2003

Gestural Communication over Video Stream: Supporting Multimodal Interaction for Remote Collaborative Physical Tasks

Jiazhi Ou
Carnegie Mellon University

Susan R. Fussell
Carnegie Mellon University

Xilin Chen
Carnegie Mellon University

Leslie D. Setlock
Carnegie Mellon University

Jie Yang
Carnegie Mellon University

Gestural Communication over Video Stream: Supporting Multimodal Interaction for Remote Collaborative Physical Tasks

Jiazhi Ou, Susan R. Fussell, Xilin Chen, Leslie D. Setlock, Jie Yang
School of Computer Science
Carnegie Mellon University
Pitsburgh, PA 15213 USA
{jzou,sfussell,xlchen,lsetlock, yang+}@cs.cmu.edu

ABSTRACT
We present a system integrating gesture and live video to support collaboration on physical tasks. The architecture combines network IP cameras, desktop PCs, and tablet PCs to allow a remote helper to draw on a video feed of a workspace as he/she provides task instructions. A gesture recognition component enables the system both to normalize freehand drawings to facilitate communication with remote partners and to use pen-based input as a camera control device. Results of a preliminary user study suggest that our gesture over video communication system enhances task performance over traditional video-only systems. Implications for the design of multimodal systems to support collaborative physical tasks are also discussed.

Categories and Subject Descriptors
H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces - *Collaborative computing, Computer-supported cooperative work, Evaluation/methodology.*

General Terms
Algorithms, Performance, Design, Experimentation, Human Factors.

Keywords
Gestural communication, gesture recognition, video stream, video conferencing, video mediated communication, computer-supported cooperative work, multimodal interaction

1. INTRODUCTION
In this paper we describe the DOVE (Drawing Over Video Environment) system to support multimodal communication during collaborative physical tasks—tasks in which two or more people interact with real objects in the 3D world. Collaborative physical tasks play an important role in many domains, including education, industry, and medicine. For example, a remote expert might guide a worker's project, or a surgical expert might assist in a medical procedure at another location. Because the expertise required to perform collaborative physical tasks is becoming increasingly distributed, there is a critical need for technologies to support their remote accomplishment. Despite this need, however, the majority of previous systems for remote collaboration have been designed to support activities that can be performed without reference to the external spatial environment. Consequently, these systems have limited application in contexts in which physical objects play a key role.

Prior studies of physical collaboration suggest that people's speech and actions in this context are inherently multimodal, intricately related to the position and dynamics of objects, other people, and ongoing activities in the environment [1][2][3][6][13][20]. In particular, communication during physical tasks incorporates both speech and gesture. As they talk, people use several types of gestures to clarify or enhance their messages [1][16]. Pointing gestures are used to refer to task objects and locations. Representational gestures, such as hand shapes and hand movements, are used to represent the form of task objects and the nature of actions to be used with those objects, respectively.

When collaborating face-to-face on physical tasks, people can readily combine speech and gesture because they share the same environment. Combining speech and gesture is more 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][14][15]). In the current work, we aim to develop an inexpensive multimodal system that can be easily incorporated into existing video conferencing systems. Our goal is to allow remote collaborators to communicate about their physical world through speech and gesture with the same ease as they can do so when co-located.

The DOVE system supports remote interaction using gestural communication over video streams using video cameras, tablet PCs, and desktop PCs. DOVE allows collaborators to share the workspace through video connections. It also provides remote support for pointing and representational gesture by overlaying pen-based gestures on video streams.

An example of our technique is shown in Figure 1. Here, a remote medical expert draws on a tablet PC (lower left) on which is displayed a video feed from an operating room. The same video fed with the expert's gestures overlaid is shown on a PC monitor in the operating room (upper right) and on another doctor's PDA (lower right).

ICMI'03, November 5–7, 2003, Vancouver, British Columbia, Canada.

Figure 1. An example of gestural communication over a video stream in a medical context.

DOVE further provides support for gesture recognition, both to enhance interpersonal communication and as a camera control device. Unlike existing gesture recognition systems used for human computer interaction, which support recognition only of predefined gestures, DOVE supports recognition of predefined gestures, freehand drawing, and a combination of the two. We discuss the DOVE system and our gesture recognition algorithms in detail. We also evaluate DOVE's gesture recognition accuracy and assess the value of the system through user studies.

