutting board detection explained in section 3.2.2. x was manually given.

In the calculation of the pose of grasping a vegetable, the weight value about a torso joint in a weight matrix was set according to the consideration described in section 3.3.2. That is, because much rotation angle of the torso for the grasping leads to a whole body pose that was difficult to reach a knife to a target vegetable, the torso link was limited to small movement.

At the beginning of the cutting, a robot detected a vertical distance between the knife and the cutting board. They were done by using a force sensor mounted on the wrist of the robot while moving the knife down. When the load on the wrist exceeded the predefined threshold that was given as a task knowledge, the robot judged that the target object was there and calculated the vertical distance from a floor surface. These results were used to initialize visual recognitions explained at section 3.2.2 that were applied after this procedure.

One result is shown in Fig.8. Left 2 columns in the figure are obtained by visualizing the robot states during the experiment. Arrows drawn on the wrist show the load sensed by force sensors. Right 2 columns in the figure are the motion of the real robot and the result of image processing. In this experiment, after recognizing and grasping a cucumber

	Task Knowledge	Note
Common	Kitchen information	The height of kitchen board and its shape
	Cutting board information	3D shape, color and search area for recognition. Grasping positions for manipulation.
Cutting	Knife information	A rough 3D shape of knife.
	Parameters for cutting	Positions and angles for making trajectory(shown in Fig.7)
	Amount of load change on a wrist	Threshold value to detect the edge of vegetable and to judge whether or not a cutting success.
Peeling	Peeler information	A rough 3D shape of peeler
	Amount of load change on the wrist	A threshold value to detect the surface of a target vegetable.
	Amount of visual change of the target vegetable	A threshold value for the result of image subtraction between before and after peeling. To judge wheter or not the target has still skin.
Transfering	Bowl information	Search area for recognition.Grasping positions for manipulation.
	Amount of load change on the wrist	A threshold value for detect the height of the knife.
	Parameters for image edge detection	Threshold values included in a visual function to detect the position of the tip of the knife.

Figure 6. Task Knowledge which are given in each experiments

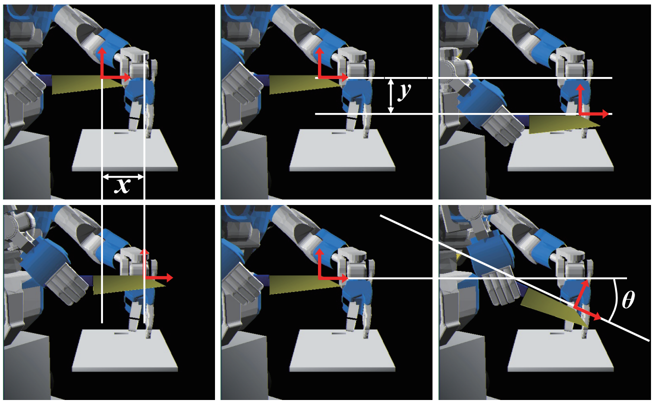

Figure 7. Parameters for cutting by knife

on a cutting board, the robot cut the cucumber into eight pieces of 5 [mm] width in 9 [sec].

The same method was applied to the case of carrot. Because carrot was generally firmer and thicker than cucumber, same parameters could not be used. For preventing a failure that the carrot could not be cut despite of getting an estimation of the cutting success, failure detection and recovery motion were inserted to our system shown in Fig. 2. To estimate the condition of the cutting, the robot moved the knife upward a little just after one cutting was considered success. After that, a small amount of motion that was an opposite direction to the hand which grasped the carrot was given to the other hand which grasped the knife. Because the load to the wrist became greater in the case of failure, the results was used to judge cutting success. That is, the threshold value to the force sensor attached on the wrist was given as a task knowledge. In the failure case, cutting motion against to same position to the vegetable was called again. This enables the robot to achieve cutting operation independent to vegetable's stiffness and shape.

