# A Platform for Facilitating Video-based Cooking Communication by Multimedia Technologies

Hidenori Tsuji[1,2], Yoko Yamakata[3], Takuya Funatomi[3], Hiromi Hiramatsu[3], Shinsuke Mori[3]
[1]Institute of Information Technology, Inc. Japan
[2]Institute of Information Security, Japan
[3]Kyoto University, Japan
hide@iit.jp, yamakata@dl.kuis.kyoto-u.ac.jp, funatomi@media.kyoto-u.ac.jp, forest@i.kyoto-u.ac.jp

***ABSTRACT:*** *"Video-based cooking communication" enables people to share the cooking experiences in their home kitchen. Multimedia technologies will facilitate such communication over the Internet. We propose a multimedia processing platform that supports multi-point video communication and provides several functions to easily access raw video and audio. In this system, we can build and test multimedia processing modules as plug-ins to facilitate the communication. We evaluated this system for actual video-based cooking communication in real kitchens, analyzed the conversation during cooking to present the effects of our system. We also present preliminary experiments and discussions how multimedia technologies can facilitate the cooking communication.*

**Keywords:** Multimedia Processing, Platform and APIs, Video-based Cooking Communication, Sharing Cooking Experiences

**Received:** Received 21 March 2012, Revised 25 April 2012, Accepted 1 May 2103



## 1. Introduction

Many people cook in their daily lives. Initially, they might learn culinary skills by cooking with their older family members. This learning style is very effective because they can observe real actions of a skilled home cook. In addition, this style is interactive and thus someone learning to cook can ask questions and immediately receive advice from a skilled home cook. Even after they start to live separately from their family, many would-be cooks want to learn more about cooking. Therefore, they read recipe books, search for recipes on the Web, or watch TV cooking shows. However, recipe books do not always provide sufficient visual information about cooking details. Moreover, with TV cooking shows, cooking students cannot ask questions and cooking experts cannot offer personalized advice by observing a student's cooking techniques. Even after they learn almost all they can from their family members, books, internet, or TV cooking shows, they still want to know various useful tips and interesting variations.

A computer system equipped with some cameras, a microphone, and an Internet connection has the potential to allow students to learn more about cooking as if they were cooking with a skilled cook or a friend. In this paper, we propose a system for "*video-based cooking communication*," in which two persons make their own dishes in different kitchens while communicating with each other over the Internet. For example, with our system, a skilled cook or an instructor can observe the other cook's way of using a knife to slice food and give appropriate advice for the cook to avoid an accident. When a pair of friends cooks a similar dish with our system, they can share the cooking experience to exchange their tips and to learn about an unexpected ingredient from the other.

The existing video-based chat services such as Skype, Avaya, Appia, etc[1]. can be a platform for this purpose. Our preliminary observations, however, had shown that such a communication system requires cutting edge media processing technologies such as image processing and speech recognition to facilitate the communication when users are engaged in cooking or some other work.

Just as [9] automatically controlled cameras to produce a cooking show in a TV studio, multimedia technologies will contribute to natural cooking communication over the Internet. Recently, some researches have been conducted for the cooking domain in the multimedia field [1–5,8,10,12]. To apply these multimedia technologies to facilitate videobased cooking communication, we also propose a multimedia processing platform. The contributions of our platform are as follows:

• Our system supports multi-point video communication and provides several functions to help researchers to easily access raw video and audio. It automatically supports the handling of devices and the control of videoand audio-stream transmission and provides a default user interface (UI). Therefore, researchers do not have to be concerned about handling devices, establishing network connections, and controlling media streams. Researchers can concentrate on developing modules for processing video and audio streams.

• Our system supports to operate on a consumer-level personal computer (PC) and web cameras, and communicates via home networks. It also helps researchers collect data in a practical environment, i.e., in a standard home rather than in their laboratories.

• Our system provides some application programming interfaces (APIs) for researchers to build multimedia processing modules as plug-ins. Our system also provides the settings to test these plug-ins in actual kitchen environment. Because this platform can handle multiple plug-ins, it also helps to accelerate collaborations with other researchers. In this paper, we also present some examples of plug-in implementation and their collaboration.

• We evaluated our system for actual vide-based cooking communication in real kitchens. We analyzed the conversation during cooking to present the effects of our system.We also present preliminary experiments and discussions how multimedia technologies can facilitate the cooking communication.

In the subsequent sections, we first discuss the requirements for video-based cooking communication. Next, we describe the architecture of our platform, which we call *IwaCam*, for video-based communication. In addition, we implement several simple functions using video processing technologies as plug-ins to meet the requirements. Then, we present our experiments of real cooking communication in home kitchens and discussions to investigate the usability of our system. Finally, we give preliminary experiments and discussions of some multimedia processing to facilitate video-based cooking communication.

## 2. Video-based cooking communication

### 2.1 How does it differ from video communication?

For video-based communication to convey the circumstances in each kitchen, the system needs to handle devices such as cameras, microphones, and loud speakers and control the transmission of video and audio streams. In addition, the system must also facilitate human-to-human communication. With the conventional tools for video-based communication, such as Skype, it is assumed that the users sit in front of the display equipped with a single video camera. Because the users keep watching the display and concentrating on the conversation, it is sufficient for the video-based communication to transmit the video and audio of their face or upper body and voice in real time. On the other hand, in the case of cooking communication, the users mainly concentrate on cooking and not communication. Because cooking needs to occupy their attention, they cannot keep watching the display. They also need to pay attention to what they are handling at the sink, countertop, and oven — that is to say, all over the kitchen. Therefore, even if the user keeps listening to the other participants, he or she might not watch the video. Because cooks mostly watch their hands and the food they are preparing, they can only occasionally glance at the display. To understand the situation of the other participants, a desirable feature of the system is the ability to summarize the video and keep it presented on the display as in [11].

### 2.2 Requirements for video-based cooking communication

As a result, a cooking communication system must have the following equipment for each user's environment:

[1]http://www.skype.com, http://www.appiaservices.com/, http://www.avaya.com, *accessed* 2013 *May* 19

**E1. Multiple cameras:** Because a kitchen is wide or sometimes angled, it is difficult to cover the entire kitchen with a single camera. Therefore, the system needs at least two cameras located within the kitchen, e.g. one covers the stove area and the other covers the countertop and sink area. Another camera might be useful to capture the cook's face.

**E2. One microphone:** A microphone is used to detect the voice and transmit it to the other participants. Because it is not necessary to detect all the sounds in a kitchen, for example, the sound of boiling water, a small headset microphone is the most suitable.

**E3. A display and an earphone:** A kitchen must have a display and a loud speaker to show the actions of the other participants. Although a kitchen is very noisy, a cook has to listen to the sounds of cooking as well. Therefore, an earphone is more suitable than a loud speaker.

**E4. A personal computer:** The above devices are connected to a computer with Internet access. The computer must be small to fit into a kitchen environment well and must be capable of executing all the system functions, as we discuss below.

Most of the above equipments are similar to the conventional video-based communication, but different in the number of cameras. Since our system is aiming at the real use, these devices must be consumer-level which is in an ordinary home. Moreover, the system is desirable to provide the following functions for facilitating the communication.