The remainder of the paper is organized as follows: In Section 2, we describe the problem and related work in greater detail. Then, in Section 3, we describe the implementation of the DOVE system to support multimodal interaction for remote collaborative physical tasks. We present the DOVE architecture and describe each component. In Section 4, we describe the gesture recognition component of the system. In Section 5, we present results from an initial user study. Finally, in Section 6, we summarize the paper and discuss future work.

2. PROBLEM DESCRIPTION

In this research, our goal is to develop technologies to support communication through speech and gesture during collaborative physical tasks. In this section, we first describe the nature of collaboration during physical tasks in greater detail; then, we describe several research issues regarding the development of tools to support remote collaboration on physical tasks; finally we provide an overview of our approach to this problem which is followed by a more detailed presentation in subsequent sections of the paper.

2.1 Collaboration on Physical Tasks

It is well known that people communicate using multiple modalities when collaborating on physical tasks in face-to-face settings. People's speech and actions during such tasks are intricately related to the position and dynamics of objects, other people, and activities in the environment (e.g., [1][2][3][13][20]). Conversations focus on identifying target objects, describing actions to be performed on those targets, and acknowledging the success or failure of these actions [11]. As they speak, people use gestures to clarify or enhance their messages. For example, pointing gestures are used to refer to task objects and locations (e.g., "that piece goes over there"). Representational gestures, such as hand shapes and hand movements, are used to represent the form of task objects and the nature of actions to be used with those objects, respectively [1][16]. For example, a speaker might say, "turn the screw," while using a hand gesture to indicate the direction to turn it, or refer to an object as "the odd-shaped one" while indicating the shape of the object with his/her hands.

Previous studies of video systems to support remote collaboration on physical tasks (e.g., [3][4]) have repeatedly observed that remote participants have difficulty communicating because of their inability to gesture or point at objects in the workspace. These communication problems have negative effects on performance, in that remote performance on physical tasks takes longer than performance when the team is co-located. To facilitate remote communication on physical tasks, it is thus necessary to provide a tool that allows remote collaborators to use both speech and gesture in the same way they would do so if co-located.

2.2 Research Issues

The problem of supporting gesture and speech in remote collaboration on physical tasks differs from what has been explored in previous research in at least four ways. First, the majority of previous systems for computer supported cooperative work (CSCW) have paid little attention to supporting group activities that must reference the external spatial environment. Consequently, gestural communication is not explicitly supported by most existing CSCW technologies.

Second, technologies for supporting gesture communication in CSCW must be different from those supporting human-computer interaction (HCI). In HCI, a gesture-based interface, which translates input gestures to coded data, is designed to implement human-computer communication through human-like styles. Humans are in the human computer interaction loop. The gesture recognition system recognizes the predefined gestures. On the other hand, the function of a gestural tool in CSCW systems is to mediate human-human communication. Instead of the human in the loop, we have put the computers into the human communication loop. The role and functions of the computer have been changed.

Third, a gesture tool might ideally have both HCI and CSCW functions. As an HCI tool, gestures can be used as an input device for camera control (pan, tilt, zoom). As a CSCW tool, gestures are intended to communicate meaning to a remote partner. Understanding how gesture recognition should be implemented to facilitate both of these functions is a complex problem that has not previously been addressed.

Fourth, the existing literature provides few theoretical guidelines for developing gesture systems to support CSCW and HCI functions simultaneously. Unlike traditional multimodal human computer interfaces, we know little about gestural communication in remote collaboration. In order to better understand this process better, we need systems to support it. This is a chicken-and-egg problem. The optimal solution is to tightly couple technology development and user studies together, which requires collaboration from a multidisciplinary research team.

2.3 Overview of Current System

As noted above, in collaborative physical tasks speakers use both pointing gestures, which require a view of the target object and a pointing device, and representational gesture,

which require a method of demonstrating information concerning size, distance, method of movement, and the like. Most current technologies to support gesture either enable pointing at remote objects or show a view of the speaker's hands, but not both. The few exceptions (e.g. [8]), require expensive specialized equipment that makes their use impractical for most collaborative work.

