THE INTERACTIVE COOKING SUPPORT SYSTEM IN MIXED REALITY ENVIRONMENT

Arata Horie, Satoru Mega and Kuniaki Uehara

Graduate School of Science and Technology, Kobe University

ABSTRACT

Recently, many learning systems, such as e-learning and WBT (Web Based Teaching) systems have been developed. In these systems, users can get educational contents and graphical material in a remote place through the network. However, the learning is always one-sided. That is, all users learn the same contents and cannot convey their states like “What is he or she doing now?” to these systems. Therefore, these systems are not necessarily satisfying all the users’ demands.

In this paper, we propose an interactive learning application of cooking in mixed reality (MR) environment. In an MR environment, a system should not only provide the useful knowledge, but also recognize users’ state. Furthermore, by making a parallel transition model about cooking, it is possible to control the user’s cooking process. Therefore, users can learn details about cooking more flexibly and effectively.

1. INTRODUCTION

MR is a technology in the field of communicating virtual objects to users [1]. Virtual objects are composed of objects that add interactivity in the MR environment. For example, in an art gallery, we can see a beautiful art along with virtual objects about it, such as the profile of the artist and the image which relates to the art. By adding virtual objects in the real world, we can get more knowledge and explanation about real world objects. In our research, we use video data extracted from TV cooking programs to build an interactive cooking support system used in MR environment. The system automatically recognizes users’ cooking actions and provides them with appropriate cooking instructions by means of virtual objects and video data.

Some researches have already been conducted on cooking support systems. We can mainly divide them into two categories. One category focuses on making instructive cooking systems by combining videos and text extracted from recipes. For example in [2], a system instructs the users on the cooking by following procedures derived from the recipe. The other category, like the researches proposed in [3][4], focuses on the interaction with the users. The system instructs the users on their next possible actions by taking into account their past behaviors, which are automatically recognized by means of sensors. In our research, we focus on the interaction of the system with the users and virtual objects to make the system as intuitive as possible.

The remainder of this paper is organized as follows: In section 2, we present the outline of the proposed system. The method to recognize the users’ actions is discussed in section 3, and how to extract cooking data is presented in section 4. In section 5, we show our demonstrative results. Finally, we conclude our proposed system in section 6.

2. THE OUTLINE OF MIXED REALITY APPLICATION

The purpose of this project is to construct an MR application to help the user cook. Our proposed system has two modes which are “recognition mode” and “instruction mode”. The recognition mode is the mode which recognizes the user’s cooking actions. The instruction mode is the mode which provides the user with useful information extracted from TV cooking programs and text recipes. The system switches between these modes to guide the user during the cooking process. For example, in the recognition mode, if the system recognizes the user gripping the knife, it changes to the instruction mode and advises him on how to cut the ingredient with the knife.

Fig. 1 shows the outline of this system. To construct the MR environment, we set two cameras (a color camera and an infrared camera) and a projector over the table. In the recognition mode, the system estimates the user’s state with the two cameras. In the instruction mode, it displays through the projector the cooking process and virtual objects to help him.

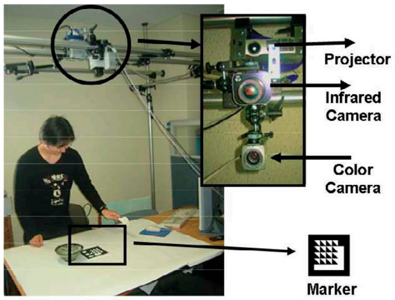


Fig. 1. The outline of mixed reality system.

3. RECOGNITION OF OBJECTS AND ACTIONS IN COOKING

A cooking action is defined as the movement of the users' hands and ingredients on the table. Therefore, before recognizing cooking actions, we have to identify the hands' and ingredients' areas. Recognizing the hands is split into two steps: First, we locate the hands by taking images with both color and infrared cameras. Second, we apply binarization to the hands' area in these images and detect the shapes of the hands.

To recognize ingredients on the table, we also use the infrared camera followed by binarization. Each ingredient's temperature is often lower than the surrounding temperature. Therefore, by recognizing low-temperature areas on the table, we can locate the ingredients. However, we cannot know what ingredient the recognized object is because we cannot extract ingredient's features from the infrared camera. To solve this problem, we use the marker proposed in [5]. Fig. 2 is an example of the markers. The marker consists of 32 bits data area with error detection code.