**F1. Temporal summarization:** As [11] proposed, a visual summary of the ongoing cooking will be helpful to the other participants. This must not be a raw video but a summary because there are unnecessary scenes within the entire sequence. For example, an action such as a cook washing his or her hands is not required to illustrate cooking methods. The system must determine key scenes of the instructor for illustrating methods at a glance.

**F2. Spatial summarization:** To summarize actions over time, spatial summarization will also be required. Although the system uses multiple cameras to cover the major actions in the kitchen, they are inefficient to present everything simultaneously, for example, capturing what is happening on the stove, on the countertop, at the sink, and the cook's expressions. As in TV cooking shows, it is required to choose appropriate camera shots to convey the situation in the kitchen. Moreover, a suitable region should be shown in close-up to illustrate the details of the cooking action.

**F3. Camera and scene selection controlled by voice:** The above functions should be performed automatically on the basis of computer vision techniques and not as in [11], which used processes performed with a Wizard of Oz approach. However, such techniques might sometimes fail. For the system to be user friendly, it should provide some methods to manually control the summarizations. As an example, it would be useful to allow the users to switch between cameras using voice commands, thus not having to manipulate a mouse and keyboard.

## 3. Proposed platform and APIs

To satisfy the previously described requirements, much effort will be needed to incorporate multimedia technologies. However, a system for video-based cooking communication also requires much attention to handle devices, establish network connections, and control media streams. Our system, *IwaCam*, supports researchers in multimedia technologies by supporting fundamental functions for video-based communication and provides many application programming interfaces (APIs) to access raw video and audio streams.

### 3.1 Base architecture

*IwaCam* communicates using a bidirectional star topology that can accommodate up to four sites via a TCP/IP network. Communication is limited to four sites in order to meet the requirements of screen separation to allow for the processing capacity of PCs and to provide a human-scale of communication. Figure 1 illustrates a basic four-site communication model. This model adopts a central server and has no privileged users; therefore, all participating users have the same status and all sites (clients) communicate with the central server. With the central server model, it is easy to implement the bidirectional star topology for multiple-user communication within various user-level Internet environments. Each client application can communicate with the other three sites by connecting to the central server. The disadvantage of using a central server model is that the traffic concentrates in the central server. As a result, the server and its connected network form a bottleneck. Although we can solve the problem using a hybrid peer-to-peer technique [13] for load distribution and scalability, we think that it is not essential to

provide a solution at this time, and therefore, we leave it for future work.

*IwaCam* consists of host applications running on a Windows operating system (OS) and a server application named "*roomserver*" (roomsrv) running on Linux, FreeBSD, and any other UNIX-like OS. The host application runs on client computers and can handle up to three cameras. Only three cameras can be used owing to the restrictions of the USB 2.0 bandwidth. The communication protocol of real-time video and audio transmission uses UDP for minimizing communication delay. UDP allows packet drops and hence *IwaCam* also has an alternate assured data transmission method on TCP. To conserve the bandwidth, *IwaCam* compresses video and audio streams using Motion JPEG and Speex codec, respectively.

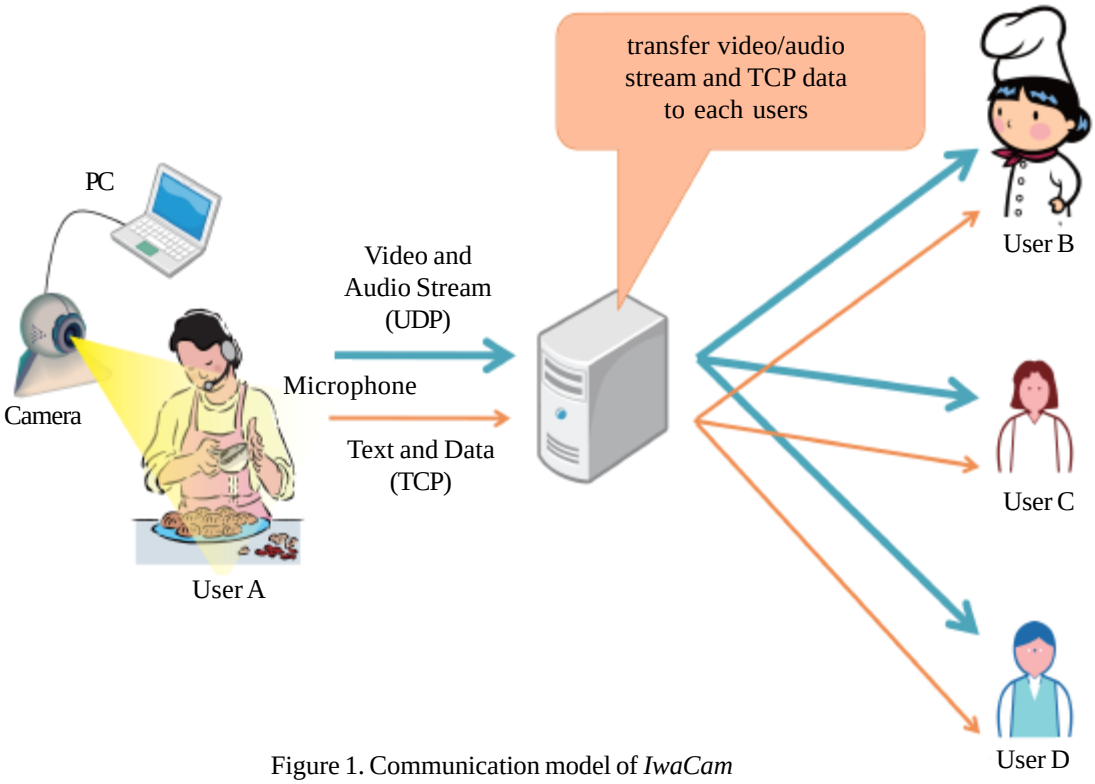



Figure 1. Communication model of *IwaCam*

Because users' computer and network environments can vary, *IwaCam* has a UI for setting parameters. This UI has five setting groups: camera selection, microphone selection, destination server information, video filtering parameters, and network transmission parameters. Using a camera selection setting, a user can select up to three camera devices that are connected to a PC. Microphone selection is performed similar to camera selection, except that there is only one audio source. The destination server's information includes hostname or IP address, connecting port, and username. Filtering parameters configure parameters of video and audio streaming from a device. Network transmission parameters are important for *IwaCam* because a user's network environment can include diverse features, such as delay times, throughput speed, and amount of fluctuation of the communication line. *IwaCam* can be used to select video size, video compression rate, and frame rate.

### 3.2 Plug-in architecture

*IwaCam* enables researchers to add any optional multimedia processing codes in the form of a plug-in. We call that "plug-in architecture." Plug-in codes can directly process sequential frames from input devices. After processing, the plug-in returns the results to *IwaCam* for transmitting the data stream. The plug-in model is shown in Figure 2. Plugin codes can be developed independently from the *IwaCam* host application in the Windows Dynamic Link Library (DLL) format. Therefore, these DLL plug-in codes are applied by simply placing them into the *IwaCam* plug-ins folder. If multiple plug-ins are placed in the folder, they will be all running one by one in file name order.

*IwaCam* uses Microsoft DirectShow libraries for the efficient processing of sequential frames. Microsoft DirectShow is the media streaming architecture for the Microsoft Windows platform, which provides many libraries for handling multimedia

streams.