We approach the problem by using pen-based gesturing over video stream. Our goal is to devise a system, using desktop PC and Tablet PC platforms, that enables speakers and listeners to produce and interpret both pointing and representational gestures as readily as they do in face-to-face settings. The preliminary idea has been evaluated by implementation of a cursor pointing device [5][6]. User studies concluded that cursor pointing is valuable for collaboration on physical tasks, but that additional gestural support will be required to make performance using video systems as good as performance working side-by-side.

In this research, we support gestural communication for remote collaborative physical tasks using pen-based gestures over video streams. The video stream plays a dual purpose in the proposed paradigm: (a) it establishes remote communication among collaborators, and (b) it provides gestural communication media. The system allows collaborators to share the workspace through video connections. It also provides remote support for gesture by overlaying pen-based gestures over video streams. We develop technology to support recognition of the predefined gestures, freehand drawing, and a combination of the two. We also evaluate the effectiveness of the proposed new paradigm.

3. SYSTEM IMPLEMENTATION

The DOVE system was designed to facilitate gesturing over video within the context of an instructional collaborative physical task where two or more people interact with real objects in the 3D world [17]. In this type of task, one person, whom we will call the "helper", provides instructions to another person, whom we will call the "worker", as the worker performs the task. The worker can directly manipulate task objects and tools, whereas the helper is located at a distance and must rely on a computer-mediated communications system. In our initial experimental setup, pairs are collaborating to build a large toy robot; however, the technology can be generalized to any type of instructional task, such as telemedicine or distance education, in which a remote party needs to refer to physical objects in a workspace.

The DOVE architecture is shown in Figure 2. The workspace is visually shared through video cameras and equipped with tablet PCs, desktop PCs or other handheld devices. Real-time video streams from these cameras are sent to collaborators' computing devices in the workspace. A helper 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 a face-to-face setting. The results are observable by all collaborators on their own monitors. Details of the implementation are discussed in the remainder of this section.

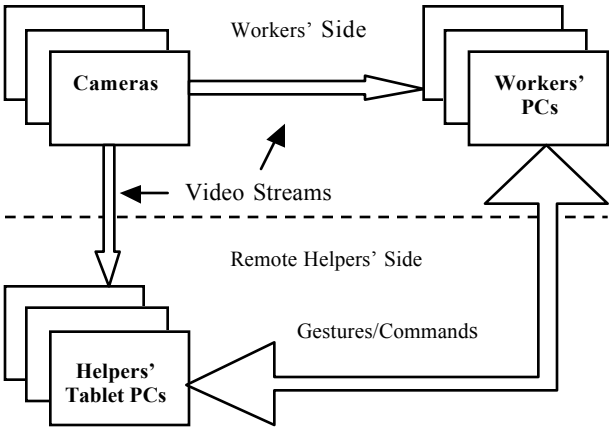

Figure 2. Overview of the DOVE system architecture.

3.1 Camera

There are two alternatives for cameras within our architecture: ordinary video cameras and network IP cameras. With ordinary video cameras, an additional computer has to be used as a 'video server'. The server handles video communications among computing devices within the workspace. Although there is no network delay between cameras and the video server, the streams may jam the server's outgoing network traffic, which causes delay and provides incoherent images to the recipients.

Because of these problems with ordinary video cameras, DOVE incorporates network IP cameras, which are inherent servers, to solve the problem of distributing network traffic (Figure 3). Each network IP camera is a server and connected to the network independently; other computers on the network can be its clients. Once started, a network IP camera opens a TCP/IP port and waits for its clients. When a connection is established, the server's status message and the client's authentication messages will be exchanged. If the client is authenticated, video data will be sent in JPEG format upon a client's image request message. By using this technique, the video flow and process overhead is shared by all network IP cameras.

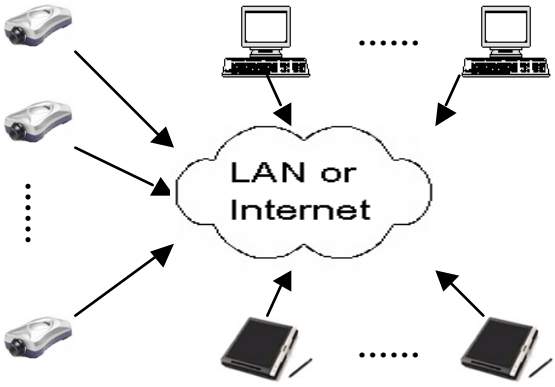

Figure 3. The architecture of using network IP cameras. Each camera is its own server.

