Smart Kitchen: A User Centric Cooking Support System

Atsushi HASHIMOTO† **Naoyuki MORI**† **Takuya FUNATOMI** ††

Yoko YAMAKATA ††† **Koh KAKUSHO** †† **Michihiko MINOH** ††

{a_hasimoto/mori/funatomi/kakusho/minoh}@mm.media.kyoto-u.ac.jp
yamakata@nict.go.jp

† Graduate School of Informatics, Kyoto University, Kyoto 606-8501, JAPAN
†† Academic Center for Computing and Media Studies, Kyoto University, Kyoto 606-8501, JAPAN
††† National Institute of Information and Communications Technology, Tokyo, 184-9785, JAPAN

Abstract

Several cooking support systems have been studied to give users instructions based on the recipes step-by-step, using multi-media contents. These systems usually disturb users' cooking process forcing them to provide information to the system in order to give them beneficial information. In this sense, these systems are considered to be "system centric". We propose a system called "Smart Kitchen" considered to be in "user centric", in which a user can cook normally without being concerned about the system. Smart Kitchen can understand cooking processes, in other words, what the user is doing. In this paper, we discuss the design of the Smart Kitchen system and explain three essential modules of tracking food, recognizing food material, and recognizing cooking action.

Keywords: Context-Aware Computing, Smart Environments, Intelligent Kitchen, Object Tracking, Food Material Recognition.

1 Introduction

Cooking is one of the most important activities in our daily life. With the deployment of ICT, it becomes possible and important to help humans cook with ICT. In cooking, we sometimes refer to a recipe. The recipe describes the cooking process, the way how to make a dish as a sequential order of cooking steps, each of which is specified by cooking actions such as boiling, cutting etc. and food materials to be handled. Since the recipe is described in a cook book, we need to refer it to follow the recipe during the cooking process. The cookbook includes textual representation and static images, which are sometimes too vague to convey the condition of handled food materials and the timing to do something. For example, a sentence, *"To make the custard, whisk the egg whites until stiff enough to be cut with a knife"* with a static image of stiffly whisked egg whites is an instruction written in a cookbook. It is not easy to imagine how stiff it is, from the sentence and the image precisely. A video guidance in a cookery program gives more concrete idea of the condition of the food materials. We can imagine the stiffness more precisely from appearance of the egg whites shown in the video.

Today, there are several systems [1, 2, 3] which give cooks instructions in the recipes by multimedia guidance. We call them interactive recipe systems. Interactive recipe system is to give instructions of recipe through display, voice guidance and other output devices step-by-step in the proper timing during the cooking.

These existing systems are "system centric" in the sense that the user has to follow the given instructions, and the system requires him/her to show the timing i.e. the end of the cooking steps in order to get an appropri-

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 848–854
Torremolinos (Málaga), June 22–27, 2008

ate instruction. Such system centric design is undesirable to the daily use.

With a "user centric" cooking support system, a user can cook without being concerned to the system. In this paper, we propose a "user centric" cooking support system named *Smart Kitchen*. Smart Kitchen is assumed to know the recipe the user is following before the cooking as the given knowledge. It gives an instruction that the user should do the next either the cooking steps finish or when he/she asks. The user can change the cooking steps freely. Under this user centric concept, we designed Smart Kitchen and developed three essential modules of tracking food, recognizing food material, and recognizing cooking actions.

This paper is organized as follows. In section 2, discusses the user centric concept in more detail. Section 3 then overviews the system and Section 4, 5 and 6 go into details of the three main modules. Finally, Section 7 describes conclusion and future problems.

2 A user centric cooking support system

We are designing Smart Kitchen as a user centric support system. There are two points to be discussed in this section. One is the order of cooking steps and the other is how to notify the system the end of the cooking step.

Observing daily cooking processes makes us notice that a cook does not always cook in the same way. Particularly, the order of cooking steps is different from that of the steps written in a recipe. This is because the order of cooking steps defined in the recipe is intrinsically in the partial order with respect to each food material. In other words, the cooking steps for the same food material are in the total order and cannot be changed on site.

Due to the nature of partial order of the cooking steps described in the recipe, there are two design policies for cooking support system. One is, what we call, "system centric" design in which the partial order of the cooking steps is decided by the supporting system as the previous works did. The other is "user centric" in which a user can decide the order for his/her own sake. In this case, when a user is novice to the cook and has no idea what to do next, the system detects the situation and shows one of such cooking steps as guidance.

