IwaCam: a Multimedia Processing Platform for Supporting Video-Based Cooking Communication

Hidenori Tsuji*†, Yoko Yamakata‡, Takuya Funatomi‡, Hiromi Hiramatsu‡ and Shinsuke Mori‡
* Institute of Information Technology, Inc., Nihonbashi-Kobunacho, Chuo-ku, Tokyo, 103-0024, Japan.
Email: hide@iit.jp
† Institute of Information Security, 2-14-1 Tsuruya-Cho Kanagawa-ku, Yokohama, Kanagawa, 221-0835, Japan.
‡ Kyoto University, Yoshida-Honmachi, Kyoto, 606-8501, Japan.
Email: yamakata@dl.kuis.kyoto-u.ac.jp, funatomi@media.kyoto-u.ac.jp, mori@ar.media.kyoto-u.ac.jp

***Abstract*—In this paper, we propose a system for "video-based cooking communication" that allows a home cook to learn cooking by communicating with other skilled cooks or friends over the internet. Because users will want to concentrate on cooking rather than the process of communicating, it is desirable to include multimedia technologies in the system, such as video summarization, to support communication. Our proposed multimedia processing platform for video-based cooking communication, "*IwaCam*," includes software to enable video-based communication and functions to access raw video and audio. Researchers are only required to implement multimedia processing as a plugin to support communication. We also present simple plugins. In addition, we tested *IwaCam* with them in home kitchens. Our platform is friendly to both researchers and users in several aspects.**

I. 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.

Just as [1] 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 researched have been conducted for the cooking domain in the multimedia field [2]–[9]. To apply these multimedia technologies to video-based cooking communication, we also propose a multimedia processing platform to support video-based communication. The contributions of our platform are as follows:

- The 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 video- and 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.
- Researchers can build multimedia processing modules as plugins and test them on the video communication platform. Because this platform can handle multiple plugins, it also helps to accelerate collaborations with other researchers.
- To make the system user friendly, it operates 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.
- By introducing state-of-the-art multimedia processing, the system has the potential to be more user friendly. This will accelerate the transfer from multimedia technologies to the development of support for human-to-human communication.

In this paper, 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 plugins of *IwaCam* to meet the requirements. Then, we report the results of our preliminary experiments to investigate the usability of the system along with real cooking communication experiments in home kitchens. Finally, we discuss future works to implement multimedia processing to support video-based cooking communication.

II. Video-based cooking communication

A. 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 support 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 [10].

B. Requirements

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

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 supporting the communication.

F1. Temporal summarization: As [10] 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 [10], 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.

III. Proposed platform and APIs [11]

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.

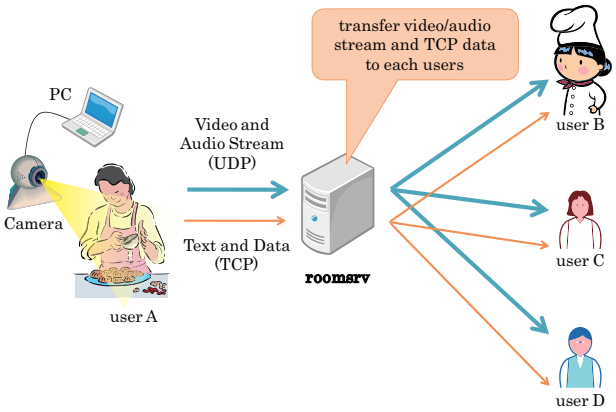


Fig. 1. Communication model of *IwaCam*

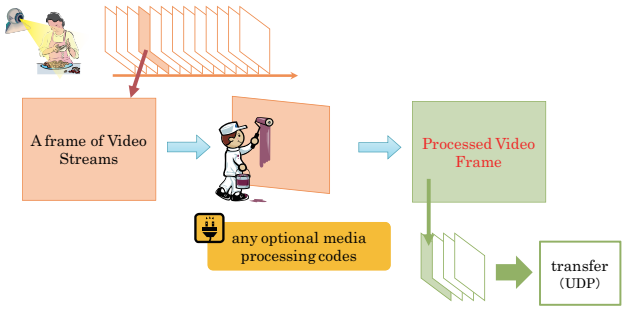


Fig. 2. Basic model of plugin architecture

A. 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 [12] 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.

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.

