WEBDOVE: A WEB-BASED COLLABORATION SYSTEM FOR PHYSICAL TASKS

Weiyi Yang[1], Jiazhi Ou[1], Yong Rui[2], Jie Yang[1]

[1]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA

[2]Microsoft Research
Microsoft Corporation
Redmond, WA 98052, USA

ABSTRACT

While many systems are available for audio-visual people collaboration and data collaboration, systems for collaboration on physical objects are few. In this paper, we present WebDOVE, a system designed to address the needs of collaborative physical tasks. WebDOVE supports both live video streams and pen-based gesture recognition in multi-party bidirectional communication via inexpensive web cameras. WebDOVE allows distributed collaborators to draw over video streams to produce and interpret pointing and representational gestures as readily as they do in face-to-face settings. To accommodate potential diverse platform requirements from different participants, WebDOVE is designed to be a web-based platform-independent and browser-independent collaboration solution. We show via experiments that despite WebDOVE's platform independency, it requires moderate network bandwidth and CPU load, which make WebDOVE a practical solution for real-world applications.

1. INTRODUCTION

With the globalization process, more and more projects are becoming distributed. While there are solutions available today for people and data collaboration, *e.g.*, PolyCom [5], Tandberg [6], WebEx [7] and LiveMeeting [4], solutions for distributed physical tasks are few. Collaborative physical tasks, in which two or more people interact with physical objects, play an important role in education, manufacturing, and tele-medicine. For example, a mechanical expert of Boeing may need to supervise several workers to assemble airplane parts in different locations. Similarly, surgical experts from different countries may assist in a medical procedure in another country. During physical tasks, communications involve both speech and gesture. As they talk, people use gestures to clarify or enhance their messages [1][12]. Studies have indicated that collaboration benefits from shared visual space, where different people can see the same physical objects at the same time. For example, when equipped with a camera at the workspace, task performance is faster than when audio-only configuration is used [2]. Recent studies have further demonstrated that allowing remote experts to point and gesture in the workspace via a shared live video stream further improves performance [3].

However, combining speech and gesture is complicated in remote collaboration because of the need to reference external objects. Previous approaches to this problem have used specialized and/or expensive equipment that makes their widespread adoption unlikely (*e.g.*, [8][10][11]). Our work in 2004 DOVE (Drawing Over Video Environment) is a multimodal system that allows collaborators to point and draw pen-based gestures over video streams [13]. In DOVE, the workspace is visually shared through an Internet Protocol (IP) camera and equipped with computers. A real-time video stream from the camera is sent to the expert's computer. The expert can make freehand drawings and pen-based gestures on the touch-sensitive screen of a computing device, overlaid on the video stream, just like using a real pen on a piece of paper. In the case of a regular computer (desktop or laptop) is used, an expert can make freehand drawings using a mouse. The results are observable by both the expert and the worker. While DOVE is a good step towards solving distributed physical tasks, it has several limitations:

- DOVE requires a special network IP camera instead of a regular camera, *e.g.*, a cheaper USB web camera.
- DOVE only provides unidirectional video and pen-based gesture communication instead of bidirectional.
- DOVE only supports two participants, one expert and one worker. It facilitates communications via two unidirectional channels: video from the worker to expert, and pen-based gesture/drawing from the expert to worker. Many applications might require multiple parties to work together to accomplish the task.
- DOVE is runs on Microsoft Windows only. As different parties may very likely use different platforms, it is highly desirable to have a platform-independent solution. In addition, this solution needs to be efficient in terms of network bandwidth and CPU usage even if it is platform-independent.

In this paper, we develop and present WebDOVE, a web-based bidirectional multi-party collaboration system to support physical tasks via inexpensive cameras (*e.g.*, USB web cameras). Specifically, in Section 2, we discuss how WebDOVE solves the problems mentioned above. We will describe both the overall system architecture and each of the system components. In Section 3, we report experiments to demonstrate that WebDOVE is not only platform-independent, but also moderate in its network bandwidth and CPU usage so that it can be used in real-world applications. We conclude the paper in Section 4.

2. SYSTEM ARCTHITECTURE AND DESIGN

WebDOVE is designed as a reconfigurable system that allows an arbitrary (up to the bandwidth limit of the network) number of experts and/or workers to collaborate over the Internet. In order to achieve the aforementioned goals, we employed a Java-based infrastructure. Compared with other popular web application development tools (such as Microsoft .Net, ActiveX controls, DHTML), Java is the only platform-independent (*e.g.*, Windows, Linux, Mac OS), browser-independent (*e.g.*, Mozilla, IE, Netscape, Firefox) foundation for robust and complex web based applications. WebDOVE has the following important features:

- It uses regular web cameras instead of network IP cameras. Since web cameras are widely available and cost only about 10% of IP cameras, it makes a low cost system possible.

