ComeCam: A Communication Support System Between Both Ends of The Live Camera Connection

Jun Higuchi, Shin Takahashi, and Jiro Tanaka
Graduate School of Systems and Information Engineering, University of Tsukuba
{ higuchi, shin, jiro } @ iplab.cs.tsukuba.ac.jp

Abstract

A live camera setup in an office or laboratory provides a rich level of awareness for collaborators who may be separated physically. In particular, a camera with the pan, tilt, and zoom functions provide more detailed information of the monitored room. However, monitoring via a live camera is one-way communication only. We propose two techniques to achieve bidirectional interaction for a live camera connection. The first lets the observed person know when s/he is being watched by transmitting gaze information of the observer. The second enables the observed person to take the action to the observer using simple hand gestures. We implemented the ComeCam *system based on these techniques, and put it into operation in our group.*

1. Introduction

Always-on video provides a rich level of awareness for collaborators who are separated physically [6, 7, 8]. Setting up a live camera link in an office or laboratory, for example, provides an overall impression of how things are going at the other end, as well as who is there and what they are doing. Figure1 is an example of our live camera image setup. Using the pan, tilt, and zoom (PTZ) functions of the camera as we do provides more detailed information of the monitored room, and becomes the *gazing* of the observer clearer.

Since camera observation is one-way communication only, the person being observed may not even notice that s/he is being watched. Bidirectional interaction is important to facilitate communication among people. We believe that providing a way for the observed person to notice that s/he is being watched is the starting point for communication between both ends of a live camera connection. In this paper, we propose two techniques to support work cooperation and easy communication among people at distributed locations.

Figure 1. An example of a live camera image showing the PTZ functions

The first technique lets the observed person know s/he is being monitored, and by whom. This is determined in part from the PTZ parameters of the camera, then it brings the gaze information of the observer to the observed person. This is a first step in facilitating good communications.

The second technique provides a way for the observed person to take the action to the observer. For example, when s/he notices that the observer's attention is on him/her, the person being observed might decide that s/he does not want to be monitored at that time. Our technique enables him/her to make a gesture to the camera to cause a virtual mask to cover him/her in the image captured by any observer. A different gesture enables him/her to attract the attention of all observers. Taking the action to the observer from the observed person provides a means of interaction over the live video connection.

The rest of this paper is organized as follows. In Section 2, we provide an overview and describe the use of our *ComeCam* system based on our proposed techniques. In Section 3, we explain the function of transmitting the gaze information concerning the observer, and in Section 4 we describe the action to the observer. In Section 5, we cover the architecture of the system and the implementation details. Related work is in Section 6, and the conclusion is in Section 7.

2. *ComeCam*: A communication support system

2.1. Overview

We implemented the *ComeCam* system based on the two techniques described above, and used it between different rooms occupied by our group at the university. Briefly, *ComeCam* is a communication system based on live cameras monitoring a distributed workspace. It also enables the detection of information on the observer's gaze and the action from the person being observed.

Our system consists of networked PTZ cameras in each room, and two software applications, *ComeCam Viewer* and *ComeCam Feedback*, running on the PC of each system user. The cameras and PCs are connected to a central server. *ComeCam Viewer* enables a user to monitor other rooms through live camera images, and *ComeCam Feedback* notifies a user that s/he is being monitored. A person being observed can trigger certain actions by making appropriate gestures to the live camera in the room.

ComeCam Viewer is a browser that permits registered users to view live camera images. It can display images from two or more rooms in a single window. Users can control the PTZ parameters of the camera with the *ComeCam Viewer*. Clicking a position on the screen image causes the camera to pan and tilt, so that that position becomes the center of view. Moving the mouse wheel controls the zoom. Since the camera can be controlled by one person at a time only, multiple users take turns. The chat function among the registered users is also provided as part of *ComeCam Viewer*.

ComeCam Feedback and the action to the observer are described in detail in Sections 3 and 4.

2.2. Usage of *ComeCam*

This section describes typical uses of *ComeCam*.

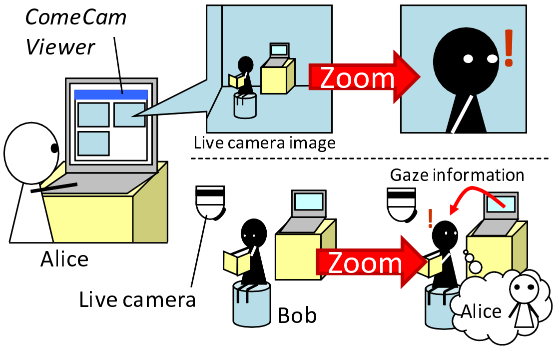

Figure 2. Bob notices that Alice is looking at him.