B. Plugin architecture

IwaCam enables researchers to add any optional multimedia processing codes in the form of a plugin. We call that "plugin architecture." Plugin codes can directly process sequential frames from input devices. After processing, the plugin returns the results to *IwaCam* for transmitting the data stream. The plugin 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 plugin codes are applied by simply placing them into the *IwaCam* plugins folder. If multiple plugins 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 plugin 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 plugin 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 plugin functions: callback functions and API functions. Callback functions that are defined by

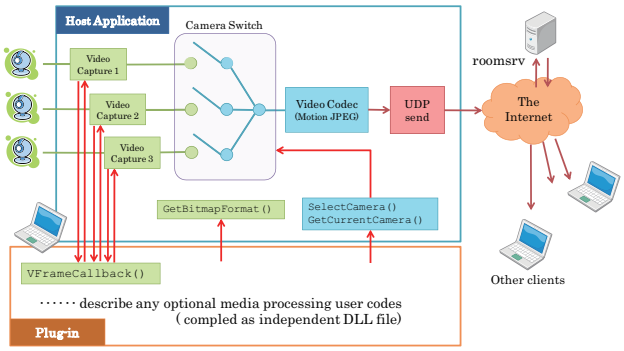


Fig. 3. Detail of plugin architecture (video stream)

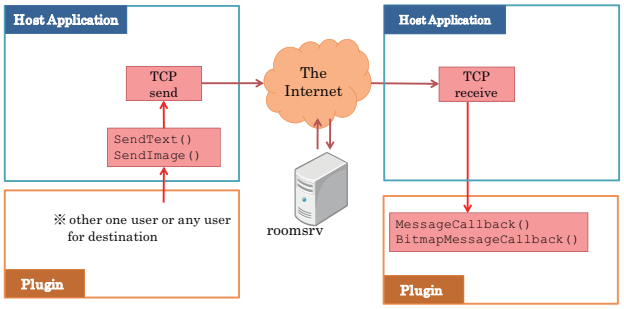


Fig. 4. Reliable data transmit APIs

plugins are called by the host application, and API functions that are defined by the host application are called by plugins.

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.

C. Examples of plugins

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

1) Archiving plugin: For analyzing the cooking communication, it is important to review the situation in the kitchens. Therefore, we also implemented a plugin 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 plugin archives a video as a sequence of images, each of which has a timestamp in its filename. The plugin enables users to select the frame rate for controlling the load on the computer. The plugin 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 plugin records sound using the WAV format with a sampling rate of 44.1 kHz and a bit rate of 16.

This plugin has an interface 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.

2) Plugin for switching cameras: As a simple solution for spatial summarization (F2), we implemented a simple plugin for switching cameras. This plugin 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 plugin 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 plugin 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.

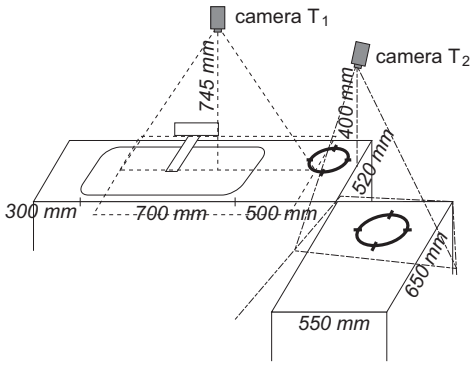
3) A plugin of timeline: As a simple solution for temporal summarization (F1), we implemented a simple plugin for taking snapshots to present the timeline of the cooking activities. This plugin creates a thumbnail image of the captured image using any user-defined condition and displays six recent thumbnails on the window.

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

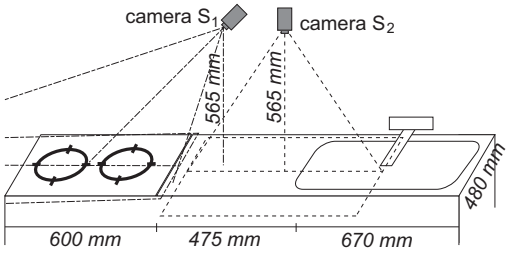
IV. Experiments

We tested *IwaCam* and the plugins 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**.