- It facilitates bidirectional communication between any pair of collaborators. Both the expert and the worker can now see each other's video environment when they are equipped with video cameras. By sharing his/her own visual space, the remote expert can show examples over a distance and facilitate discussions. In addition, both the expert and the worker can draw their pen-based gestures over any shared video environment. As a result, instead of simply following the expert's instructions, the worker can raise questions by drawing over the video stream and be engaged in a more active discussion.
- It supports multiparty communication. Instead of supporting a single expert and a single worker, WebDOVE can support multiple experts and multiple workers working together.
- It is a web-based system. There are a few advantages to a web-based design. First, it is platform-independent. Second, it makes software distribution and update easier. When a new version comes out, instead of installing new software to every computer that participates in the system, we update the web server and all participants will run the same version upon connecting to the server.

2.1 System architecture

There is a web server, a "Mediator" and multiple "Collaborators" in WebDOVE. A Collaborator can be used by either an expert or a worker. Figure 1 illustrates the architecture for the "one expert with two workers" scenario. There are two channels of communication: video streams and messages. Video streams are sent between Collaborators in a direct point-to-point mode using real-time protocol (RTP) [14]. Collaborator sign-on and sign-off, drawing data, and controlling commands are handled by a central server called the Mediator. The Mediator keeps a list of the Collaborators, monitors the availability of them and allows an easier and more stable implementation of bidirectional multiparty communication. In Figure 1, all Collaborators are equipped with video cameras (not necessarily of the same type). The video environment of each participant is shared with all the others. Pen-based gestures are communicated as messages through the Mediator.

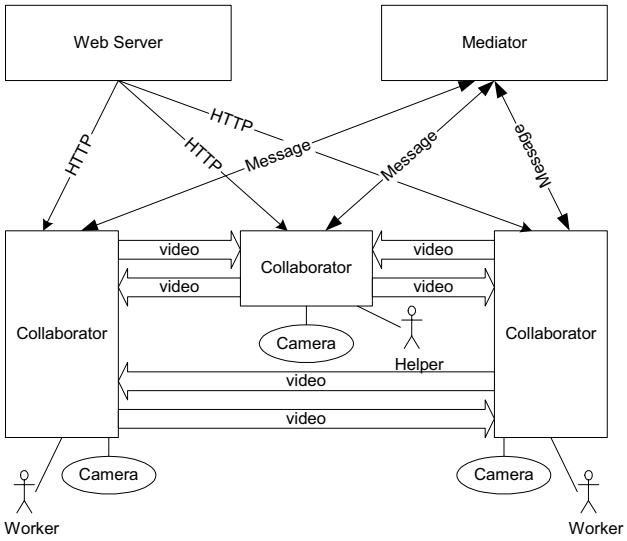

Figure 1. WebDOVE system architecture for one expert (helper) with two workers.

When a new user comes in, he/she first connects to the web server and downloads the web interface and java codes. He/she then runs the java applet (Collaborator in Figure 1) in his/her browser window. Each Collaborator captures, encodes, and transmits the video stream to other participants, and receives and displays the video from other Collaborators (Figure 2). There is a main video window for the user to work on. All other videos are shown in small size windows. After the establishment of a connection, the user can draw over the video image in the main window and the results will be observable on the corresponding windows of all other users' interface. He/she can choose any of the video feeds as the main working window by clicking the small live videos.