3.2 Communication among Collaborators' Computing Devices

After connecting to network IP cameras, the communication among collaborators' computing devices is also in client-server mode. For example, the worker's computer can be a server and the helper's computer can be clients. A socket is created on the worker's computer. It waits and accepts client sockets from the helper's computer. After the establishment of a connection, a helper can send remote gestures and commands through socket communication, or vice versa. The trajectories of freehand drawing and gesture recognition results are observable on all collaborators' monitors.

3.3 Freehand Drawing and Gesture Recognition/Fitting Mode

Pen-based gesture and freehand drawing consist of sequences of points. Each sequence starts from the pen touching the screen and ends when the pen is lifted. When the helper is drawing, the sequence of points will be added to a link list of the current gesture and sent to the workers' computers simultaneously. While drawing, the helper can choose among freehand drawing, gesture recognition, or drawing normalization. In freehand drawing, what is sketched will be shown exactly as drawn on the screen. In gesture recognition mode, a predefined gesture will be recognized and a certain command will be executed. In drawing 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 recognition module recognizes the shape that the user is trying to draw (e.g., arrow, circle) and returns a set of parameters to approximate the recognized shape. The interface, on the other hand, will use these parameters to synthesize and display the normalized shapes. Details of recognition and gesture fitting will be presented in Section 4. There are several parameters that a user can set for sketching, including pen width and color of the drawing.

3.4 Gestures over Video Streams

Since we want to overlay gestures over video streams and display them together, we need two running threads: one is for video communication, the other is for gesture communication. Because they are concurrent procedures—i.e., the order of these two threads are undetermined—displaying them directly on the screen will have flashing effect. Therefore, an image buffer is prepared before the ultimate image is displayed. The synthesis process is illustrated in Figure 4.

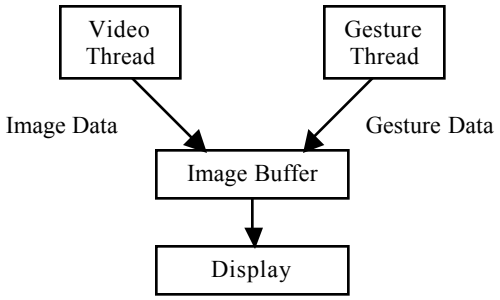

Figure 4. An illustration of gestures over video streams. The output image is first prepared in the image buffer to avoid flashing.

3.5 Commands

The current DOVE prototype provides four sets of commands to the user in addition to the sketching function. First, the system enables users to erase gestures already drawn. Users can choose remove all gestures, the first gesture, or the latest gesture. Second, users can specify an "automatic erase" time interval. After the specified time interval, the gestures disappear from the screen automatically. Third, DOVE provides a set of undo/redo commands. A pair of buttons enables users to undo the last action (i.e., drawing or erasure) or redo what is undone. Inverse action is taken after each undo/redo command as shown in Table 1. Finally, the system provides a "snapshot" function that allows users to save any image appearing on the screen as a JPEG file on the local disk.

Table 1. Actions to Take for Undo/Redo

Last Action	Undo	Redo
Draw a Gesture	Erase Last Gesture	Resume Last Gesture
Erase Last Gesture	Resume Last Gesture	Erase Last Gesture
Erase First Gesture	Resume First Gesture	Erase First Gesture
Erase All Gestures	Resume All Gesture	Erase All Gestures

3.6 Local Area Network

Because jitter is more likely to happen in an Internet environment because of a higher chance of collision, we establish a local area network (LAN) for our preliminary tests of the DOVE system. A wireless router is used to connect network IP cameras, workers', and helper's computers. The devices communicate with each other locally, isolated from the Internet. In this way, we can minimize effects caused network delay. Disruption of remote gestures by network jitter was investigated by Gutwin [7], and is an important issue we will be addressing in future work.