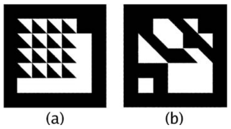


Fig. 2. examples of marker

Table 1. information stored in the markers.

items	examples
ID(marker's number)	1, 45, 255
food or utensil's name	cabbage, knife
cooking actions	cut, mix
color information	$120 < H < 140$, $100 < S < 120$

As shown in Table 1, a marker stores an ID, the name of the ingredient or utensil, the cooking action which can be performed on that ingredient and the color information of the ingredient in HSV mode. For instance, beside the ID and color information, the marker near the cabbage stores the name "cabbage" and the action "cut" because the cabbage need to be cut, and the marker near the portion of pork stores the name "pork" and the action "heat". By using this marker, the system can recognize the ingredient being cooked by the user at that moment.

Recently, RFID tag system [6] is very popular and is used for tracking of items, electronic toll collection (ETC), electric money, and so on. However, it requires a special reader to read the information embedded in an RFID tag. On the other hand, our marker can be read by the color camera and is more simple than RFID tag.

After reading the cooking action stored in a marker, we need to recognize that action when performed by the user. To do so, we monitor the movement of the user's hands on the cooking table. For example, the action "mix" is assumed when the trajectory of the hand movement draws a circle on the table. In detail, we extract the motion vector of the gravity center of the recognized right hand. As the user grips chopsticks with the right hand during the mixing, the shape of the right hand hardly changes. Therefore we think of the movement of the right hand as that of its gravity center.

In addition, when cutting an ingredient, the user's right hand grips a knife and the left hand fixes the ingredient. Hence, the right and left hands move in the same direction and keep constant distance. Consequently, to recognize the cutting, the hands and ingredient have to satisfy the following two assumptions: the ingredient exists between both hands and the distance between the center of gravities of the two hands keeps constant.

4. COOKING DATA EXTRACTION

If the system recognizes a cooking action, it switches to the instruction mode and displays on the table cooking videos to the user. It is necessary to prepare the video materials for instruction about the cooking. To serve this purpose, we extract it from a TV cooking program and a text recipe. And after that, we manually associate a text recipe with the video material. In a cooking program, long shots are used to show the chef and the MC as well as their actions while cooking. Whereas, middle shots are when the camera is focused on the chef. Furthermore, tight shots are used to focus on the cooking and finished dishes. Fig. 3 shows these three shots in a cooking program.

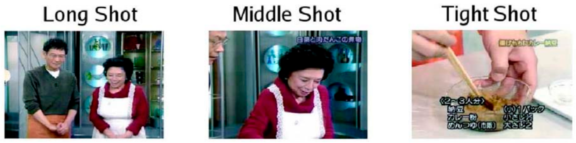


Fig. 3. an example of three types of shots in a cooking program.

To obtain a cooking movie which describes the cooking as detailed as possible, we only extract tight shots from the cooking program. The process of cutting the video into seperate shots is known as *Shot Boundary Detection*, and it is described as follows: First, to distinguish between shots in the cooking program, the optical flow algorithm is applied to every consecutive two frames. The optical flow is calculated from the changes between adjacent frames in the cooking program. If a lot of optical flow is calculated between two frames, the boundary between shots exists within these frames. So we calculate the magnitude of the optical flow between every two frames and split the cooking program at the point where the peak of magnitude appears.

Second, to extract tight shots from sets of shots, we use the Hough transform to each shot. Fig. 4 is an example of the application of the Hough transform. The result of applying Hough transform to a long shot is described in Fig. 4(a). Fig. 4(b) shows the result of applying it to a tight shot. The number of straight lines was 656 in Fig. 4(a) and 206 in Fig. 4(b). As long shots are used to reveal objects around the scene, there are many straight lines in long shots. Whereas tight shots are used for emphasis of the object, and the color distribution is often flat and there are few straight lines.

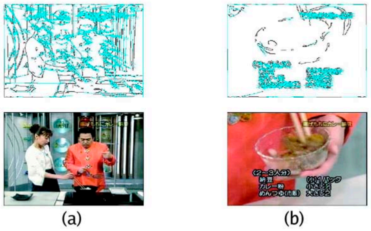


Fig. 4. Hough transform applied in two shots. (a) Long shot. (b) Tight shot.