Figure 3 shows the plug-in structure for processing a video stream. The host application can handle three cameras; hence, each camera's capture streams call the VFrameCallback () function that is defined in the host application. The video stream selector, Camera Switch, selects the video stream to be processed and turns over the stream to the codec section. A plug-in can select or identify an active camera with the SelectCamera () or GetCurrentCamera () functions, respectively. The audio stream architecture is the same as the video stream architecture, except that there is only one audio source.

There are two types of plug-in functions: callback functions and API functions. Callback functions that are defined by plug-ins are called by the host application, and API functions that are defined by the host application are called by plug-ins.

Although there are many callback functions and APIs for handling video and audio streams, *IwaCam* also has reliable data transmission APIs. These APIs such as SendText () and SendImage () functions (see Figure 4) transmit via TCP, whereas video and audio streams are transmitted via UDP. These APIs provide reliable text transfer or binary transfer.

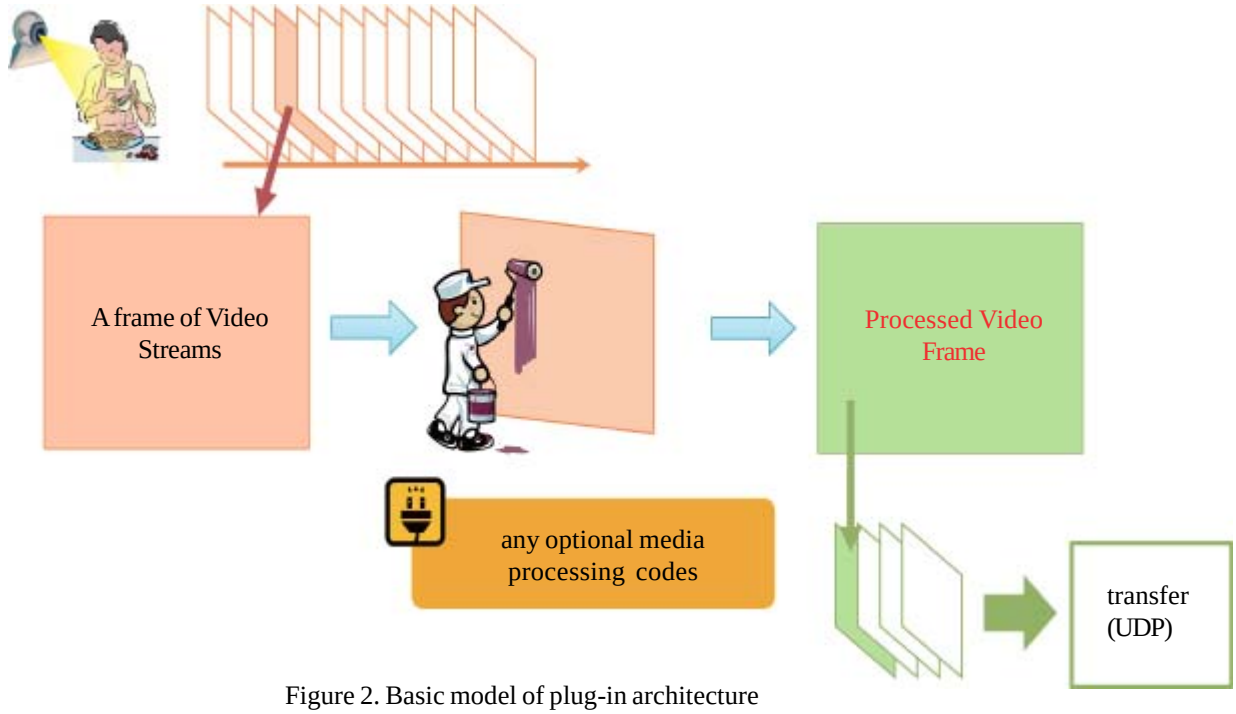



Figure 2. Basic model of plug-in architecture

**3.3 Examples of plug-ins**

We implemented some simple but useful plug-ins for video-based cooking communication on the basis of the *IwaCam* architecture.

**3.3.1 Archiving plug-in**

For analyzing the cooking communication, it is important to review the situation in the kitchens. Therefore, we also implemented a plug-in to archive captured video and audio.

As for video, *IwaCam* cannot guarantee the isochronous capture of images owing to the limitation of the performance of users' computers. Thus, this plug-in archives a video as a sequence of images, each of which has a timestamp in its filename. The plug-in enables users to select the frame rate for controlling the load on the computer. The plug-in also supports several file formats, BMP, PNG, JPG, GIF, and TIFF, for storing the images. When we stored the images of two VGA cameras using the PNG file format, we obtained video with a frame rate of about 8 fps without dropping frames.

As for audio, the plug-in records sound using the WAV format with a sampling rate of 44.1 kHz and a bit rate of 16. Figure 5 shows the user interface of this plug-in that enables users to control archiving and select the output folder, file format, and frame rate. The interface also displays all videos to the users for confirmation.

### 3.3.2 Plug-in for switching cameras

As a simple solution for spatial summarization (F2), we implemented a simple plug-in for switching cameras. This plug-in selects the camera by detecting the cook's activities. In a cooking situation, a cook prepares food in different locations in the kitchen. The cameras capture various scenes in the kitchen. The plug-in continuously evaluates the changes in the scenes from each camera and selects the camera that is capturing scenes in which there is more change or movement than in the scenes from the other cameras. To evaluate the change, the plug-in calculates the change in intensity for each pixel and converts the difference into a binary image by thresholding. The threshold is determined on the basis of the distribution of the differences. When the exposure of the camera is automatically adjusted, the difference will be uniformly enlarged to the entire image.