4. GESTURE RECOGNITION/FITTING

Gesture recognition serves three functions in DOVE. First, by recognizing and normalizing shapes such as lines, arrows and circles, gesture recognition is intended to facilitate communication between remote partners. We anticipate that workers will find it easier to discern the meaning of helpers' gestures when those gestures have been normalized. Second, gesture recognition provides a way to reduce amount of data transmitted over the network. Rather than sending the entire freehand point sequence, the system can send parameters of the predefined gestures to the remote machine for creating the normalized images. Third, gesture recognition can be used as a way to input commands to the system itself. For example, a straight arrow could represent a command to move the camera, indicating the direction and length of movement, whereas curved arrows might represent requests to zoom the camera in and out. In this section, we describe a recognition system we developed which can support these functions.

The current version of the software recognizes 12 gestures (Figure 5), selected on the basis of preliminary user studies that indicated that they were the most common freehand drawings during a collaborative construction task. In addition,

we can draw almost arbitrary gestures by combining these 12 gestures together in multiple stroke sequences.

The point sequence is recorded when the pen is being pressed and moved on the screen and recognition is carried out as soon as the pen leaves the screen. These graphs, sequences of points, are called gestures if we relate the corresponding scope and command to each shape [18]. Technically it can be viewed as a graph recognition or classification problem if we do not take their scopes and commands into account.

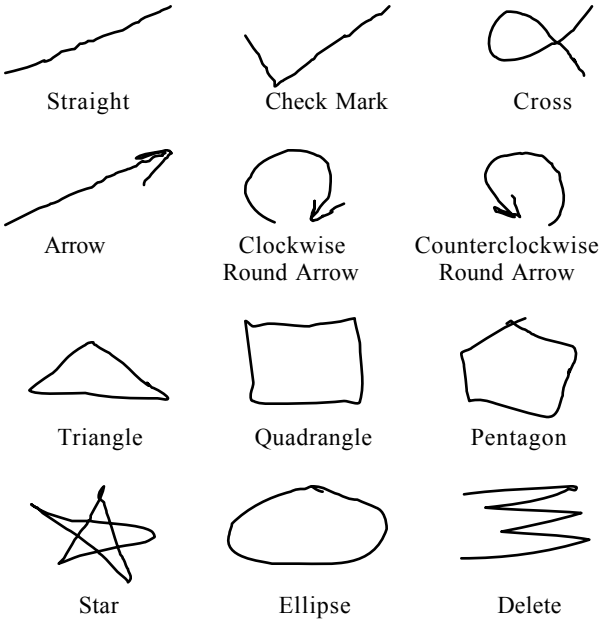

Figure 5. Gestures recognized by the DOVE prototype.

4.1 Related Work

We first review some approaches presented by other researchers. Rubine [18] described a gesture based interface called GRANDMA. GRANDMA specifies single-stroke gestures drawn by mouse movement, beginning with the press of a mouse button. Statistical methods are used to do the recognition. First, thirteen locally and globally geometrical features are extracted to represent the input stroke. Then, the feature vector is classified as one of the C possible gestures via a linear evaluation function. Finally, a closed formula is used to calculate the weights in the function.

Jorge's [10] approach recognizes elementary shapes using a decision tree and fuzzy logic. The recognition process starts from the first pen-down event until a set timeout value after the last pen-up. First, global geometric properties of the input stoke are extracted as features. Second, a decision tree is applied to filter out unwanted shapes using distinctive criteria. Third, fuzzy logic is used to associate degrees of certainty to recognized shapes.

Jin et al. [9] proposed an on-line sketchy graphics recognition algorithm. There are four pre-processing steps. First, it removes redundant intermediate points using polygonal approximation. Second, agglomerate points filtering is employed to reduce hooklets at the end of the lines and circlets at the turning corners. Third, end point refinement is used to delete extra points for a self-crossed stroke and extend endpoints for an open stroke. Fourth, convex hull is calculated to select n vertexes to represent the original line. After the pre-processing, m points from the original n vertexes are selected with a recursive vertex combination algorithm. The closed-shape graph is classified according to the number m.

In contrast to the sophisticated methods described above, DOVE employs a rather simple but effective algorithm to recognize the pre-defined gestures.

4.2 Preprocessing

Like many other pattern recognition tasks, preprocessing is necessary for enhancing robustness and recognition accuracy. We have performed two different preprocessing techniques before the feature extraction.

A user will draw a gesture at different speeds; thus the sampling rate for the same gesture is not a constant, i.e., for a given period of time, the number of samples is changeable. Several methods can be used for this task, such as linear, B-spline, Bezier interpolation, etc. In the current system we apply linear interpolation before resampling the sequence of points.