Figure 8. Cutting vegetable with a knife and its result

4.2. Peeling vegetables

Experiments about peeling vegetables by using a peeler were performed. First the robot detected and grasped a sweet potato with the

same way as the cutting experiment, and put the potato on the center of a cutting board. After the direction estimation of the potato by approximating its image region as an ellipse, the robot moved the peeler close to the potato from upper side. The peeler was moved until the amount of change of load to the wrist was over a pre-defined threshold, and the maximum value of the force sensor was recorded. The value was used as parameters of an impedance control for peeling operation. It was essential to give a task knowledge that was for stopping to push a peeler on a vegetable. For instance, because a cucumber and a sweet potato have different stiffness, the robot might break them if a constant threshold regardless of the target vegetable was used. From this reason, a set of task knowledge with respect to each vegetable was given as the amount of the change of load value to the wrist. Although these knowledge were defined by manual, parameters for impedance control that was for peeling were automatically selected on the spot.
In the peeling, the robot removed skin of the potato by moving the peeler with fixing its angle toward longitudinal direction of the potato. After that, the robot judged whether or not the potato had still skin by means of image subtraction between before and after the peeling. The subtraction result was evaluated by using a threshold given as a task knowledge. If the peeling had not already been done, the peeling was continued after the potato was rotated.
Because a two-fingered gripper we used lacked the dexterity to rotating a vegetables in it, the rotating was achieved by rolling over the vegetable on the cutting board. Fig.9 shows one of the results in the case of a sweet potato.

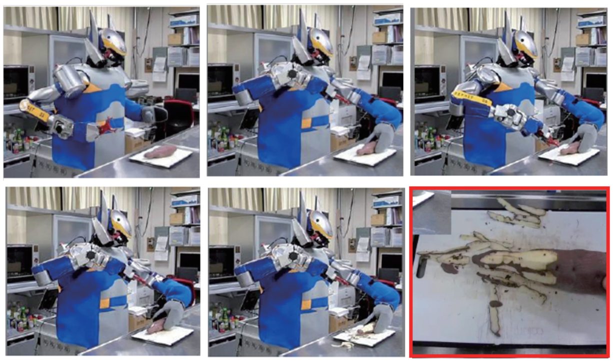

Figure 9. Peeing action and its result

4.3. Transferring operation experiment

Experiments about transferring processed vegetables from a cutting board to a bowl were performed. First, detection and pose recognition of the cutting board and the bowl were performed based on task knowledge that were search area, rough shape, how to grasp, and visual information. The judgement whether or not the processed vegetables remaining on the cutting board after one transferring action was achieved by the same method as vegetable recognition in cutting and peeling experiments.
In the case that the motion of a hand having a knife was just only considered in the transferring, there was a problem that the robot could not generate a motion because of the limitation of the range of movement. Therefore, as shown in Fig. 10, both trajectories of hands having the knife and the cutting board were simultaneously considered. The target trajectories for the motion were given as a pair of a center position of the cutting board and an edge point of the knife. Especially the former was important. If the robot moved only the knife, the trajectory of the hand having the knife was likely to include singularity because target point was near robot body. On the other hand, adding the trajectory of the cutting board enabled the robot to ensure broader solution space. Fig. 11 (7) and (8) that is a part of making a salad show an example of transferring. In many cases, after two or three times of sliding a knife on the board, all of pieces were transferred to the bowl.

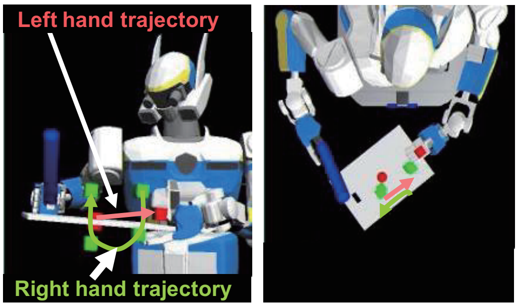

Figure 10. Manipulation with both arms

4.4. Making a Salad by combination with basic cooking operations

An experiment of making a salad is introduced. The procedure of this experiment was as follows:

- Cut a cabbage laid on a board,
- Transfer the cut pieces into a bowl,
- Dress the pieces.