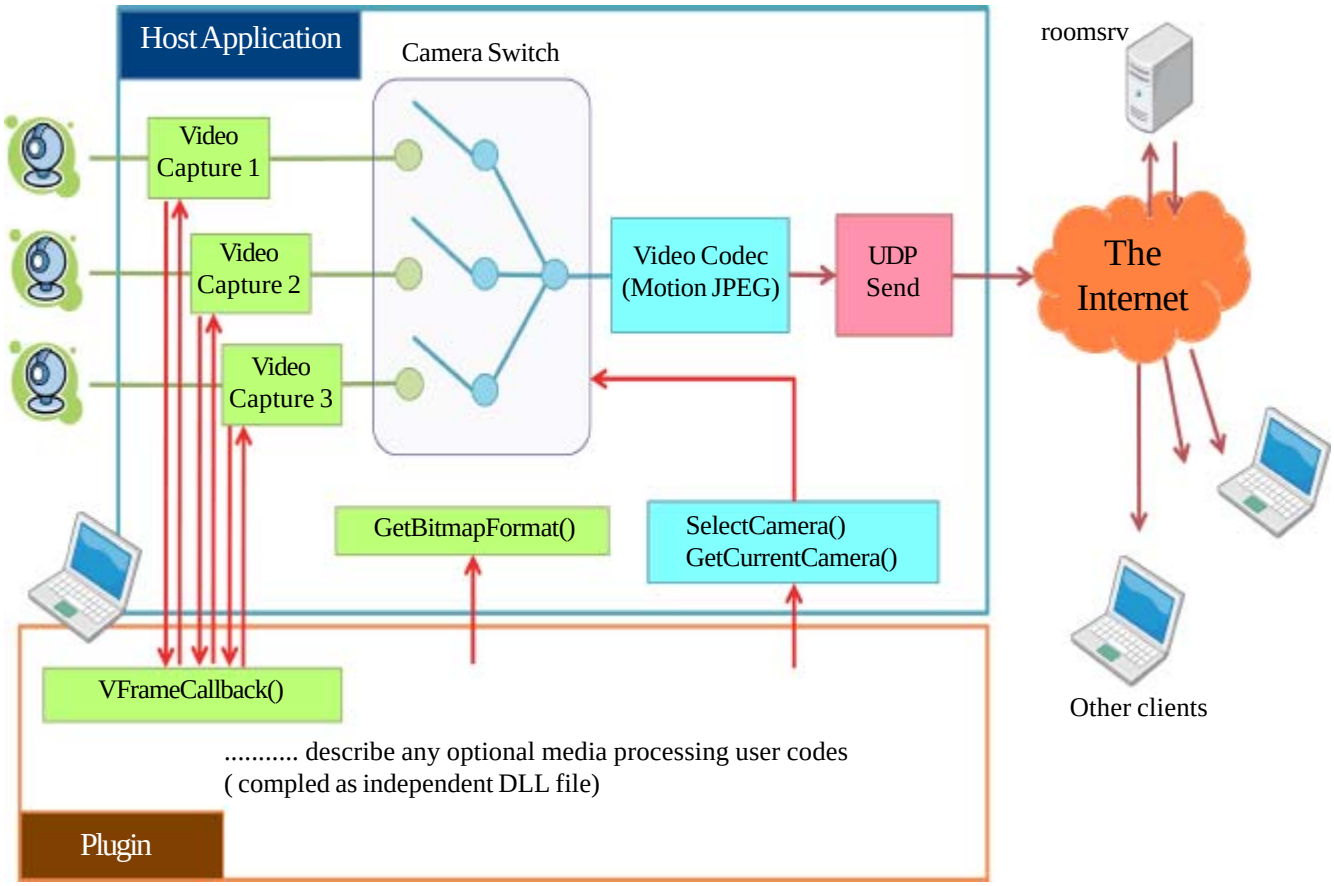



Figure 3. Detail of plug-in architecture (video stream)

### 3.3.3 A plug-in of timeline

As a simple solution for temporal summarization (F1), we implemented a plug-in for taking snapshots to present the timeline of the cooking activities. This plug-in creates a thumbnail image of the captured image and displays recent thumbnails on the window.

This plug-in can collaborate with the plug-in used to switch cameras, thus making thumbnails of the images from the camera that has been selected by the camera-switching plug-in. In this way, *IwaCam* handles spatial and temporal summarization with only two simple collaborating plug-ins.

Figure 6 shows the output window of this plug-in. In this figure, six recent images are displayed and each row corresponds to one site in video communication. In this case, output window has two rows because two sites connected to the same server. The newer thumbnail is placed on the left. This plug-in produces new thumbnail minute by minute from the selected camera in every site and update the recent images. Because the plug-in for switching camera appropriately selects the camera to watch the cooking activities, we can browse the recent activities in these sites.Actually, we can see several camera switching in Figure 6

according to the activities of the users. Due to the plug-in architecture and these plug-ins, both spatial and temporal summarization is achieved.

## 4. Experiments: video-based cooking communication in home kitchens

We tested *IwaCam* and the plug-ins for video-based cooking communication. We assumed the communication would be between two cooks ***T***(eacher) and ***S***(tudent) who know each other well (e.g., a pair of friends or a mother and a daughter). Both of them routinely cook in their homes in their daily lives. ***S*** is less proficient at cooking and wants to learn how to cook a meal from ***T***. If they lived together, sharing the same kitchen, ***S*** would typically learn how to cook from ***T*** by practicing cooking together. *IwaCam* facilitates such cooking communication even in the case where the participants live far apart.

Before they start this experiment, ***T*** connects offline with ***S*** and communicates the ingredient list for the meal. The participant prepares each ingredient individually. Then, they connect to the roomserver using *IwaCam* at the appointed time. When the cooking begins, ***T*** instructs ***S*** on how to cook the meal step- by-step from ***T***'s home, and *IwaCam* captures ***T***'s words and actions and sends the audio and video to ***S***, who learns how to cook by listening and watching the audio and video and following ***T***'s cooking actions step-by-step. ***S*** is allowed to ask questions at any time. ***S*** can also ask ***T*** to suspend cooking whenever ***S*** has trouble or cannot keep up with ***T***.

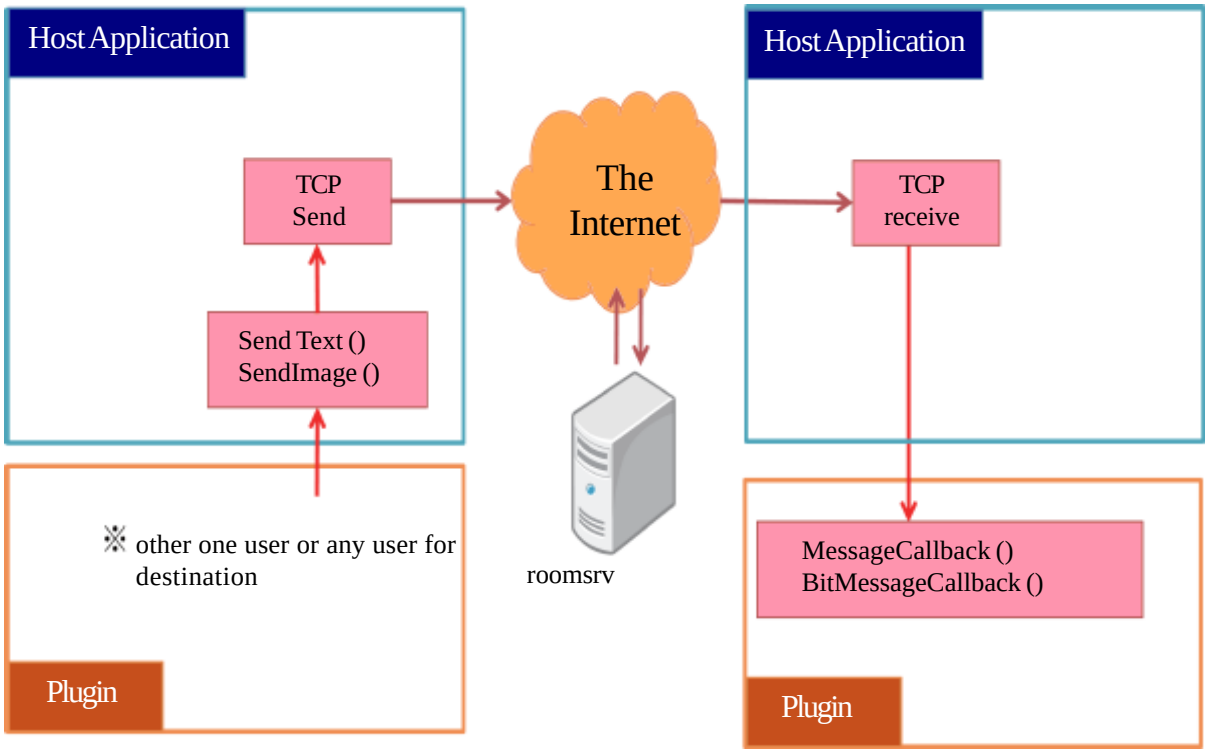



Figure 4. Reliable data transmit APIs

At each step of cooking, they might have the following four types of conversations:

- ***T*** instructs ***S*** about a cooking step they are going to do.
- ***S*** reports whether he or she understands ***T***'s instructions and asks ***T*** whenever he or she has any questions.
- ***S*** notifies ***T*** when he or she finishes the cooking step.
- They chat while performing a given cooking step.