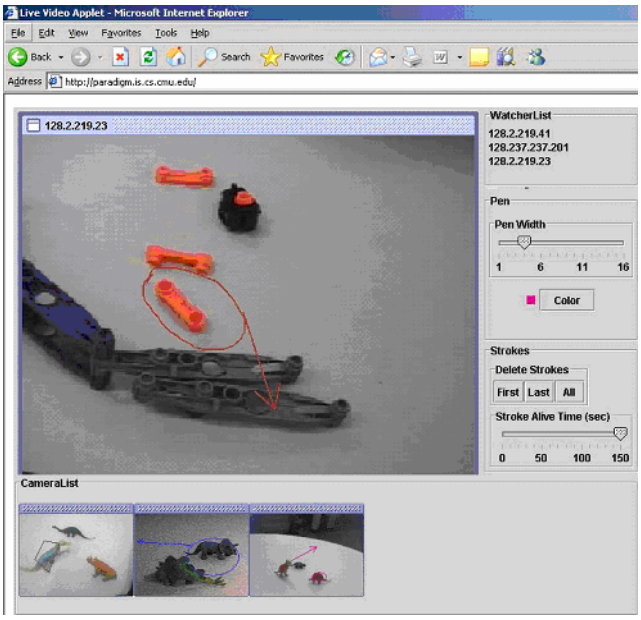

Figure 2. Close-up of WebDOVE interface on one of the Collaborators. The user was helping a worker in the main window. There were three small windows showing the video environments of two other workers and his/her own. He/She could choose any of them as the main working window by clicking on the small videos.

2.2 Web server and mediator

The web server and Mediator are stand-alone applications. The web server provides the participants an HTML page, which includes the web interface, java applet codes, IP address and the port number of the Mediator. It does not participate in communications afterwards.

The Mediator retransmits messages and gestures between Collaborators, handles Collaborator sign-on, detects and notifies users of Collaborator sign-off. The use of the Mediator was inspired by [9]. Through the Mediator. a Collaborator can send a message either to all other Collaborators through the "broadcast" mode, or to a specific Collaborator using the targeted Collaborator ID. The "broadcast" message is especially useful when a Collaborator is newly connected, and does not know about its peers. A simple yet robust sign-on and sign-off process is very important for reliable operations. An example of two Collaborators' sign-on sequences is shown in Figure 3. Each Collaborator gets a unique ID upon connecting to the Mediator. It will then try to negotiate two rounds of handshaking, one as a

presenter if equipped with a camera, one as a watcher. Collaborator 1 connects first and sends two broadcast messages to the Mediator. It does not get any replies since there are no other collaborators yet. Collaborator 2 connects later, and starts a three-step presenter handshaking process. First, Collaborator 2 sends a "broadcast camera" message with its ID. Secondly, Collaborator 1 receives it and replies a 'watcher' message with its own ID. Thirdly, Collaborator 2 creates a video session for Collaborator 1, and replies a 'watcherACK' message that encodes the RTP address of the video session. Collaborator 1 will connect to the RTP address and start receiving Collaborator 2's video. The watcher handshaking have two steps that are similar to the last two in the presenter handshaking process. Mediator maintains a connection to each Collaborator, and if one of the connections become unavailable, it will try to notify the remaining Collaborators about the missing one.

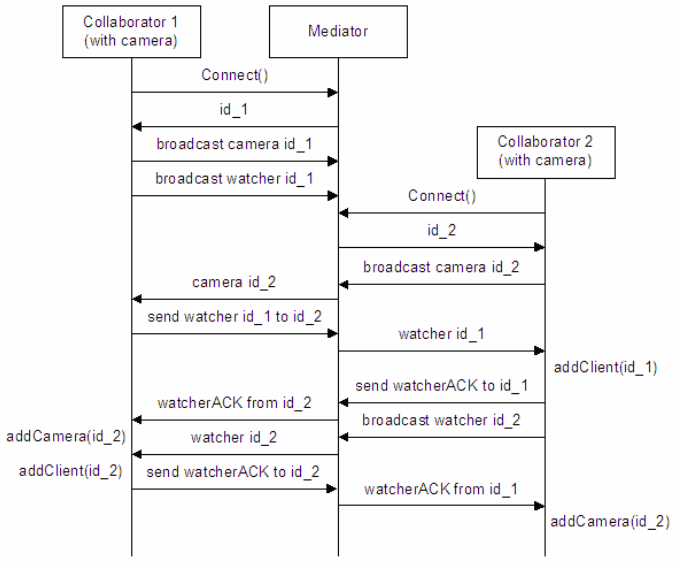

Figure 3. An example of registration of two collaborators

2.3 Video communication

To support video capturing, encoding, transmitting, receiving, decoding and playing-back, we utilize the Java Media Framework (JMF) [15]. JMF specifies a unified architecture to synchronize and control audio, video, and other time-based data within Java applications and applets. JMF provides an interface for video capturing over different lower level APIs (such as Video for Windows, Video for Linux, and SunVideo), and transmits video using RTP [14].