In order to show a cooking step a user can perform next, the system has to know the end of the previous cooking steps in real time. To know the end of the cooking steps, there are three strategies; user notification [2, 3], use of electronic labels, and recognition of the cooking action and food material in the cooking steps.

With the strategy of using notification, a user notifies the end of the cooking step to the system by pushing a kind of switches. Then, the system selects one of the instructions for the next possible cooking steps. This strategy works well only under the assumption that the user follows the given instructions faithfully.

The strategy to label objects with tags is effective. To recognize food materials and instruments in the kitchen, it is possible to attach RFID tags to them. The reader of the RFID tags is mounted on the worktop and in the refrigerator. Food materials and the instruments are recognized by reading tags, which results in recognizing the cooking step. However, it is annoying to attach the tags to all the cooking instruments in the home kitchen, only to sense the end of the cooking step.

The strategy for the system to recognize the end of the cooking step forces no additional tasks to the users. The cooking step consists of food materials and cooking action, so the recognition of the cooking step has to recognize both food materials and cooking actions. It is very difficult to recognize food material because of the variety of color and shape of food materials, which are the key features for image recognition. It is also very difficult to recognize the cooking actions, because of the complexity of the action itself. (For example, cutting, washing and peeling are cooking actions.) Therefore, the recognition errors are inevitable, but the system copes with these errors with the collaboration of the users.

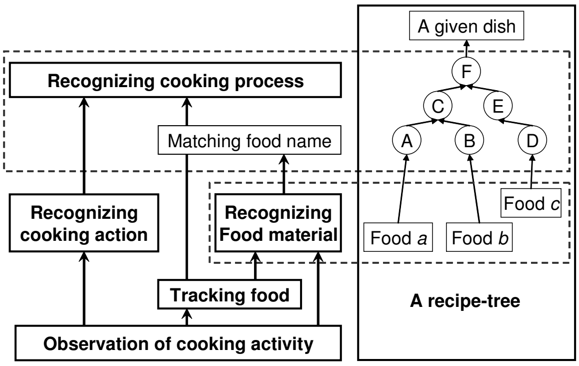

Figure 1: Overview of Smart Kitchen

Based on the discussion above, the system centric strategy requests the user notification, because it forms interaction between user and the system. On the other hand, the user centric system request the recognition, either pattern recognition or RFID tags. The user centric system observes the cooking process silently and support the user whenever necessary.

Considering that the kitchen is used daily, except that the user may try to challenge a new cook following the recipe, he/she usually makes dishes that have already tried several times. The user has his/her order of the cooking steps, so the system centric strategy is not suitable as the design strategy. In addition, the user possibly skips or forgets the notification process and goes on the cooking. Therefore we select to use the recognition techniques to know finished cooking steps.

3 Smart Kitchen system

3.1 Overview

Based on the discussion above, we have designed Smart Kitchen system. The logical structure is shown in Figure 1. At first, the system observes the cooking process in the kitchen with sensor devices: it has three optical cameras and one thermal camera which are installed over the tabletop (see Figure 2). The optical cameras observe food materials and user's hands. The thermal camera captured heating condition in stove area.

Second, the recipe is represented as recipe tree

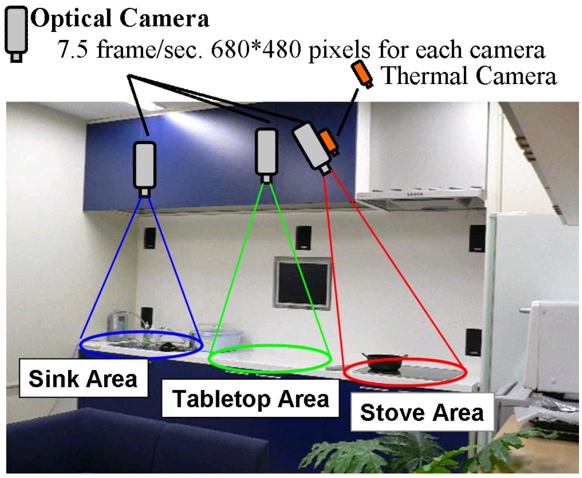

Figure 2: The Smart Kitchen environment

which is suitable to represent the partially-ordered nature of recipe mentioned in Section 2. Each leaf node corresponds to a food material, and the root node to a product of the recipe. Internal tree nodes (A to F in the recipe-tree shown in Figure 1) correspond to cooking steps in the recipe. The cooking step node that all the children have already done becomes a candidate of the next cooking step of the cooking process. These nodes have multimedia guidance contents for presenting timely.