### 4.1 Device settings for experiments

Because kitchen environments vary among homes, device settings should flexibly respond to each environment. We installed

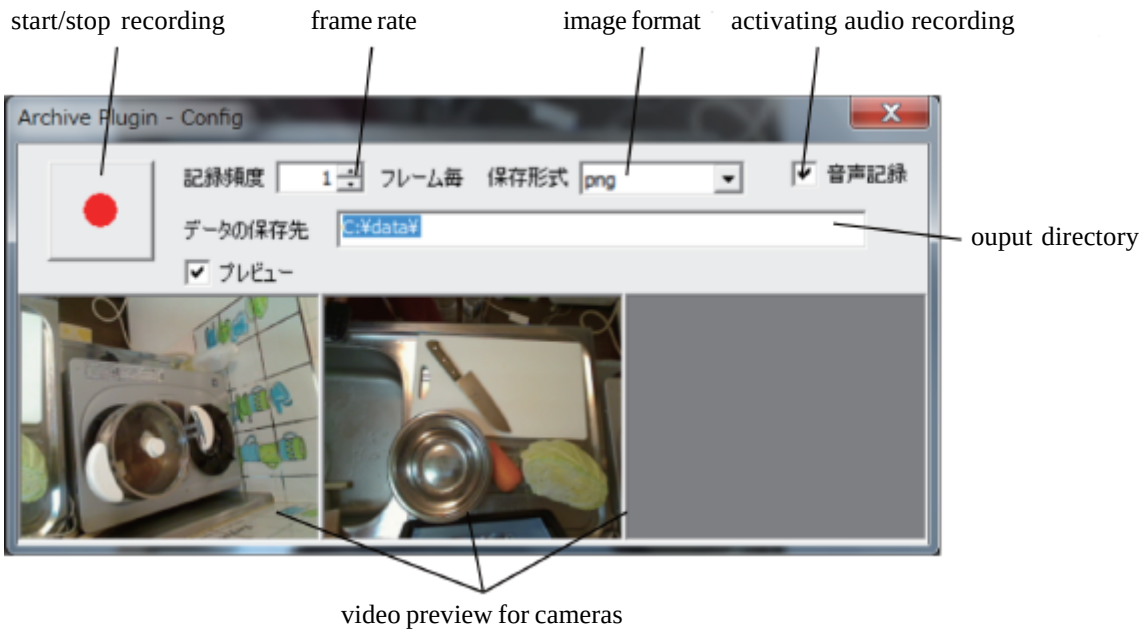



Figure 5. User interface of archiving plug-in


Figure 6. The plug-in of timeline shows the thumbnail images as the timeline

*IwaCam* in the kitchen in the homes of two test subjects.

The photos of Figure 7 shows their kitchen environment. Figure 8 shows the layout of their kitchens and the arrangement of the devices. Each kitchen is composed of a sink, a countertop, and a stove area. We set one PC on the front wall of the countertop in each kitchen. Each PC was an ASUS Eee Slate B121 computer with a 12.1 inch (1280 × 800) LED, an Intel i5-470UM (1.33GHz) processor, 4 GB of DDR3 RAM, and 64 GB of SSD using a Windows 7 Professional 64 bits operating system. *IwaCam* works on the Slate PC and connects to the central server via a wireless LAN and home Internet connection (***T***: E-mobile Pocket WiFi(GP02), ***S***: FTTH supplied by NTT West, Japan).

We set two cameras (Logicool portable webcam C950m) in each kitchen. One camera views the entire countertop and half of the sink area. The other camera views the entire stove area. Each cook wears a headset microphone (Planttronics Voyager PRO+).

**4.2 Transmission speed and video resolution**

IwaCam can select video transmission parameters of the video resolutions, video frame-rate and each video frame compression rate (of Motion JPEG). Each parameter has the following selections:

• **resolution:** 640 × 320, 320 × 240, 160 × 120,

• **frame-rate:** 30fps, 15fps, 8fps, 4fps, 1fps,

• **frame compression rate:** 75%, 50%, 25%, 12%.

So it is difficult to ensure the stable network bandwidth on the various user network environment, an over data transmission causes the drop frame, voice intermissive, and so on. Therefore we adjust those parameters to the extent possible, so that users can grasp the cooking conditions each other. Table 1 shows the user's network bandwidth before each experiment.

In these experiments, we assumed that to recognize partner side cooking is the necessary condition actions. Interestingly, for the framerate it was no problem at the minimum speed (1fps.) because the users concentrated the cooking actions and they glanced at monitor only short time. Thus we first fixed the frame-rate at 1fps and tested the various resolutions and frame compression rates. The trials told that it is difficult to recognize partner side cooking actions at a resolution less than 320 × 240 and a compression rate less than 25%. For a comparison of different resolutions Figure 9 shows two types of resolution, 320 × 240 and 160 × 120, with 25% compression rate. Therefore we decided to use 320 × 240 for the resolution, 25% for the compression rate, and 1fps for the frame rate in the later experiments.

**4.3 Cooking experiments**

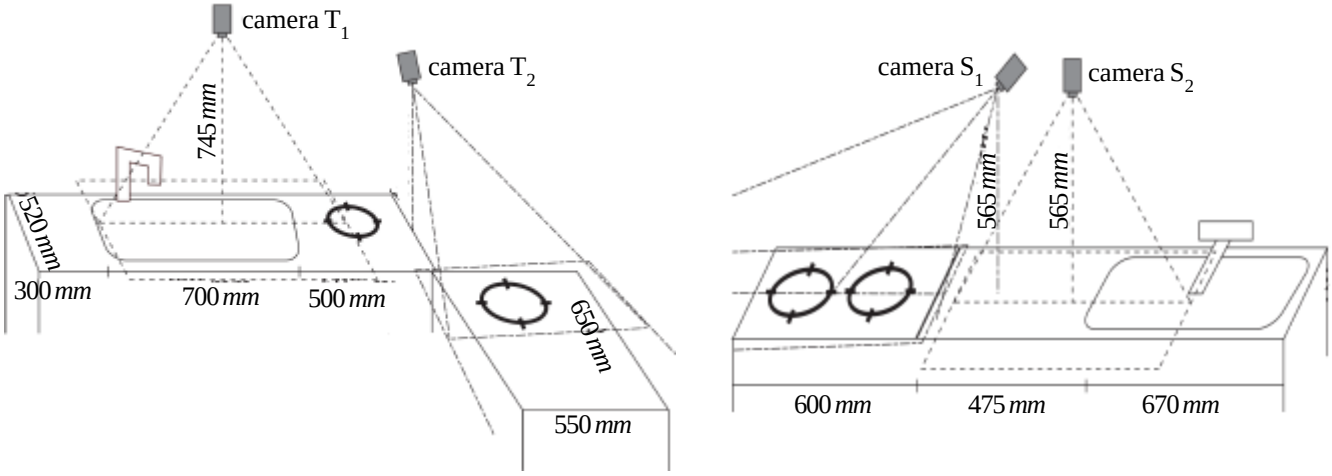
To clarify the aspects of the cooking communication, we tested the system with two subjects assigned as the cooking teacher ***T*** and the cooking student ***S***. The experiments were conducted once per week and for a total of seven times. The meals and cooking times for each experiment are shown in Table 2. To compare the cooking communication when the subjects cooked the same dishes, the two subjects cooked different ones only during ID 7.