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



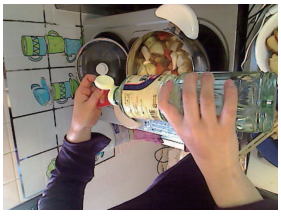
(b) The kitchen of chef **S**

Fig. 5. Iwacam configuration in practical kitchen environments

(a) countertop of kitchen **T**

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

(c) countertop of kitchen **S**



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

Fig. 6. Examples of captured images in each kitchen.

Fig. 7. Screenshot of display of kitchen **T**.

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.

A. Device settings in a practical kitchen environment

Because kitchen environments vary among homes, device settings should flexibly respond to each environment. We installed *IwaCam* in the kitchen in the homes of two test subjects.

Figure 5 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 W/, 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+).

B. Operations and user interface design of IwaCam

Figure 6 shows the captured images. (a) and (b) were captured from **T**'s kitchen, and (c) and (d) were captured from **S**'s kitchen. The plugin 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 7. 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 7, the load on the PC by the task manager is shown, and in the lower right portion, the interface of the plugin is shown.

V. Discussions

To clarify the aspects of the cooking communication, we tested the system with two subjects assigned as the cooking teacher **T** (**T**eacher) and the cooking student**S** (**S**tudent). 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 I. To compare the cooking communication when the subjects cooked the same menu with that when they cooked a different menu, the two subjects cooked different meals only during Experiment ID 7.

TABLE I
FOOD NAME AND COOKING TIME ON THE EXPERIMENTS.

ID	Menu	Cooking time
1	Beef boiled by soy sauce	28 min.
2	[Different Menu for each][1] Steamed pork and chinese cabbage	55 min.
3	Boiled cabbage and salmon in milk	30 min.
4	Chikuzenni	1 h 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.

A. Roles and requirements of visual information

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.

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.

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.

B. Roles and requirements of speech information

Headset microphone setting: Because the headset microphone was an ear-hook design, both cooks were not annoyed by wearing it while cooking. A pin microphone would be another alternative, but it was unfit for this purpose 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.

Sound quality: Although the sound quality was not high, it was sufficient to keep the conversation going. However, it was insufficient to recognize the conversational speech using an automatic speech recognition system (see Section VI-B). 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.

Behaviors when chefs cooked different menu: In Experiment ID 7, the subjects cooked different meals. In this case, the subjects chatted less compared to that in the other cooking experiments because they had difficulty finding time to talk. The frequency of watching the display was also reduced because they did not know what cooking actions the other was doing, and they did not understand the situation from the video with just a quick glance.

VI. FUTURE WORKS IN MULTIMEDIA PROCESSING TO SUPPORT COOKING COMMUNICATION

A. Image processing

Our camera switching plugin (see Section III-C2) 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.

This plugin, which simply evaluates the change in the appearance, could deal with most global changes caused by the automatic 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.

B. Speech processing

The system supports more than just video communication because it recognizes speech. For example, if a cook did not hear the utterances of the other cook, then he or she can view the transcription of the conversation or/view an important clip/video of the other cook by using voice commands. 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 ID5, as shown in Table I,

The ASR system we used was Julius-4.0[2]. We used the acoustic model distributed along the system. **S** respoke the transcribed sentences with a hand microphone in a silent room.

The language model is a word tri-gram model constructed from 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.

Because Japanese sentences have no white spaces between words, we used the short unit defined by the National Institute for Japanese Language and Linguistics [13] as the word unit. The sentences were divided into words and their pronunciations were estimated using KyTea [14].

We measured the ASR accuracy of 313 utterances under the above conditions and obtained a word accuracy of 59.9%. The overall accuracy was not sufficiently high and there is room for improvement. A closer observation of the results revealed that the ASR system tended to misrecognize colloquial expressions in human conversations or small-talk about topics other than cooking. This is quite natural because these conversations included terms that were not covered by the language model used in the experiment. On the other hand, the utterances about foods or cooking procedures were recognized with high accuracy. 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. A language model built from texts containing the transcription of small-talk could improve the ASR accuracy.