The robot detected and grasped a cabbage by left hand for holding it on a cutting board, and cut the cabbage into strips. Then, after detecting a bowl, the robot attracted it, and transferred the cut pieces of the cabbage to the bowl. Finally, dressing in a bottle was poured to the pieces for completing the making a salad.
In this sequence, actions that human supported to the robot were just to grasp a knife and to hold the dressing. Although the shape of the knife was not given in advance, the robot recognized it by visual and force sensing. That is, width of the edge and the position of the tip from a robot hand were automatically modeled. Fig.11 shows the sequence of the experiment. The robot could achieve such compositive task because it recognized manipulation targets and around situations with using task knowledge and sensor data, with appropriate dual-arm manipulation.

5. Conclusions

In this paper we described a system integration for a life-sized robot working at a kitchen. On a cooking tasks, there should be various tools and foods, and a kitchen table may have reflective surface with blots and scratch. Recognition functions should be robust to noises derived

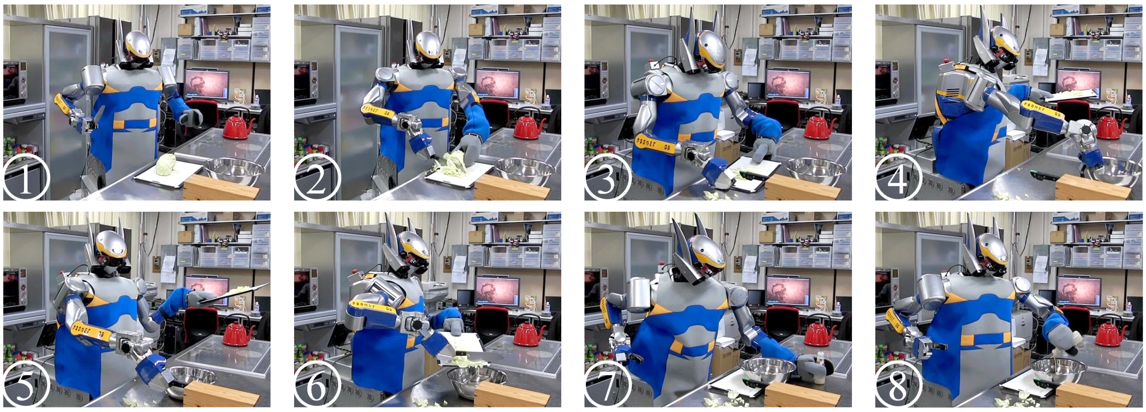

Figure 11. Making a salad

from them. As other problems, cooking behaviors impose motion sequences by using whole body of the robot. For instance, while cutting a vegetable, the robot has to hold one hand against the vegetable even if another hand with a knife should be moved for the cutting. This motion requires to consider full articulation of the robot simultaneously.
We attacked difficulties mentioned above. For instance, recognition functions for a cutting board and containers were implemented. These are extensions of existing methods. By using not only shape model but also the combination with task knowledge, the improvement of robustness against the influence of reflective surface was achieved with keeping universality. Combining these functions with color-based vegetable detection, we made it possible to recognize several cooking situations. On the other hand, we pointed out the importance of the use of torso joints while dual arm manipulation. Our approach enabled the robot to keep manipulability of both arms and viewing field of a head. Based on the products, we introduced an integrated system incorporating recognition modules and motion generation modules.
Due to the system, our robot could manipulate tools ingeniously and achieved three basic operations; cutting, peeling and transferring. Furthermore, we succeeded to make a salad by the robot. These results show that our system is one of feasible approaches to achieve cooking tasks that require to manipulate non-uniform target with handling cooking tools.
From the viewpoint of ease of implementation, we showed methods for cooking behavior as follows:

- Cutting operation using a knife:

  We defined knife manipulation as combination of basic trajectory and three parameters for position and direction. This enabled a robot to cut various sized vegetables.