Contrary to [9], which states there are hooklet-like segments at the end of the sketchy lines, we found that hooklets are more likely to happen at the beginning of gestures. That is, right before the pen touches the point where a user wants to start, it is much likely to sketch over the screen and create a hook. Therefore, if we find a sharp curvature change after a few points from the start, we remove those points, as shown in Figure 6.

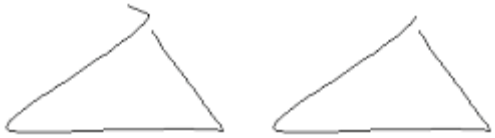

Figure 6. An example of removing the hook at the beginning of the gesture.

4.3 Vertex and Curve Detection

After preprocessing, DOVE performs a two-pass scan to detect vertexes and curves. In the first pass, we use a small window and record the slope difference between ith and (i+*window*)th slice, defined as the connection between two sequential points. When the absolute value of slope change exceeds a threshold, it corresponds to a vertex. The slope changes for a quadrangle are shown in Figure 7. If no vertexes are detected within a line segment, we perform a second pass scan with a larger window to see if it is a curve, as shown in Figure 8.

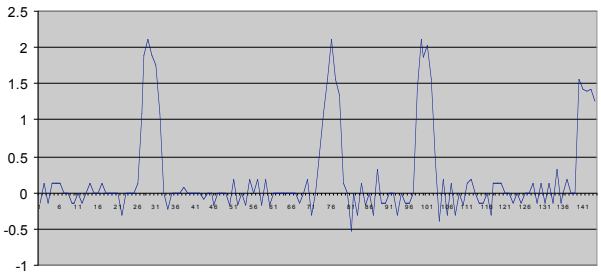

Figure 7. Slope changes within a small window for a quadrangle. Four vertexes were detected.

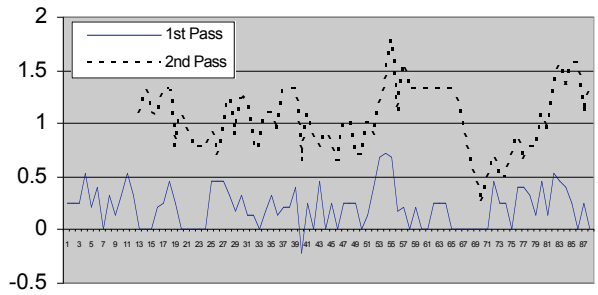

Figure 8. Results of a two-pass scan for an ellipse. In the first pass, no vertexes were detected. In the second pass with a larger window, curves were detected.

There are three advantages to our approach. First, it can be done efficiently in *O(n)* time, where *n* is the number of points after pre-processing. Second, it is insensitive to the rotation of the gestures and the speed of current drawing—i.e., it allows users to sketch in a flexible manner. Third, the detected vertexes form the basic components for shape fitting, as described in the next section. Furthermore, the sign of the slope difference indicates whether a curve is drawn in a clockwise or a counter-clockwise direction.

4.4 Classification Rules

When vertexes and curves are found, DOVE measures the distance between the first and the last points to see whether the gesture is closed. If it is closed, a vertex will be added at the last point. Finally, a set of rules pertinent to the 12 gestures are applied to classify the input stoke (Table 2).

Table 2. Classification Rules

Open Gestures	
Straight Line	Has no vertexes or curves
Check Mark	Has only one vertex
Cross	Has no more than two vertexes, and a segment of curve
Arrow	Begins with a straight line and has three vertexes in the latter part
Round Arrow	Begins with a curve and has three vertexes in the latter part
Delete	Has more than three vertexes
Closed Gestures	
Triangle	Has three vertexes
Quadrangle	Has four vertexes
Pentagon	Has five vertexes and the average slope changes at vertexes is lower than a threshold
Star	Has five vertexes and the average slope changes at vertexes is higher than a threshold
Ellipse	Has fewer than two vertexes

4.5 Gesture Fitting

To better support human gestural communication, DOVE provides a gesture fitting tool. DOVE recognizes the intentional shape a user is drawing and regularizes it. An example from our robot building task is shown in Figure 9. A helper used free-hand drawing to specify a robot and its movement. The ellipse and arrow were regularized when the pen left the screen. The output was seen by all participants.