Third, to recognize the cooking step, both the cooking actions and the food materials handled in the step have to be recognized. In the cooking action, one or more food materials are handled. The end of the cooking step is detected by recognizing the final cooking action in the step. The cooking actions are recognized by the position of the kitchen where the user is working.

To recognize food materials handled in a cooking step, we use two techniques of recognizing food material through a part of cooking process, and tracking food on tabletop. In this paper, we use a word "food" to point a food material or food materials which are merged through some cooking actions.

Recognizing food material from an image is a difficult task because of two reasons. The first reason is that naturally-derived food materials have no uniform shape or texture of the skin, dislike manufactured one. The second reason is that several pigments dominate colors of food materials and then different food materials often have similar color. To recognize food material, we have developed a method using food preparing steps. The food preparing steps are always performed to the food not depending on recipes. Normally, those who do cooking everyday know well the preparing steps for each food so it is not necessary to be instructed. Thus, we use these preparing steps to recognize food material.

We have also developed food tracking module to recognize the cooking step. Cooking action often changes appearance of food material. For example, cutting action changes a food into uniform shape, which makes the shape change. Smart Kitchen system tracks a food, and the ID numbers are assigned to the same food tracked. In early stage of cooking process, this ID number is linked to the food materials recognized in the preparing steps.

This module is also used to recognize a cooking action *mix*. The *mix* action is featured by geometric relation between two or more foods rather than user's behavior. Thus we treat *mix* recognition in the tracking module.

In the following three sections, the recognition method of cooking actions and food materials, and the tracking method are explained in detail.

4 Recognizing cooking action

The system recognizes the end of cooking step by detecting the last cooking action in the step. A cooking action is normally expressed by cooking terms with which it is difficult for the system to recognize the cooking actions in the same level described. This is because there are many similar actions which have different names. Hence, we defined a roughly divided categorization of cooking actions, which the system can recognize robustly (e.g. action class *cut* includes "cut finely", "cut coarsely", "peel" etc.).

At first, we investigated the cooking terms in recipes. We collected 1420 recipes (written in Japanese) from WWW and analyzed 424 terms related to cooking actions (15737 instances). As a result, most of them were categorized into *cut/peel*, *heat*, or *mix* . Dividing heat category into *s-fry*, *d-fry* and *boil* (see Table 1), we get five categories which the system can recognize robustly.

Category	Some instances	Frequency (of top 30)
cut/peel	cut, peel, finely cut	19.5%
s-fry	stir-fry, sauté	9.3%
d-fry	deep-fry	1.4%
boil	boil, stew	9.9%
mix	mix, add, blend	31.9%
others	heat, melt, pour dish, cool divide, season	28.0%

Table 1: 5 categories of cooking action and frequency in top 30 frequent words

Table 2: The relationship between user location and the cooking actions

user location	cooking action category
table-top	cut/peel
stove	heat

These categories (*cut/peel*, *s-fry*, *d-fry*, *boil* and *mix*) covered 79.2% of top 30 frequent terms of cooking actions in total.

The system recognizes these categories as follows. All the categories except *mix* are divided into two groups according to the user location as shown in Table 2: If the user is in tabletop area, the user will *cut/peel* something. Similarly, the user will *heat* something if he/she is in stove area.

In *heat* action, the system distinguishes *d-fry* by monitoring a switch which is special for deep-frying. Such a switch is popular in Japanese kitchen units of induction heaters. The other two categories, *s-fry* and *boil*, are recognized from the thermal camera image. We set threshold as 120 degree on the Celsius scale. If the temperature is higher than the threshold, it will be *s-fry*. These rules make a decision tree shown in Figure 3. We describe

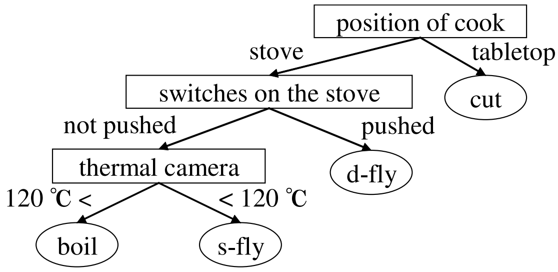

Figure 3: the decision tree to recognize cooking actions except *mix*.

how to recognize *mix* action in Section 5.