- Peeling operation using a peeler:

  We set a basic trajectory that was parallel to the longitude of a target vegetable, and a robot made a peeling by compliance control considering pressing force to the vegetable. As a result, the robot completed peeling action with adopting to vegetable's uneven surface. Pushing force could be automatically determined by the robot just before the peeling.

- Dual-arm cooperation at transferring operation:

  We succeeded to make transferring operation by a dual arm robot. Even if many vegetables are on a cutting board, and the position of the cutting board and a bowl are changed, our approach enables to achieve the transferring.

For these reasons, it is possible to design cooking behavior with reducing programming complexity.
For future work, there are issues in respect to both recognition and manipulation. In terms of recognition, it is needed to enhance the versatility of object recognition under various situations. To complete complex cooking sequence, it is required to choose cooking ingredient and its container on a case-by-case basis. In addition, to avoid recognition failure is also important. In terms of manipulation, it is important to plan more feasible manipulation to achieve a task under more difficult situations. For instance, to make an adaptable peeling motion adjusting to the surface of uneven cooking ingredient, and to generate a dual arm motion for cutting cooking ingredients on a cutting board.
Moreover, to whisk cooking ingredients in a container and to clean up blot are very interesting tasks. In dual-arm manipulation, we picked up transferring operation that was an example of manipulation that uses two tools with monotonous trajectories, but frying and other cooking actions need to move respective arms in more complex ways. If we can define them by formulated functions and add them into our system, it enables a robot to complete various cooking actions. We will get various effective acknowledge for daily life support.

References

[1] iRobot Corp. roomba, http://www.irobot.com/us/robots/home/roomba.aspx.
[2] R. Hamada, J. Okabe, I. Ide, S. Satoh, S. Sakai, H. Tanaka, Cooking Navi: Assistant for Daily Cooking in Kitchen. In Proceedings of the 13th annual ACM international conference on Multimedia, pages 371–374, Singapore, 2005.
[3] W. Ju, R. Hurwitz, T. Judd, B. Lee, CounterActive:An Interactive Cookbook for the Kitchen Counter. In CHI '01 Extended Abstracts on Human Factors in Computing Systems, pages 269–270, USA, 2001.

[4] L. Bonanni, C.H. Lee, T. Selker, CounterIntelligence:Augmented Reality Kitchen. In CHI '05 Extended Abstracts of Computer Human Interaction(CHI), pages 2239–2245. ACM Press, USA, 2005.

[5] I. Siio, N. Mima, I. Frank, T. Ono, H. Weintraub, Making Recipes in the Kitchen of the Future. In CHI '04 Extended Abstracts on Human Factors in Computing Systems, pages 22–29, Austria, 2004.

[6] T. Fukuda, Y. Nakauchi, K. Noguchi, T. Matsubara, Human Behavior Recognition for Cooking Support Robot. In Proceedings of 2004 IEEE Intl. Workshop on Robot and Human Interactive Communication(ROMAN), pages 359–364, Japan, 2004.

[7] M. Bollini, S. Tellex, T. Thompson, N. Roy, D. Rus, Interpreting and Executing Recipes with a Cooking Robot. In Proceedings of International Symposium on Experimental Robotics(ISER), 2012.

[8] S.S. Srinivasa et al., HERB: A Home Exploring Robotic Butler. In Journal of Autonomous Robots, Vol.28, No.1, pages 5–20, 2009.

[9] Motoman industrial robot cooks okonomiyaki. Available at http://www.danshope.com/news/showarticle.php?article_id=43.