In addition, for the interactive interface between users and the system, we construct a parallel transition model based on the cooking process in a recipe. In cooking recipes, cooking procedures are described in a sequential manner. However, when we cook, the process is not always sequential and sometimes these steps are treated in parallel or in a different order. By using parallel transition model, the system can add this flexibility in the instruction mode. That is, the user does not have to obey the order of a serial recipe and can handle the steps simultaneously or in a different order as long as the transition model permits to act at that time. Therefore, the steps of cooking might differ from one user to another.

To construct the parallel transition model, we analyse sentences in the cooking recipe and define constraints. For example, the word "after" specifies cooking order, therefore the cooking order extracted from the sentence "cut A after cutting B" is expressed as "B → A". Whereas, a sentence which has the word "and" and commas, like "cut C, cut D, and Cut E", can mean that whether the cooking is done simultaneously or in a different order does not affect the cooking. In addition, some actions, such as "heat", "mix" and so on, must not change the order because this would effect the cooking. Table 1 is an example of constraints based on a cooking recipe. We finally construct the parallel transition model by solving the constraints satisfaction problem according to these constraints.

5. DEMONSTRATION

To demonstrate our method, we choose to cook "Okonomi-Yaki', which is a Japanese-style pancake containing vegetables and other ingredients. Fig. 5 is the reconstructed parallel transition model of a text recipe for "Okonomi-Yaki". We use this transition model to control the two modes in the system.

Table 2. constraints in a text recipe.

word or action	constraint of order
A after B	fix(B → A)
A before B	fix(A → B)
A and B	don't care
A, B	don't care
A and mix	fix(A → mix)
A and heat	fix(A → heat)

At the beginning of the cooking process, the possible procedures are "Cut cabbage", "Put yam in", "Put pickled ginger in" and "Put welsh onion in". When the user touches ingredients which do not relate with these procedures, the system gives an error message. For instance, if the user touches the pork first, since it is not related with any of the above procedures, the system gives an error message. However, if the user touches ingredients related to one of the procedures, the system changes into the recognition mode.

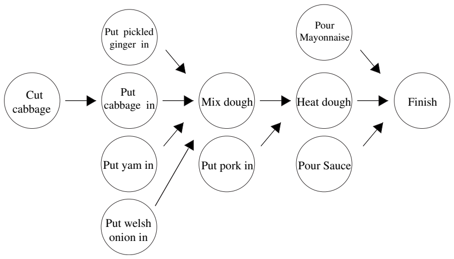


Fig. 5. Transition model constructed by constraints.

To read the corresponding cooking actions stored in relevant markers, the system starts identifying when the user touches ingredients. For example, if the user touches the cabbage, the system identifies this action then reads the knowledge stored in the marker about cabbage.

After recognizing the action, the mode is changed into the instruction mode and the system advises the user on how to cook by means of virtual objects and video data until the user finishes the action. After that, the information stored in the marker is changed so as not to recognize the same action again. That is, the system deletes the knowledge stored in the marker and overwrites it with new knowledge according to the parallel transition model. For instance, if the user finishes "Cut cabbage", according to the transition model, the next procedure is "Put cabbage in". Therefore the system overwrites the action embedded in the marker with the new action "Put in". By overwriting the information stored in the marker, the state also transits to the next state while cooking.

Fig. 6 shows screen shots of our proposed system. Through the color camera, we can get the knowledge stored in the markers and identify the ingredients. The possible cooking actions allowed by the parallel transition model, that the user can perform in the next step are displayed by the projector in the lower right of the table as text. Fig. 7 describes the close up of this area. This text is extracted from the parallel transition model described in Fig. 5. Fig. 7 shows that possible procedures according to the model are “Put cabbage in”, “Put pickled ginger in”, “Put yam in” and “Put welsh onion in”. After the user finishes the action “Cut cabbage”, the procedure “Cut cabbage” changes to the procedure “Put cabbage in” while the other procedures are kept in the parallel model. After the user follows the instructions and touches the next ingredient, the system instructs him/her once again where the ingredient should go by projecting an arrow and a rectangle over the cooking table. The arrow links the ingredient with its destination (rectangle) as shown in Fig. 6(a).