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

VII. Conclusion

In this paper, we discussed our implementation of "video-based cooking communication," which allows someone to learn how to cook by communicating with other skilled cooks or friends over the Internet. We discussed about the requirements for supporting such communication and proposed the *IwaCam* architecture, which enables us to introduce multimedia technologies as a plugin to support communication. We tested *IwaCam* and several plugins during video-based cooking communication in home kitchens and studied the aspects of the cooking communication. Because video-based cooking communication is significantly different from conventional video-based communication, we verified that several functions from multimedia processing technologies are required for the communication. Our future work will introduce state-of-the-art technologies to implement such functions and evaluate them during cooking communication.

Acknowledgments

This work was supported by MEXT/JSPS KAKENHI Grant Number 23500137, 23700144 and 24240030. The authors would like to thank Enago (www.enago.jp) for the English language review.

References

[1] C. S. Pinhanez and A. F. Bobick, "Intelligent studios modeling space and action to control tv cameras." *Applied Artificial Intelligence*, vol. 11, no. 4, pp. 285–305, 1997. [Online]. Available: http://dblp.uni-trier.de/db/journals/aai/aai11.html#PinhanezB97

[2] E. Spriggs, F. De La Torre, and M. Hebert, "Temporal segmentation and activity classification from first-person sensing," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, june 2009, pp. 17 –24.

[3] A. Hashimoto, N. Mori, T. Funatomi, M. Mukunoki, K. Kakusho, and M. Minoh, "Tracking food materials with changing their appearance in food preparing," in *The 2nd Workshop on Multimedia for Cooking and Eating Activities*, December 2010.

[4] K. Doman, C. Y. Kuai, T. Takahashi, I. Ide, and H. Murase, "Video cooking: towards the synthesis of multimedia cooking recipes," in *Proceedings of the 17th international conference on Advances in multimedia modeling - Volume Part II*, ser. MMM'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 135–145. [Online]. Available: http://dl.acm.org/citation.cfm?id=1950054.1950071

[5] I. Ide, T. Kuhara, D. Deguchi, T. Takahashi, and H. Murase, "Detection and classification of repetitious human motions combining shift variant and invariant features," in *3rd Int. Conf. on Emerging Security Technologies (EST2012)*, 2012.

[6] P.-Y. P. Chi, J.-H. Chen, H.-H. Chu, and J.-L. Lo, "Enabling calorie-aware cooking in a smart kitchen," in *Proceedings of the 3rd international conference on Persuasive Technology*, ser. PERSUASIVE '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 116–127. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-68504-3_11

[7] Y. Nakauchi, T. Fukuda, K. Noguchi, and T. Matsubara, "Intelligent kitchen: cooking support by lcd and mobile robot with ic-labeled objects," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, aug. 2005, pp. 1911 – 1916.

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

[9] Y. Yamakata, Y. Tsuchimoto, A. Hashimoto, T. Funatomi, M. Ueda, and M. Minoh, “Cooking ingredient recognition based on the load on a chopping board during cutting,” in *Workshop on Multimedia for Cooking and Eating Activities(CEA2011) in conjunction with The IEEE International Symposium on Multimedia 2011*, December 2011.

[10] Q. T. Tran, G. Calcaterra, and E. D. Mynatt, “Cookfs collage: De’ja‘ vu display for a home kitchen,” in *In Proceedings of HOIT: Home-Oriented Informatics and Telematics 2005*, 2005, pp. 15–32.

[11] H. Tsuji, F. Takayama, and T. Matsuzaki, “Iwacam: Cooking vision communication platform software,” *IEICE Tech. Rep. Multimedia and Virtual Environment (in Japanese)*, vol. 109, no. 281, pp. 59–63, nov 2009. [Online]. Available: http://ci.nii.ac.jp/naid/110007521586/

[12] B. Yang and H. Garcia-Molina, “Designing a super-peer network,” *Data Engineering, International Conference on*, vol. 0, p. 49, 2003.

[13] K. Maekawa, M. Yamazaki, T. Maruyama, M. Yamaguchi, H. Ogura, W. Kashino, T. Ogiso, H. Koiso, and Y. Den, “Design, compilation, and preliminary analyses of balanced corpus of contemporary written japanese,” in *Proceedings of the International Conference on Language Resources and Evaluation*, 2010.

[14] S. Mori and G. Neubig, “A pointwise approach to pronunciation estimation for a tts front-end,” in *Proceedings of the InterSpeech*, 2011.