We observed 65 cooking actions as test samples. In this experiment, we regard the user performs a cooking action iff the user stay in same area longer than five seconds. As a result, we achieved 84.6% accuracy of cooking action recognition in the test samples. There were one miss-recognition case, six false-positive and six false-negative cases of the action detection.

5 Tracking food on cooking

Food tracking has characteristic difficulties, which are not considered in most of existing tracking algorithms. There are some cooking actions which change the appearance of foods. Especially, cut and peel action changes food appearance drastically. Furthermore, user's hand often hides the food during the cooking action, and then the system cannot observe the change in appearance. In short, a cooking action may change food appearance discontinuously. Traditional image-feature-based algorithms (e.g. particle filter [4] and mean-shift [5]) are not available in such a case. Thus we developed a different algorithm which does not depend on image features, to match the discontinuously changed food.

The foods are divided into two groups during a cooking action, handled group and not-handled group. The foods which are not handled during the action do not change in appearance and traditional algorithms are available to them. Only the handled foods changes in appearance discontinuously during a cooking action. To detect the handled foods, we focused on the input-output relation of a cooking action. Usually, a user cannot perform two different cooking actions with his/her hands at the same time and A cooking action always products only one kind of food, in which all handled foods are merged. Thus, the all ID numbers of the foods disappeared at the start of the action must be assigned to the food reappeared at the end of the action. Then the ID numbers of input foods are merged into a set of ID numbers and added to the output food. The output food may have one or more ID numbers derived from its component. The action includes *mix* action when two or more different foods are input.

A cooking action is done only when the user grasps one or more foods. Therefore we regard a grasp as a cooking action in this module. The grasp recognition also detects which food is input to the cooking action.

We recognize grasp as shown in Figure 4. With this camera setting, only user's hands come into the workspace beyond the edge of the worktop. Thus, we can easily identify the user's region. In the left image, the user has no foods. The user extends hand between the left and middle images, and occludes the region of tomatoes and carrot. The system registers all such occluded regions to candidate list of grasped objects. Then, in the right image, the system detects that the tomatoes are remained as they were and the carrot has disappeared, using change in values of image features. The system deletes the tomatoes from the candidate list, and regard the carrot has grasped. The end of the grasp action and the output foods are detected by applying input food detection process to the reverse order. If you see Figure 4 from right to left, a carrot is appeared and regarded as an output food in the left image.

We validated the algorithm by applying it to cooking processes in the tabletop area. The results of this experiment are shown in Figure 5. For simplicity of implementation, we placed all food materials apart on the tabletop, and give the ID numbers only to food regions. The upper row shows that the user peels and cuts two potatoes. The color has

Figure 4: Cooking action detection by touched region. The red, yellow, green and gray regions are cook, carrot, tomato and the others respectively.

changed after peeling. In the right image, the ID numbers of the potatoes (2 and 4) are succeeded to the pieces of the skin and the cut potatoes successfully. On the lower row, the object with ID number 5 is a carrot. The system detects a *mix* action because ID number 2 (or 4) and ID number 5, each of which corresponds to different food names (potato and carrot) respectively, are merged.

6 Recognizing food material

We recognize a food material by using food preparing steps. A food preparing step is derived from properties of the food material (For example, cutting a side when it has the root, peeling when it has the skin, etc.). When we categorize the preparing actions by the similar way described in Section 4, there are only four categories; *cut*, *peel*, *boil* and a new category *wash*. Here, we divided *cut* and *peel*, which are most frequent actions in the preparing steps.

We have investigated 100 recipes randomly selected from those investigated in Section 4 to ensure that the categorization is detailed enough to recognize food material. We studied brawn food material cases, which has 57 instances (33.5%) in 170 problematic instances appeared in the 100 recipes. (Brawn was the top rate color.) As a result, 47 instances (82.5%) were judged as theoretically recognizable with the preparing steps. 7 instances (12.2%) were manufactured foods, which require no preparing steps. The food preparing steps cannot distinguish manufacture foods in principle. The rest 3 instances (5.3%) show us that more detailed categorization will bring us only 5.3% improvement at the maximum.