Figure 10 shows the captured images. (a) and (b) were captured from ***T***'s kitchen, and (c) and (d) were captured from ***S***'s kitchen. The plug-in selected one camera to send the video to the other cook's display. During the experiments, the selected video images captured from each home were displayed side-by-side on the Slate PCs' screens. The screenshot of the *IwaCam* interface on ***S***'s PC is shown in Figure 11. The upper left portion shows the local video and the upper right portion shows the remote video. The size of each video is about 8 cm × 11.7 cm. In the lower left portion of Figure 11, the load on the PC by the task manager is shown, and in the lower right portion, the interface of the plug-in is shown.

(a) The kitchen of chef ***T***


(b) The kitchen of chef ***S***

Figure 7. Photos of the kitchen environments




(a) The kitchen of chef ***T***

(b) The kitchen of chef ***S***

Figure 8. Configuration in the kitchen environments

### 4.4 Discussions about visual information

#### 4.4.1 Camera setting

Unlike the common video chat methods, both cooks never felt that they wanted to watch the face of the other. However, ***T*** wanted to know whether ***S*** was watching the display when ***T*** taught the way of the cooking action not in words but visually. Both cooks felt comfortable that the cameras captured only the top parts of the kitchen but not their faces, clothes, and other private spaces in their home. The cameras near the stoves did not get dirty by splattered oil because each camera was set obliquely upward on the stove and at the other side of the ventilating fan. However, because the camera position and cooks' viewpoints differed widely, they had difficulty understanding the geometrical relationship between the camera and kitchen scenes from the captured video. Because the stainless steel countertop reflected light like a mirror, the camera capturing the countertop was sometimes selected although the subject was not working there.

#### 4.4.2 Window size on the display for the cooking video

Although the window size for each video was not large enough to understand the details of the cooking action, both cooks did not feel frustrated. In the experiments, at any time, each subject could clarify what cooking actions the other was taking through the use of conversation. This strongly helped the cooks to understand the video of each other's actions. ***S*** wanted to enlarge the video around the area of ***T***'s hands when ***T*** taught the way of the cooking action not in words but visually.

| Date | Kitchen A | | Kitchen B | |
|---|---|---|---|---|
| | Download | Upload | Download | Upload |
| 2012/04/05 09:09-09:15 | 13.77Mbps | 930.13kbps | 15.85Mbps | 3.93Mbps |
| 2012/04/12 09:09-09:11 | 8.25Mbps | 950.35kbps | 25.12Mbps | 4.40Mbps |
| 2012/04/25 18:02-18:03 | 9.94Mbps | 1.00Mbps | 20.53Mbps | 3.68Mbps |
| 2012/05/02 18:02-18:03 | 14.83Mbps | 922.30kbps | 23.28Mbps | 4.11Mbps |

Table 1. Data transmission speed


(a) 320 × 240


(b) 160 × 120

Figure 9. Screenshot of recieved video for each resolution

**4.4.3 Frame rate of cooking video**

Because of the narrow bandwidth of home networks and low performance speed of the Slate PCs, we set the video frame rate at 1 fps for this experiment. Although it seems difficult to understand any action from a 1 fps video, both cooks felt very little stress from viewing the video at this frame rate.

Because the cooks handled dangerous tools, such as sharp knives and hot stoves, they kept watching their hands. In the experiments, both cooks kept watching their own cooking most of the time and sometimes glanced at the video. Because they knew which cooking action they were doing at any time from their conversation, it is considered that just a glance was enough to understand the video.

**4.5 Discussions about speech information**

**4.5.1 Microphone setting**

For speech detection there are three major types of microphones: a stand microphone, a pin microphone, and a closetalk microphone. Since a kitchin is very noisy place, we selected a close-talk microphone. The close-talk microphone was an ear-hook design and both cooks were not annoyed by wearing it while cooking. A pin microphone had been another alternative, thus we had tried one. But it had been unfit for cooking communication because a cook tends to bend over while cooking and then the pin microphone on the cook's neck captures the ambient cooking sounds at the same volume level as the cook's voice.

**4.5.2 Sound quality**

Although the sound quality was not high, it was sufficient for the cooks to keep the conversation going. However, it was insufficient to recognize the conversational speech using an automatic speech recognition system (see Section 5.1). Sometimes the speech sound jumped owing to the narrow bandwidth of the home network and low performance speed of the slate PC, thus disrupting the cooks' conversation.

| ID | Dishes | Cooking time |
|---|---|---|
| 1 | Beef boiled by soy sauce | 28 min. |
| 2 | Steamed pork and chinese cabbage | 55 min. |
| 3 | Boiled cabbage and salmon in milk | 30 min. |
| 4 | *Chikuzenni* | 1h 41 min. |
| 5 | Boiled poak and onion with ginger | 48 min. |
| 6 | Boiled poak with ketchup | 23 min. |
| 7 | Chef ***T***: Steamed pork and chinese cabbage, Chef ***S***: *Nikujaga* | 49 min. |

Table 2. Food name and cooking time on the experiments

### 4.6 Analysis of cooking conversations

In order to figure out how *IwaCam* facilitates the cooking communication, we analyzed the contents of the conversations in cases. One is the case in which ***T*** and ***S*** cook the same dishes and the other is the case in which they cook different ones. As the characteristics of the cooking communication, it is expected that the conversation will be about the cooking and progress simultaneously with the cooking steps.

#### 4.6.1 Case: cook the same dishes

As the same dishes case we selected ID 5 because its duration is close to the average (47.5 min). Here, we briefly introduce the cooking steps of ID 5, "*boiled pork and onion with ginger*", as follows;

**Step 1:** Slice two onions into wedges and grate a piece of ginger.

**Step 2:** Fry a pork with a teaspoon of vegetable oil in a pan till it changes color and add the onions.

**Step 3:** Add 3 cups of water. After boiling, add seasonings (soy sauce, *sake*, and sugar) and cook it over medium heat.

**Step 4:** Boil till only one-third of the liquid remains, add a teaspoon of soy sauce and grated ginger. Remove the pan from heat when the liquid is reduced by half.

According to the video and audio stored by the archiving plug-in, we manually analyzed the conversations between ***T*** and ***S*** during the cooking. We classified the audio area into the following three groups:

**Cooking-talk**: conversations about how to cook,

**Small-talk**: conversations about other things,

**No-voice**: area without any voice between conversations.

The timetable decomposed into these groups is shown in Figure 12. From this figure it can be roughly seen that **Cooking-talk**, **Small-talk**, and **No-voice** accounted for 40.6%, 49.7%, and 9.7%, respectively. Because the **No-voice** areas were short and dispersed, it is difficult to see the silent part. A close look at the figure shows that they mainly talked about how to cook (**Cooking-talk**) in the first half of cooking. In the first half they have to perform cooking actions such as cutting and frying one after another, they have to talk about how to cook.