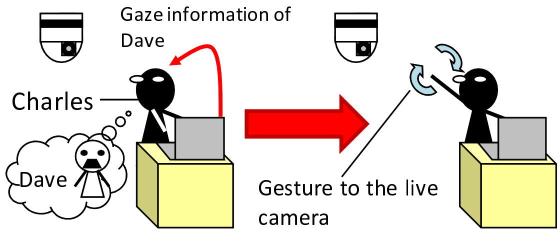

Figure 3. Charles notices Dave's gaze, and makes a gesture toward the live camera.

Case 1: Using *ComeCam Viewer*, Alice is looking at the room where Bob is sitting at his desk reading a book. Alice decides that Bob is not too busy and wants to chat with him. However, Bob is not facing his computer. Alice directly zooms on Bob to attract his attention. Bob notices that Alice is looking at him, immediately invokes *ComeCam Viewer*, and starts to chat with Alice (Figure 2).

Case 2: Graduate student Charles really wants to sleep for a few hours because he was up all night and is very tired. However, he notices that Dave, his professor and boss, is observing him. Charles determines it would not be a good thing to be seen sleeping. Therefore, he makes the appropriate gesture to the camera to mask his area in the camera image (Figure 3).

Case 3: Tom is writing a document, but is uncertain about some of the details that are very visual in nature. He is under a tight deadline and wants to ask for help, but there is no one in the room with him, and no one using *ComeCam Viewer* either. Therefore, Tom makes a gesture to the camera to attract the attention of users in other rooms. The users who notice his call invoke *ComeCam Viewer*. Tom asks them for assistance, speaking while holding up the document to the camera.

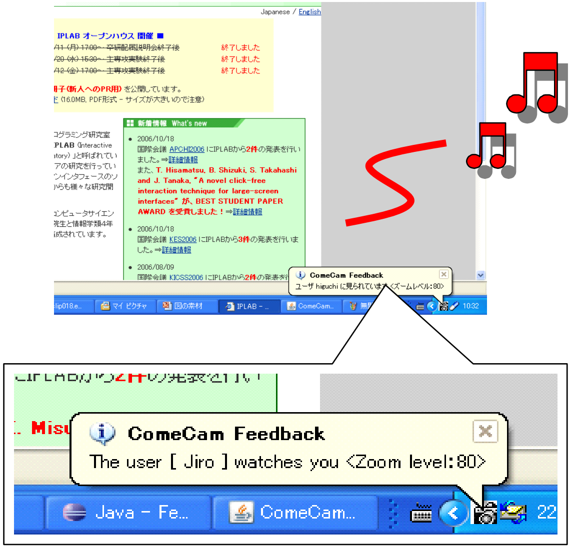

Figure 4. Displaying a message and playing a sound

Figure 5. A hand gesture directed toward the live camera involving the ARToolKit marker

3. Transmitting gaze information

With *ComeCam*, the observer can attract the attention of the person being observed, and the observed person is informed of the observer's attention and identity. *ComeCam* also enables the observed person to attract the observer's attention. *ComeCam* improves on normal one-way live camera use by providing a sense of being in the same room, and facilitating communications between the two locations.

ComeCam Feedback is a background application that transmits the gaze information of the observers to those being observed based on the camera PTZ information. When the observer's *ComeCam Viewer* controls the PTZ parameters, the system checks to see if there is a user in the field of view, and determines if notification is necessary. If so, *ComeCam Feedback* sends the message "User XX is watching you" to the person being observed.

The message pops up on the user's screen, as shown in Figure4. It gives the name of the observer and the zoom level of the camera, out of a maximum of 100. This information is useful because it will help the person being observed to determine how to react. For example, the user may start communicating with the observer if s/he wants to communicate, or s/he may want to mask his/her image if the zoom level is high.

In addition, the system plays a unique sound when the message pops up to let the user know s/he is being monitored even when s/he is not looking at his/her screen. The sound varies according to the zoom level. The higher the zoom level, the more strident the sound is. The user can use the nature of the sound to help determine his/her reaction to the observer.