[10] Fa-men: robot cookin'. Available at http://www.rameniac.com/reviews/comments/fa-men_nagoya/

[11] K. Kosuge, Development of an automatic dishwashing robot system. In Proceedings of IEEE International Conference on Mechatronics and Automation(ICMA), pages 43–48, China, 2009.

[12] Y. Sugiura, D. Sakamoto, A. Withana, M. Inami, T. Igarashi, Cooking with Robots: Designing a Household System Working in Open Environments. In Proceedings of International Conference On Human Factors In Computing Systems(CHI), pages 2427–2430, USA, 2010.

[13] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp and R. Dillman, Armar-III: An integrated humanoid platform for sensory-motor control. In Proc. of IEEE-RAS Int. Conf. On Humanoid Robots (Humanoids 2006), pages 169–175, Italy, 2006.

[14] Ch. Borst et al., Rollin' Justin - Mobile Platform with Variable Base. In Proc. of the IEEE Int. Conference on Robotics and Automation(ICRA), pages 1597–1598, Japan, 2009.

[15] B. Bäuml et al., Catching flying balls and preparing coffee: Humanoid Rollin'Justin performs dynamic and sensitive tasks. In Proc. IEEE International Conference on Robotics and Automation(ICRA), pages 3443–3444, China, 2011.

[16] H. Iwata and S. Sugano, Design of Human Symbiotic Robot TWENDY-ONE. In Proc. of IEEE International Conference on Robotics and Automation(ICRA), pages 580–586, Japan, 2009.

[17] K. Okada, M. Kojima, S. Tokutsu, T. Maki, Y. Mori and M. Inaba, Multi-cue 3D Object Recognition in Knowledge-based Vision-guided Humanoid Robot System. In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), pages 3217–3222, USA, 2007.

[18] D. Pangercic, V. Haltakov and M. Beetz, Fast and Robust Object Detection in Household Environments Using Vocabulary Trees with SIFT Descriptors. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception and Object Search in the Real World, USA, 2011.

[19] J. Bohg et al., Task-based Grasp Adaptation on a Humanoid Robot. In 10th International IFAC Symposium on Robot Control(SYROCO), Croatia, 2012.

[20] A. Rocha, D.C. Hauagge, J. Wainer, S. Goldenstein, Automatic fruit and vegetable classification from images. In Journal of Computers and Electronics in Agriculture, Vol.70, pages 96–104, 2010.

[21] H. Patel, R.K. Jain, M.V. Joshi, Fruit Detection using Improved Multiple Features based Algorithm. International Journal of Computer Applications, Volume 13 No.2, January, pages 1–5, 2011.

[22] M. Mudrová, A. Procházka, Principal Component Analysis in Image Processing. In Technical Computing Conference, Czech Republic, 2005.

[23] S.K. Pedersen, Circular Hough Transformation. Aalborg University, Vision, Graphic, and Interactive Systems, 2007.

[24] R. Maini, H. Aggarwal, Study and Comparison of Various Image Edge Detection Techniques. In International Journal of Image Processing(IJIP), Volume 3 Issue 1, January-February, pages 1–11, 2009.

[25] Tutorial on Gabor Filters, Available at http://mplab.ucsd.edu/tutorials/tutorials.html.

[26] V.J. Traver, A. Bernardino, A review of log-polar imaging for visual perception in robotics. In Journal of Robotics and Autonomous Systems, Volume 58 Issue 4, April, pages 378–398, 2010.

[27] Y. Nakamura and H. Hanafusa, Inverse Kinematics Solutions with Singularity Robustness for Robot Manipulator Control. In Journal of Dynamic Systems,Measurement and Control, Vol.108, pages 163–171, 1986.

[28] T.F. Chang and R.V. Dubey, A weighted least-norm solution based scheme for avoiding joint limits for redundant manipulators. In IEEE Trans on Robotics and Automation, Vol.11, No.2, pages 286–292, 1995.

[29] N. Hogan, Impedance Control: An Approach to Manipulation. In Proc. of American Control Conference, pages 304–313, 1984.