In contrast, as their cooking goes on **Cooking-talk** gets dominant, because cooking actions to take change.

In ID 5 case for example, since the boiling task (**Step 4**) gave them long waiting time and their tasks seemed to be almost finished in the second half, they spent most of time on talking about various things other than cooking (**Smalltalk**). By way of exception, they had a conversation about how to cook to check the condition of the dishes just before the end of the cooking.

The topics of 87.0% of **Small-talk** were derived from the cooking at that moment. Only 13.0% of **Small-talk** was not related to the cooking at all. Some examples are as follow.

• ***S*** reported that her eyes became irritated when she slices the onions and ***T*** said that wearing contact lenses protect the eyes.

• they told about daily cooking of each other

• they talked that this recipe of Japanese dishes uses a lot of sugar, but European recipes use less sugar.

• ***S*** told about her children. This was only one topic that is not related to the cooking at all.

These results indicate the followings;

• ***T*** and ***S*** almost keep talking during the cooking. This means that both of teacher and student could talk even when they were cooking.

• More than half of the conversations was samll talks. This means that a facilitation for joyful communication is as much important as a facilitation for **Cooking-talk**.

• Because most of topics were derived from the cooking at that moment, we can say that they shared the cooking situation each other in the video-based cooking communication.

We can conclude that *IwaCam* successfully facilitated the cooking communication when a pair of users cooks a similar dish. As

a result, they could share the cooking experience to exchange their tips and the student could learn about cooking from the teacher.

(a) countertop of kitchen **T**

(b) stove area of kitchen **T**

(c) countertop of kitchen **S**

(d) stove area of kitchen **S**

Figure 10. Examples of captured images in each kitchen

• More than half of the conversations was samll talks. This means that a facilitation for joyful communication is as much important as a facilitation for **Cooking-talk**.

• Because most of topics were derived from the cooking at that moment, we can say that they shared the cooking situation each other in the video-based cooking communication.

We can conclude that *IwaCam* successfully facilitated the cooking communication when a pair of users cooks a similar dish. As a result, they could share the cooking experience to exchange their tips and the student could learn about cooking from the teacher.

**4.6.2 Case: cook the different dishes**

In experiment ID 7, the subjects cooked different meals. In this case, the subjects chatted less comparing to that in the other cooking experiments. All of the conversations were **Small-talk** but not **Cooking-talk**. Moreover, most of the topics were not related to the cooking at that moment. The reason will be that they could not share their cooking situation each other and it was difficult for them to find a chance to start conversation.

Furthermore, the frequency of watching the display was also reduced because they did not intend to know about the cooking situation of the other, e.g. what cooking actions the other was doing.

## 5. Preliminary experiments and discussions of multimedia processing

As we described in Section 3 the video-based cooking communication system *IwaCam* offers an access to raw video and audio streams. Thus we can add more intelligent functions to the system based on speech recognition and/or image processing. In this section we give preliminary experiments and discussions of these multimedia processing.

Figure 11. Screenshot of the display of kitchen **T**

### 5.1 Automatic speech recognition results

The system facilitates more than just video communication by detecting what users say. For example, if the transcription of the utterances is added as the subtitle to the video, a cook can read it even when a cook did not hear the other cook under a noisy situation. If the system accept a control by voice commands, a cook can view an important clip video of the other cook whenever he or she wants. Aiming to implement these functions, we conducted a speech recognition experiment. Because the voice recording function in *IwaCam* is designed for human communication and is not suitable for automatic speech recognition (ASR), we transcribed a recorded, real conversation and spoke the sentences to measure the accuracy of the ASR system (respeak). We used the utterances of ***S*** as he/she spoke about the food in experiment ID 5 (see Table 2).

The ASR system we used was Julius-4.0[2]. We used the acoustic model and the language model (word tri-gram model) distributed along the system. **S** respoke the transcribed sentences with a hand microphone in a silent room. We measured the ASR accuracy of 313 utterances under the above conditions and obtained a word accuracy of 47.21%. Table 3 shows the percentages of errors such as substitution, deletion, and insertion.

As we stated in the previous section, most of utterances are related to cooking, we performed a language model adaptation to the cooking domain. We first prepared the following texts:

• Yahoo! QA: We used 1,100,373 QA sentences from the Internet of length less than 200 characters that were categorized by the topics "*cooking*," "*food*," or "*recipe*."

• Ajinomoto recipes: We used 17,070 procedures in recipes in the web page of a food company, Ajinomoto.

[2] *http://julius.sourceforge.jp/* (*Accessed on May* 14, 2012.)

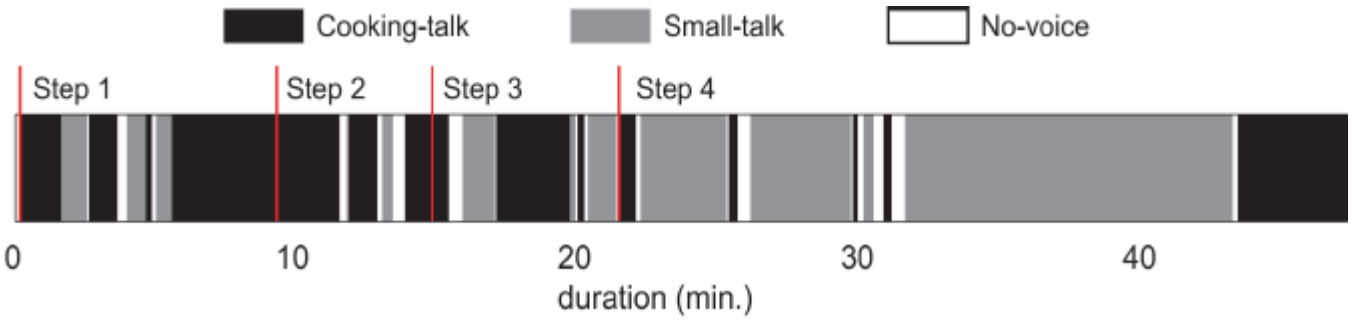



Figure 12. The timetable of the conversations in the experiment ID 5

| Language model | Substitution | Deletion | Insertion | Accuracy |
|---|---|---|---|---|
| General word tri-gram model | 36.13% | 15.18% | 1.49% | 47.21% |
| Adapted word tri-gram model | 31.33% | 8.76% | 2.83% | 57.08% |

Table 3. Speech recognition accuracy

Then we divided the sentences into words and their pronunciations were estimated using KyTea [7][3]. Finally we built a word tri-gram model from the word sequences.

We measured the ASR accuracy of the same 313 utterances with replacing the language model with the adapted one. The word accuracy was 57.08%, which is much higher than the result by the default model for a general purpose. Table 3 shows the percentages of errors such as substitution, deletion, and insertion.