The location of the user is not enough to recognize *cut* and *peel*. And also, user's hands often hide the area of the cooking action. Since the result of action recognition is not reliable, we developed a probability-based method to recognize food material. At first, the system calculates conditional probabilities of occurrence of each action from observed features. Then calculating the probability which supports all preparing steps has done for each food. The system regards the food with maximum probability as the most likely food material.

We observed ten preparing steps for four brawn food materials. For simplicity of implement, we gave action detecting result and tracking result by hand. We used six preparation data for each food to train the conditional probability calculation module, and four to test our algorithm. As a result, we recognized 93.8% (15/16) of accuracy.

7 Conclusion

Our goal is to realize a user centric system for supporting cooking, which we can use in daily cooking life. After discussing the general view of our system, developed three essential modules are explained in this paper. We defined five cooking actions and recognized these with 84.6% accuracy. The food tracking method achieved to track objects whose features are changed during occlusion. And the food material recognition achieved with 93.8% accuracy to brown food materials, by observing the food preparing steps.

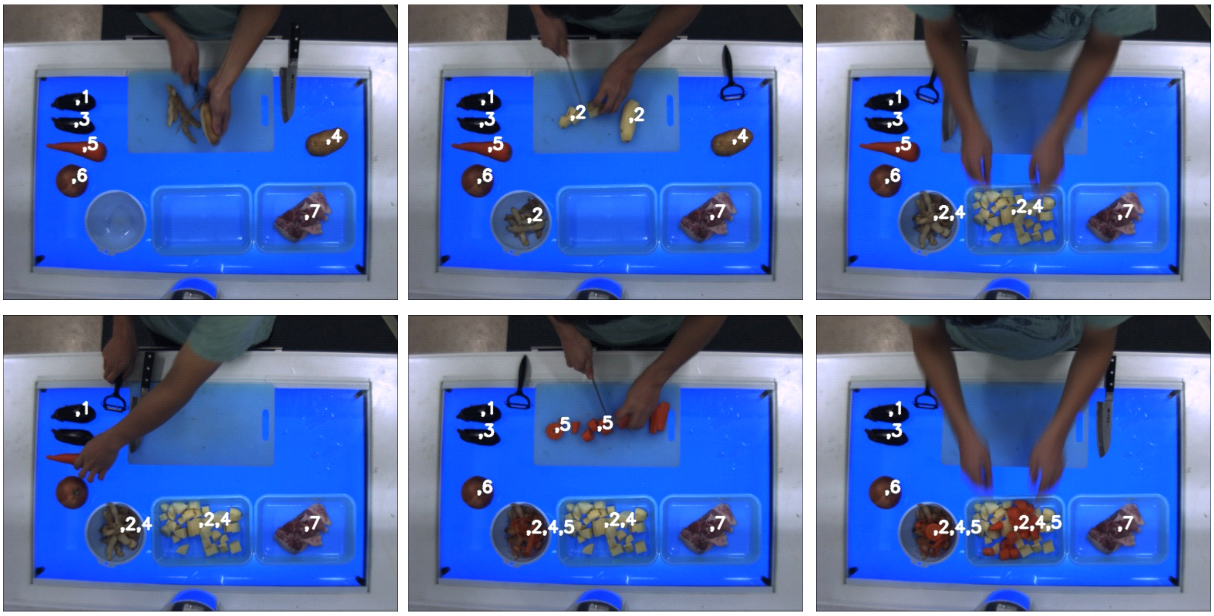

Figure 5: The tracking result of the developed algorithm. Images on upper row: working to potatoes, and lower row: to carrot

We have implemented not the entire proposed system but these three modules independently. We plan to combine the three modules, and realize the entire system of the Smart Kitchen.

References

[1] W. Ju, R. Hurwitz, T. Judd, and B. Lee. Counteractive: an interactive cookbook for the kitchen counter. *Conference on Human Factors in Computing Systems*, pages 269–270, 2001.

[2] J. S. Bradbury, J. S. Shell, and C. B. Knowles. Hands on cooking: towards an attentive kitchen. *Conference on Human Factors in Computing Systems*, pages 996–997, 2003.

[3] I. Siio, R. Hamada, and N. Mima. Kitchen of the future and applications. *Lecture Notes in Computer Science*, 4551:946–955, 2007.

[4] T. Arulampalam S. Gordon, N. Clapp and M.S. Maskell. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing,*, 50:174–188, Feb 2002.

[5] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.