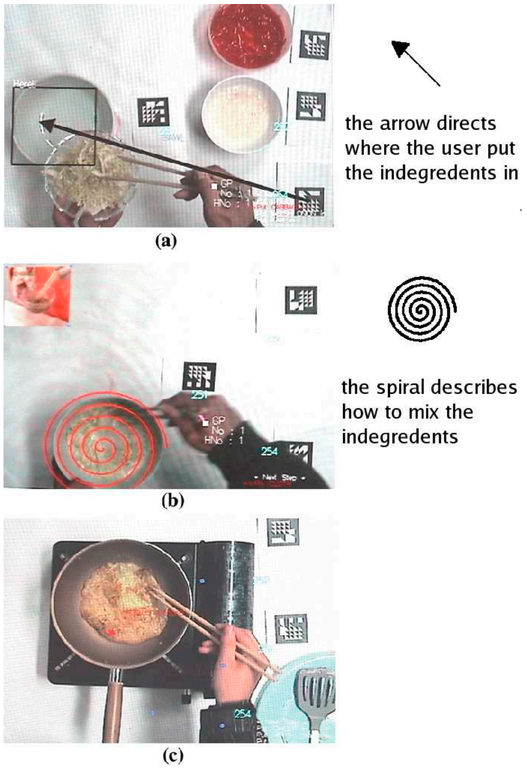
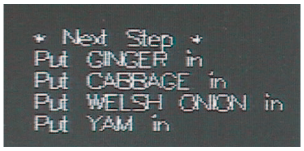


Fig. 6. Screen shots of the system. (a) Put cabbage in a bawl. (b) Mix dough of Okonomi-Yaki. (c) Heat the dough.



```
* Next Step *
Put GINGER in
Put CABBAGE in
Put WELSH ONION in
Put YAM in
```

Fig. 7. Close up of the lower right of Fig. 6(a).

Fig. 6(b) shows the procedure “Mix dough of Okonomi-Yaki”. When the system recognizes that the user is mixing in the recognition mode, the system switches to the instruction mode. In the instruction mode, a cooking movie about “mix” is displayed in the left-upper space on the table. To instruct the user how to move his hand, the virtual spiral object is also displayed on the bowl. Fig. 6(c) shows the procedure “Heat dough of Okonomi-Yaki”. If the system detects high-temperature objects on the table throught the infrared camera, we define the user’s action “heat” and the system changes into the instruction mode. However, the infrared camera we use in this system cannot detect temperature exceeding 100 degrees. Therefore we compare the temperature of the hands with that of the object, and if the temperature of the object is higher, we define the action as “heat”.

6. CONCLUSION

In this paper, we proposed an interactive application in mixed reality environment. The system switches between two modes, recognition mode and instruction mode, according to the movements of the user. In our demonstration, our proposed system performs well when we cook the “Okonomi-Yaki”. However, sometimes mis-recognition occurs when we heat the dough. This is because there is a similarity in both color and temperature between heated dough and the user’s hands. Our future work consists of solving this problem by taking into account the shapes of objects and enhancing our system. We also intend to make our system capable of automatically generating a parallel transition model. We also plan to enhance the system’s recognition section, and make it capable of recognizing other cooking actions.

7. REFERENCES

[1] Paul Milgram and Fumio Kishino, “A Taxonomy of Mixed Reality Visual Displays,” *IEICE Trans. Information Systems*, Vol. E77-D, No. 12, pp. 1321–1329, 1994.

[2] Reiko Hamada, Jun Okabe, and Ichiro Ide, “Cooking Navi: Assistant for Daily Cooking in Kitchen,” *Proc. of 13th ACM Intl. Multimedia Conf.*, pp. 371–374, 2005.

[3] Tsukasa Fukuda, Yasushi Nakauchi, Katsunori Noguchi, and Takashi Matsubara, “Human Behavior Recognition for Cooking Support Robot,” *Journal of Robotics and Mechatronics*, Vol. 17, No. 6, pp. 717–724, 2005.

[4] Taketoshi Mori, Takeru Kuroiwa, Hiroshi Morishita, and Tomomasa Sato, “Assistance with Human Actions and Individuality Adaptation by Robotic Kitchen Counter,” *Proc. of ASER 2004*, pp. 13–20, 2004.

[5] Takeshi Kawano, Yoshihiro Ban, and Kuniaki Uehara, “A Coded Visual Marker for Video Tracking System Based on Structured Image Analysis,” *Proc. of ISMAR’03*, pp. 262–263, 2003.

[6] Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison, “Bridging physical and virtual worlds with electronic tags,” *Proc. of CHI*, pp. 370–377, 1999.