Figure 9. Example of gesture fitting in DOVE

We would like users to be able to draw arbitrary sketches, not restricted by the graphs we can fit. While many free hand sketching interfaces aim at accurate approximation to the input strokes (e.g. [19]), we feel it is unrealistic in our task if parts or all of a single gesture are potentially free hand drawings. For example, a remote helper might want to designate a path of movement that consists of both free hand drawings and recognized gestures. To address this situation, DOVE combines gesture fitting with freehand drawing.

A single gesture is segmented by the vertexes we have detected. We verify a line segment using the most intuitive way. If the distance between their end points is shorter than the original length to some extend, we reject it. Otherwise we connect them with a straight line. Instead of trying to approximate the curves, which may be intractable, we leave them as free hand drawings connected with other line segments (either recognized gestures or free hand drawings).

4.6 Recognition Results

In order to test gesture recognition accuracy, we evaluated the DOVE gesture recognition system. We asked 12 participants to make each of the 12 gestures eight times, for a total of 1146 gestures after missing data was excluded. The overall accuracy of the system in identifying the gestures was 91.01%. Results of individual gestures range from 75.5% to 100%, as shown in Table 3. An analysis of error sources showed that quadrangles and pentagons had the highest error rate because users often created "round angles" for them when they drew very fast, causing our algorithm fail to detect the true vertexes. We are currently exploring solutions to this problem.

Table 3. Accuracy of individual gestures

Gestures	Accuracy	Gestures	Accuracy
Straight Line	100%	Check Mark	98.9%
Cross	94.8%	Arrow	92.7%
Round Arrow A	86.3%	Round Arrow B	82.3%
Triangle	97.9%	Quadrangle	75.5%
Ellipse	93.7%	Delete	97.9%
Pentagon	79.2%	Star	92.7%

5. INITIAL USER STUDY

Our initial user study focused on assessing the value of DOVE for supporting remote collaboration on physical tasks (see [6] for more detail). Previous work [4] has suggested that scene-oriented video cameras, such as network IP cameras used in our system, enhance communication over audio-only but do not allow partners to communicate as easily as they do when face-to-face. When a cursor-pointer is added to the video system, pairs report that it is easier to communicate but their performance times are no better than with the scene camera alone [5][6]. We hypothesized that DOVE's drawing component would improve communication over a scene camera alone because it allows helpers to display a full range of pointing and representational gestures. We tested this hypothesis by asking pairs of students to perform a collaborative physical task with and without the gesture tool.

5.1 Method

Pairs (N = 28) of undergraduate students collaborated to build a large toy robot (Figure 10). Each pair completed three robot assembly tasks (e.g., left arm, right foot) under three different media conditions: (a) *Scene camera only*: helper could view the output of the IP camera focused on the workers task environment, but could not manipulate the video feed; (b) *DOVE + manual erasure of gestures*: helper could draw on the video feed but had to manually erase their gestures; and (c) *DOVE + automatic erasure of gestures:* helpers could draw on the video feed and the gestures faded out after 3 seconds. We measured performance times in each condition.

Figure 10. Robot used in the user studies.

5.2 Results

As hypothesized, performance was significantly faster with the DOVE system than with the network IP camera alone (see Figure 11). Performance was also faster with the auto-erase function enabled than when helpers had to manually erase their gestures. A 3 (task) by 3 (order) by 3 (condition) Analysis of Variance indicted significant effects of condition ($F [2, 29] = 8.24$, $p = .002$) and task ($F [2, 29] = 34.93$, $p < .0001$). Post-hoc comparisons indicated that the Auto Erase condition was significantly faster than both the Manual Erase condition ($t [29] = -2.28$, $p < .05$) and the video-only condition ($t [29] = -4.04$, $p < .001$). The difference between the Manual Erase and Video Only conditions was in the expected direction but nonsignificant ($t [29] = -1.56$, $p = .13$).

Inspection of video recordings of each session indicated that helpers used a wide range of gestures while providing instructions (examples are shown in Figure 12). Preliminary coding of drawings suggested that the majority were intended to point out objects or locations in the work space, with the remainder intended to demonstrate the orientation of insertion or overall positioning of task elements.