The overall accuracy was not still sufficiently high and there is room for improvement. From the table, the largest contribution to the improvement is the reduction of the deletions and the second is that of the substitutions. A closer observation of the transcriptions revealed that the utterances about foods or cooking procedures were recognized with high accuracy by the adapted model. On the other hand, the ASR system still tended to misrecognize colloquial expressions in human conversations or small-talk about topics other than cooking. They are misrecognized as the cooking domain words, such as food names etc. Thus the percentage of the substitution errors is still high and this is the major type of errors. This is quite natural because these conversations included terms that were not covered by the language model used in the experiment. Because these utterances related to cooking are of importance for cooking assistance, the results indicated that the ASR system can be used in a real environment.

In future work, we need to arrange the recording environment and build an acoustic model more suitable for conversations while cooking. The recognition of small talks is as important as cooking instruction recognition as we described in the previous section. Fortunately, from our observation most of the small talks are related to cooking or daily lives. Thus a language model estimated from a wider topics may increase the accuracy. In addition, a small amount of transcription of small talks could help ASR capture the speaking style and improve the ASR accuracy.

**5.2 Automatic camera switching results**

Our camera switching plug-in (see Section 3.3.2) simply selects the camera, but the scene still includes an area with no change that is of no use to the viewer. Because **S** sometimes wanted to watch **T**'s hand area in close-up in the experiments, it would be more efficient to extract only the working space from the scene. The working space will have a larger amount of change than other parts of the image. Based on this idea, it would be useful to extend the camera switching algorithm to extract the region of interest from the captured scene.

[3]*Because Japanese sentences have no white spaces between words, we used the short unit defined by the National Institute for Japanese Language and Linguistics [6] as the word unit. The word segmentation accuracy is about 95% and the pronounciation estimation accuracy is about 97%. These are enough high for an ASR but are less than that for texts in the training data domain.*

This plug-in, which simply evaluates the change in the appearance, could deal with most global changes caused by theautomatic exposure adjustments. However, because many kitchen instruments are made with metallic materials, their appearance was affected by the change in the nearby environment. Reflections and flashes of light can cause local changes in the scene and might pose problems when we try to extend the switching algorithm to extract the region of interest. Camera switching can also suffer from these reflections. Therefore, it is expected to introduce sophisticated but computationally lightweight image processing to implement robust detection.

From the experiments, it is also required to detect whether the users watch the display and notify each other. This requires an additional camera to capture images of the cooks' faces. Because recent portable PCs have a camera mounted in their display to capture the user's face, it is practical to use it for performing eye tracking.

As for the camera that captures images of the stove area, we also expect to correct the geometrical distortion in the video. For example, by performing homographic transformation, the plane on the stove can be oriented to the plane of the countertop in the other video. However, such transformation cannot correct the distortion of any object that is not on the plane, such as the cook's hands and pots on the stove. This distortion might be more uncomfortable to the users. We also need to consider how to specify camera settings for a wider variety of kitchens.

## 6. Conclusion

In this paper, we propose a system of facilitating "Video-based cooking communication", which enables people to share the cooking experiences in their home kitchen. We first discussed the requirements for video-based cooking communication, and describe the architecture of our platform to support multi-point video-based communication and handle multimedia processing modules as plug-ins to facilitate the cooking communication. We also presented several plug-ins to meet the requirements. Then, we present our experiments of real cooking communication in home kitchens and discussions to investigate the usability of our system. We evaluated this system for actual videbased cooking communication in real kitchens, analyzed the conversation during cooking to present the effects of our system. We also present preliminary experiments and discussions how multimedia technologies can facilitate the cooking communication. Finally, we give preliminary experiments and discussions of some multimedia processing to facilitate video-based cooking communication.

We attempt to introduce state-of-the-art multimedia technologies to implement further functions of facilitating cooking communication and evaluate them in real cooking communication as a future work.

## References


[1] Pei-Yu (Peggy) Chi, Jen-Hao Chen, Hao-Hua Chu, Jin-Ling Lo. (2008). Enabling calorie-aware cooking in a smart kitchen. In *Proceedings of the* 3$^{rd}$ *international conference on Persuasive Technology*, PERSUASIVE '08, p. 116–127.

[2] Keisuke Doman, Cheng Ying Kuai, Tomokazu Takahashi, Ichiro Ide, Hiroshi Murase. (2011). Video cooking: towards the synthesis of multimedia cooking recipes. *In*: *Proceedings of the* 17$^{th}$ *international conference on Advances in multimedia modeling - Volume Part II*, MMM'11, p. 135–145.

[3] Atsushi Hashimoto, Naoyuki Mori, Takuya Funatomi, Masayuki Mukunoki, Koh Kakusho, Michihiko Minoh. (2010). Tracking food materials with changing their appearance in food preparing. *In*: *Proceedings of the* 2010 *IEEE International Symposium on Multimedia*.

[4] Ichiro Ide, Taku Kuhara, Daisuke Deguchi, Tomokazu Takahashi, Hiroshi Murase. (2012). Detection and classification of repetitious human motions combining shift variant and invariant features. In 3$^{rd}$ *International Conference on Emerging Security Technologies* (*EST*2012).

[5] Kranz, M., Schmidt, A., Rusu, R. B., Maldonado, A., Beetz, M., Hornler, B., Rigoll, G. (2007). Sensing technologies and the player-middleware for context-awareness in kitchen environments. In *Fourth International Conference on Networked Sensing Systems. INSS* '07, p. 179 –186.

[6] Kikuo Maekawa, Makoto Yamazaki, Takehiko Maruyama, Masaya Yamaguchi, Hideki Ogura,Wakako Kashino, Toshinobu Ogiso, Hanae Koiso, Yasuharu Den. Design, compilation, preliminary analyses of balanced corpus of contemporary written Japanese. *In*: *Proceedings of the International Conference on Language Resources and Evaluation*.

[7] Shinsuke Mori, Graham Neubig. (2011). A pointwise approach to pronunciation estimation for a TTS front-end. *In*: *Proceedings of the InterSpeech*, 2011.

[8] Nakauchi, Y., Fukuda, T., Noguchi, K., Matsubara, T. (2005). Intelligent kitchen: cooking support by LCD and mobile robot with IC-labeled objects. *In*: IEEE/RSJ International Conference on Intelligent Robots and Systems, (*IROS* 2005), p. 1911 – 1916.

[9] Claudio S. Pinhanez, Aaron F. Bobick. (1997). Intelligent studios modeling space and action to control TV cameras. *Applied Artificial Intelligence*, 11 (4) 285–305.

[10] Spriggs, E. H., De La Torre, F., Hebert, M. (2009). Temporal segmentation and activity classification from first-person sensing. *In*: IEEE Computer Society Conference on Computer Vision and Pattern RecognitionWorkshops, p. 17 –24.

[11] Quan T. Tran, Gina Calcaterra, Elizabeth D. Mynatt. (2005). Cook's Collage: De'ja' vu Display for a Home Kitchen. *In*: Proceedings of HOIT: Home-Oriented Informatics and Telematics, p. 15–32.

[12] Yoko Yamakata, Yoshiki Tsuchimoto, Atsushi Hashimoto, Takuya Funatomi, Mayumi Ueda, Michihiko Minoh. (2011). Cooking ingredient recognition based on the load on a chopping board during cutting. *In*: Proceedings of the 2011 IEEE International Symposium on Multimedia, ISM '11, p. 381–386.

[13] Beverly Yang, Hector Garcia-Molina. (2003). Designing a super-peer network. *In*: International Conference on Data Engineering, p. 49.