ComeCam also lets a user attract the attention of observers by making a gesture toward the live camera. This causes *ComeCam Feedback* on all observers' computers to display a message and play a sound. This will alert observers that someone is calling, even if they are not currently monitoring the *ComeCam Viewer* (see Section 4.2.).

4. Action to the observers using gestures

It is natural that a user will want to react to an observer when s/he knows s/he is being monitored. In addition, a user at the observed position may want to attract the attention of users in remote locations. *ComeCam* supports interaction between the observer and the observed person through the live camera connection, and improves communications between them.

In addition to the chat function in *ComeCam Viewer*, *ComeCam* enables the observed person to access certain system functions using hand gestures. Since gestures are more intuitive than using a graphical user interface, the user is likely to feel as if s/he is interacting directly with the person on the remote end of the video link. This will tend to strengthen the communication between the observers and the people they are watching.

To recognize hand movements, we used the marker from the ARToolKit[1] in the current implementation ((Figure5). The ARToolKit provides the relative position of the marker to the camera in three-dimensional space. Currently, *ComeCam* provides two gesture commands: hiding with the mask and attracting the attention of all observers.

4.1. Hiding with the mask

A live camera may capture more details than are really necessary for one-on-one communication. Even without PTZ functions, some people may just not want to be observed in their daily routine.

To deal with this, *ComeCam* enables the user to *hide* behind a mask, that is, the user can display a virtual mask that covers his/her image. This can be considered as one method of reacting to an observer, and is a type of communication between them. The user can display a mask by drawing a circle in the air with his/her hand. The mask is displayed at the position of the drawn circle. Since hand gestures are not linked to the user's PC and are directed at the camera, the user can react immediately once s/he knows s/he is being observed. A user can create multiple masks, which expire after a fixed time, or once the next mask is drawn.

Figure6 (top, left row) shows an example of the masks displayed in *ComeCam Viewer*. The mask is implemented as a three-dimensional graphical object in the scene, so that it can cover the target area even if the camera is panned, tilted, and zoomed (Figure 6, top right). The current implementation includes three masks: a bear, sphere, and cube (Figure 6). The mask type depends on the visual pattern of the marker used to draw it. Since the bear mask covers a large area, it is suitable for masking a person. The sphere or cube might be best to hide a work area or screen display.

The user can display the mask when s/he needs it. The user can turn the mask on or off using *ComeCam Feedback*.

4.2. Attracting the attention of all observers

Sometimes, a user in one room might want other users in remote locations to monitor the image of the user's room. When the user pulls his/her hand backward, the live camera zooms in on him/her. Simultaneously, the system identifies the person from his/her location in the room and transmits his/her

[1] ARToolKit http://www.hitl.washington.edu/artoolkit/

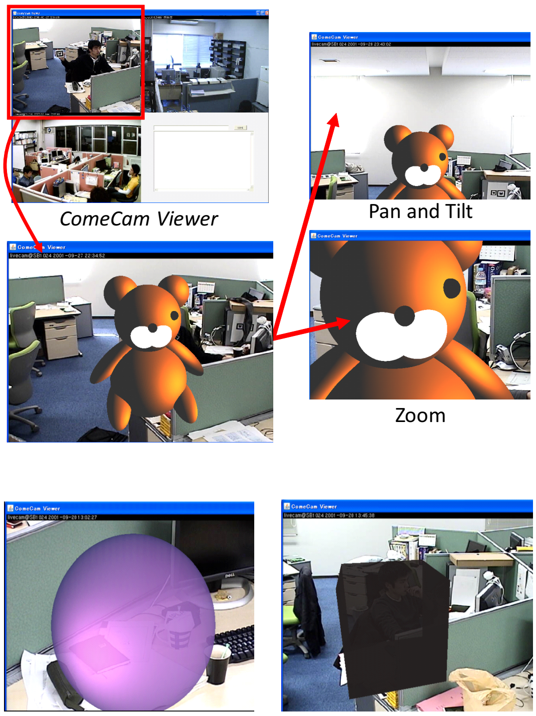

Figure 6. An example of a mask displayed in *ComeCam Viewer* (top left)
Panned, tilted, and zoomed camera images (top right)
The mask patterns (bottom)

name to all observers using *ComeCam Feedback* with the message "User XX wants you to watch" accompanied by a distinctive sound.