2.4 Overlay gestures on videos

To display pen-based gestures without flickering on videos is trivial but critical to user experience. In DOVE, we need to combine video and pen-based gestures frame by frame. With JMF, WebDOVE takes advantage of Java's GlassPane concept to synthesize gesture drawings and video feeds in a more object-oriented manner GlassPane is an object with which one can draw on top of the ContentPane, a basic Java drawing object. A series of points are captured between pen-down and pen-up events. They are shown in strokes on GlassPane, while videos are displayed in ContentPane. Therefore, the synthesis of video and pen-based gestures is done efficiently and automatically.

2.5 Gesture normalization

Besides freehand drawing, WebDOVE provides the gesture normalization function similar to DOVE. In gesture normalization mode, the current sequence of points will be sent to a gesture recognition module immediately after the user lifts the pen from the screen. The strokes will be recognized as one of the following pen-based gestures: line, folded line, polygon, arrow, round arrow, and circle. The normalization module computes the parameters and presents the normalized images. Examples of the normalized gestures are shown in Figure 4. The current version of the software recognizes 12 pen-based gestures [13]. These 12 gestures are the most common freehand drawings selected from preliminary user studies during a collaborative construction task. We utilize a hierarchical structure of classifiers that consist of hidden Markov models (HMMs) and decision trees to achieve good performance for the pen-based gesture recognition task.

Figure 4. Remote pen-based gesture recognition and normalization in a robot-building task. A freehand oval and arrow (left, center) have been recognized and regularized (right).

2.6 Other features

WebDOVE provides users with three additional capabilities. First, users can set parameters for their sketches, including pen width and drawing color. Second, a set of buttons allows users to erase all gestures, their first gesture, or their latest gesture. Third, the user can specify an "automatic erase" mode, in which gestures disappear automatically after a predefined time.

3. EXPERIMENTAL RESULTS

It is easy to see the benefits of having web cameras (vs. IP cameras), bidirectional (vs. unidirectional) communication, and multi-party (vs. two-party) collaboration. In this section, we will therefore focus our experiments on two tasks: 1) the network traffic and CPU load for our platform-independent solution, 2) the accuracy of our pen-based gesture recognition.

3.1. Network traffic and CPU load

While it is commonly believed that Java applications are less efficient than those implemented in C/C++, we did preliminary experiments to evaluate the performance of WebDOVE.

To test the efficiency of JMF's video encoding and transmission scheme, we recorded the network usage on one participating PC, which had a 2.8GHz CPU and 512M memory (Figure 5). These data were captured as the total bits/sec received from the network interface card using standard Microsoft performance analysis tools. The web server, Mediator, and Collaborator were run on this computer during a period of 24 minutes and 45 seconds as shown in Figure 5. We started with three Collaborators, including the monitored computer itself. All Collaborators were equipped with video cameras. In this three-Collaborator scenario, the network bandwidth was around 160k bps. Later the network usage was recorded with total numbers of two and four concurrent Collaborators. The maximal bandwidth usage, with four concurrent Collaborators, was around 460k bps, which is acceptable to most of the network connections.

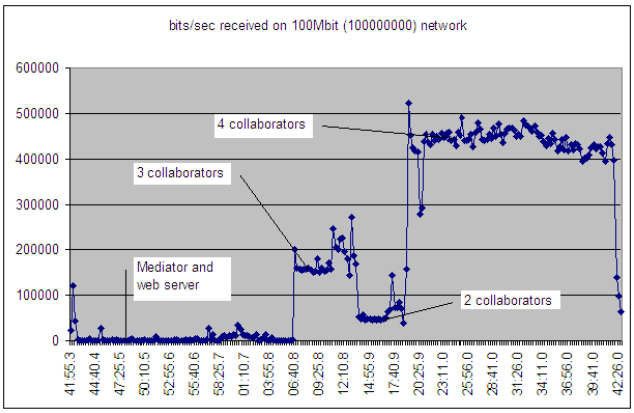

Figure 5. The network traffic of one participating computer. It runs the web server, Mediator, and a Collaborator.

CPU usage percentage on the same PC was shown in Figure 6. When there were only the web server and mediator running, the CPU usage was minimal. When it was running as a Collaborator itself with three other Collaborators, an average CPU usage between 50% and 60% was reported. About 25-30% of CPU time was used by the local video capturing, encoding and transmitting. Adding more Collaborators to the network, which adds more decoding loads, has a less-than-significant effect on CPU usage.