Using this technique, the user can immediately focus the camera on him/herself, and call for observers to select his/her image using *ComeCam Viewer*. The observers who notice the person calling can start to chat with him/her through *ComeCam Viewer*. Getting the attention of the observers is essential for bidirectional interaction.

5. Implementation

5.1. System architecture

Figure7 shows the architecture of our prototype *ComeCam* system in operation in three separate rooms in the university. *ComeCam* consists of a live, networked PTZ camera and two or more user client PCs in each room, plus the central *ComeCam Server*.

The *ComeCam Server* receives the MPEG video streams from each live camera, overlays masks if necessary, and transmits the images to *ComeCam Viewers*. Consequently, the raw unmasked video streams are never sent to users directly. The *Come-*

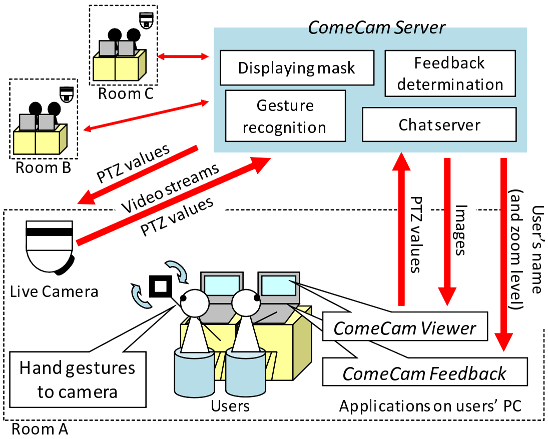

Figure 7. System architecture

Cam Server also detects markers using the same video streams, and recognizes the marker movement as gestures.

We used AXIS 214 PTZ[2] cameras. The parameters for these cameras can be queried or set simply by referring to specific URLs.

Each time the PTZ values of the camera change, *ComeCam Server* acquires the pan and tilt angle values to determine if feedback is required. If feedback is necessary, *ComeCam Server* transmits the observer's name and current zoom level to *ComeCam Feedback* running on the target user's PC. Similarly, if the pulling movement of the marker is recognized, *ComeCam Server* causes the camera to zoom in on the user making the gesture and transmits his/her name to *ComeCam Feedback* running on all observers' PCs. *ComeCam Server* also provides a chat service for the *ComeCam Viewer* clients.

5.2. Feedback determination

In this system, the positions of the user and camera are handled as three-dimensional coordinates relative to the size of the room. Currently, the user position is fixed, based on his/her desk position, since most of his/her work activity is centered there. It is necessary to measure the absolute size of the room at least once, but the relative positions of desks to the camera are easily determined using the ARToolKit and its marker.

First, the ray of the camera is defined as a half line of infinite length with starting point P (Figure 8). We also define the *ray vector* as a basic vector d, which represents the direction of the ray of the camera. The ray vector is determined from the current pan and tilt angles of the camera.

Next, we define the user's occupancy area as a sphere centered on the user's position, C, with radius r. The user sets the value of r. If the ray of the camera intersects the user's occupancy area at least once, the system decides that the camera is pointing to the user, i.e., s/he is being monitored and requires feedback.

[2]AXIS 214 PTZ http://www.axiscom.co.jp/prod/214/

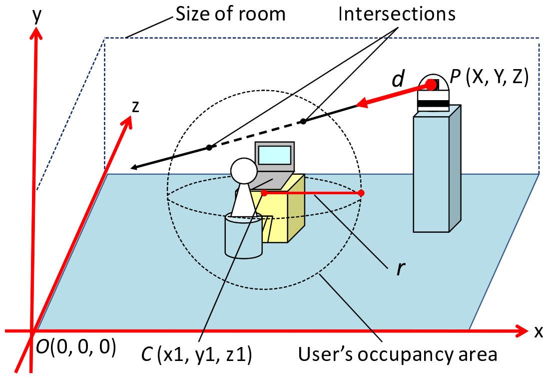

Figure 8. Feedback determination

5.3. Displaying the mask corresponding to the camera PTZ

When the PTZ operations move the camera, the two-dimensional area of the image to be masked changes considerably, and so it must be calculated in three-dimensions. We use three-dimensional graphical objects as masks, and place them virtually at the position of the user or specified work areas that require masking. By matching the direction and angle of view of the virtual camera to those of the real camera, we can correctly overlay three-dimensional objects on the real camera image. More specifically, the direction of the virtual camera is calculated from the pan and tilt angle values of the real camera, and the horizontal range of view of the virtual camera is calculated from the zoom level of the real camera. We use the Java OpenGL (JOGL[3]) library to draw the three-dimensional masks.

6. Related Work

There has been other research on indicating the virtual presence of distributed collaborators using video images [1, 2, 3]. Weak Gaze Awareness [5] determines a remote person's gaze information as

[3]jogl: https://jogl.dev.java.net/

well as our research from images of his/her face. Although these techniques share images for one-on-one communications and needs special devices, our techniques use only the live camera to provide continuous coverage of the daily routine in the entire room. Since our system is not static, it also supports the gaze awareness of the users in three-dimensions.

GestureCam [4] is a video-mediated communication system with a moving camera mounted on the actuator, which supports distributed workspace collaboration between an instructor and an operator. *ComeCam* can do this too, as well as connects the distributed rooms using an always-on video image, and our proposal teqchniques for use in daily routine functions facilitate bidirectional interaction between users.

Other research [6, 7, 8] used a camera to provide an always-on video image in the distributed workspace, and discussed the trade-off between context awareness and privacy. In particular, Context-Aware Home Media Space [8] fed back the camera state to the user, just as our technique does. However, none of these studies link the change of view to the PTZ parameters of the camera, or mask the minimum necessary area set by the user as *ComeCam* does.

7. Conclusion and Future work

This paper has proposed techniques to enhance the function of live camera links in collaborative communications. In addition to simple monitoring through a browser, we used the PTZ parameters of the camera to enable the observed person to be aware of the observer's attention. We also enabled the observed person to take the action to the observer through hand gestures directed at the camera. We implemented *ComeCam*, a communications support system based on these proposed techniques, and operated it between different rooms belonging to our group at the university. *ComeCam* improved the use of live cameras for one-way communication by permitting bidirectional interaction between both ends of the live camera connection.

For the precise feedback determination, the user has to set appropriate the value of radius r of his/her occupancy area. We should make the interface helps him/her to set it easily. In its current implementation, *ComeCam* displays a message and plays a sound to indicate both that a person is being observed, and that s/he needs attention. We plan to implement other presentation techniques, such as ambient media display [9]. Tracking the position of a user who does not remain at his/her desk location and gesturing without the use of markers or tools are other areas for future work.

Acknowledgements

The authors wish to thank all members of IPLAB (Interactive Programming Laboratory) in University of Tsukuba for providing motivation and some advices on this research.

References

[1] H. Ishii, M. Kobayashi, and J. Grudin. Integration of Interpersonal Space and Shared Workspace: ClearBoard Design and Experiments. ACM Transactions on Information Systems (TOIS), 11(4):349-375, 1993.

[2] O. Morikawa and T. Maesako. HyperMirror: Toward Pleasant-to-use Video-Mediated Communication System. the 1996 ACM Conference on Computer-supported Cooperative Work, pp149-158, 1998.

[3] H. Kuzuoka, J. Yamashita, K. Yamazaki, A. Yamazaki. Agora: A Remote Collaboration System that Enables Mutual Monitoring. in CHI'99 Extended Abstracts, pp.190-191, 1999.

[4] H. Kuzuoka, T. Kosuge, M. Tanaka. GestureCam: a video communication system for sympathetic remote collaboration. the 1996 ACM Conference on Computer-supported Cooperative Work, pp35-43, 1994.

[5] T. Ohno. Weak Gaze Awareness in Video-Mediated Communication. CHI'05 extended abstracts on Human factors in computing system, pp1709-1712, 2005.

[6] S. E. Hudson, I. Smith. Techniques for addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support System. the 1996 ACM Conference on Computer-supported Cooperative Work, pp248-257, 1996.

[7] M. Boyle, C. Edwards, S. Greenberg. The Effects of Filtered Video on Awareness and Privacy. the 2000 ACM conference on Computer supported cooperative work, pp1-10, 2000.

[8] C. Neustaedter, S. Greenberg. The Design of a Context-aware Home Media Space. UBICOMP 2003 Fifth International Conference on Ubiquitous Computing, pp297-314, 2003.

[9] H. Ishii, S. Ren, P. Frei. Pinwheels: Visualizing Information Flow in an Architectural Space. in CHI'01, pp111-112, 2001.