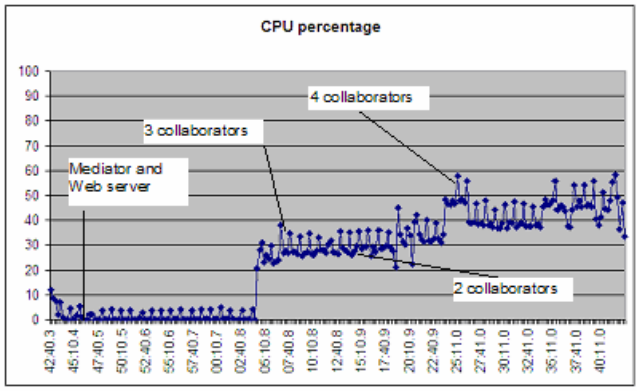

Figure 6. CPU usage (in percent) of the same computer.

Table 1. Recognition Results for Individual Gestures

Gestures	Accuracy	Gestures	Accuracy
Straight Line	100%	Check Mark	97.3%
Cross	98.2%	Delete	92.8%
Arrow	94.6%	Round Arrow (Clockwise)	94.6%
Round Arrow (Counterclockwise)	95.5%	Ellipse	99.1%
Triangle	100.0%	Quadrangle	89.2%
Pentagon	96.4%	Star	100.0%

3.2. Gesture recognition

WebDOVE inherited its pen-based gesture recognition from DOVE. We have performed experiments to evaluate the accuracy of the pen-based gesture recognition. We collected 1337 gestures from 14 people and performed two-fold cross validation. We first calculated the accuracy for each gesture individually, and then averaged the accuracy of each class. The overall accuracy of 12 gestures is 96.4%. Table 1 lists the recognition results of individual gestures.

4. CONCLUSIONS

We have presented the WebDOVE system to support multiparty collaborative physical tasks. WebDOVE is a platform-independent solution by utilizing the JMF technologies. We showed that despite WebDOVE's platform-independency, it requires moderate network bandwidth and CPU time. Furthermore, WebDOVE enables the collaborators to draw over video streams to produce and interpret pointing and representational gestures as readily as they do in face-to-face settings. Our experiments showed that the recognition results for pen-based gesture recognition are quite promising.

5. ACKNOWLEDGEMENTS

This research is partially supported by the National Science Foundation under Grant No. 0208903 and a gift grant from Microsoft Research. We thank Susan Fussell and Jeff Wong for their valuable discussions.

6. REFERENCES

[1] Bekker, M. M., Olson, J. S., & Olson, G. M. (1995). Analysis of gestures in face-to-face design teams provides guidance for how to use groupware in design. Proceedings of DIS 95.

[2] Fussell, S. R., Setlock, L. D., & Kraut, R. E. (2003). Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks. Proceedings of CHI 2003.

[3] Fussell, S. R., Setlock, L. D., Yang, J., Ou, J., Mauer, E. M., & Kramer, A. (2004). Gestures over video streams to support remote collaboration on physical tasks. HCI, 19, 273-309.

[4] http://main.livemeeting.com/

[5] http://www.polycom.com/home/

[6] http://www.tandberg.net/index.jsp

[7] http://www.webex.com/index.html

[8] Ishii, H., Kobayashi, M., & Grudin, J. (1993). Integration of interpersonal space and shared workspace: ClearBoard design and experiments. ACM Transactions on Information Systems, 11, 349-375.

[9] Jim Farley (1998). Java Distributed Computing. O'Reilly.

[10] Kuzuoka, H., Kosuge, T., & Tanaka, K. (1994) GestureCam: A video communication system for sympathetic remote collaboration. Proceedings of CSCW 9 4 (35-43).

[11] Kuzuoka, H., Oyama, S., Yamazaki, K., Suzuki, K., & Mitsuishi, M. (2000). GestureMan: A mobile robot that embodies a remote instructor's actions. Proceedings of CSCW 2000, pp. 155-162.

[12] McNeill, D. (1992). Hand and mind: What gestures reveal about thought. Chicago: University of Chicago Press.

[13] Ou, J., Fussell, S. R., Chen, X., Setlock, L. D. & Yang, J. (2003), "Gestural Communication over Video Stream: Supporting Multimodal Interaction for Remote Collaborative Physical Tasks", Proceedings of International Conference on Multimodal Interfaces, Nov. 5-7, 2003.

[14] RFC 1889 - RTP: A Transport Protocol for Real-Time Applications.

[15] Sun Microsystems, Inc. (2005). Desktop Java - Java Media Framework API (JMF), http://java.sun.com/products/javamedia/jmf/index.jsp.