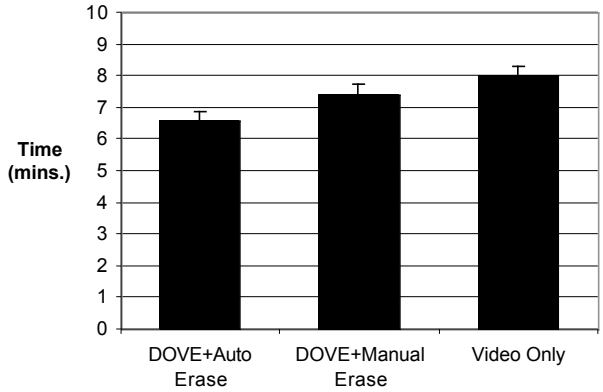

Figure 11. Performance time by media condition.

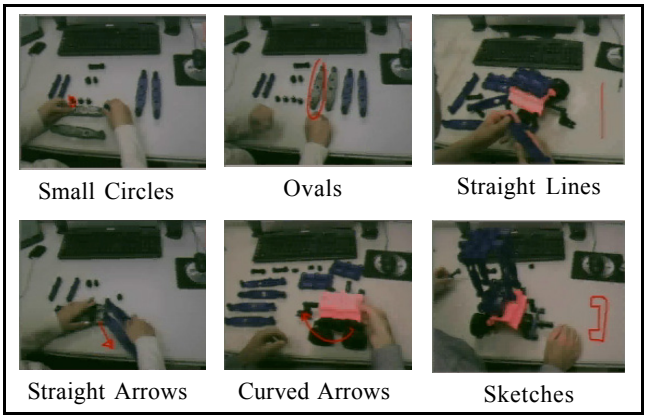

Figure 12. Sample of gestures created by participants using DOVE during a robot assembly task.

5.3 Discussion

The preliminary results from our user study strongly suggest that remote collaborators on physical tasks benefit from having a tool that allows them to draw gestures over a video feed of the workspace. The DOVE drawing tool significantly reduced performance time compared to the camera alone. Our findings further suggest that collaborators perform best when the gestures disappear automatically, much like ordinary hand gestures disappear once people have completed them. However, further analysis of helpers' drawings are required to understand whether automatic erasure would be beneficial in all situations. We observed that in some cases, the helpers used the pen tool to draw complex diagrams of task instructions involving multiple pen strokes. This suggests that the choice of automatic vs. manual erasure might best be left to user discretion rather than set by the system itself.

We are currently collecting data from additional pairs performing a different type of collaborative physical task to ensure the generalizability of our findings. In addition, we are assessing the added value of DOVE's gesture recognition component for remote collaboration on physical tasks.

6. CONCLUSIONS

We have proposed to support multimodal interaction for remote collaborative physical tasks using gestural communication over video streams. The proposed concept enables speakers and listeners to produce and interpret both pointing and representational gestures as readily as they do in face-to-face settings. We have implemented the DOVE system, comprised of video cameras, tablet PCs, and desktop PCs, to demonstrate the proposed concept. DOVE allows collaborators to share the workspace through video connections. It also provides remote support for gesture by overlaying pen-based gestures on video streams. DOVE supports not only recognition of the predefined gestures, but also freehand drawing and a combination of freehand drawing and normalization of freehand drawing. We have tested the accuracy of gesture recognition component and evaluated the value of DOVE's freehand drawing component within the context of a collaborative physical task.

Results of a preliminary user study have indicated that the DOVE gesture over video communication system enhances task performance over traditional video-only systems. We are currently extending our existing system to more platforms such as PDAs. We will evaluate other benefits of the proposed system in different applications such as camera control and two way communication. We are also performing further user studies to help guide future system development.

7. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. 9980013 and 0208903. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We thank Katherine Bankson, Angela Ho, Marina Kobayashi, Kathy Lau, Asa Sherrill, Katherine Waite, and Jim Zheng for their assistance running the user studies and preparing data for analysis.

8. 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. NY: ACM Press.

[2] Flor, N. V. (1998). Side-by-side collaboration: A case study. *International Journal of Human-Computer Studies, 49*, 201-222.

[3] Fussell, S. R., Kraut, R. E., & Siegel, J. (2000). Coordination of communication: Effects of shared visual context on collaborative work. *Proceedings of CSCW 2000* (pp. 21-30). NY: ACM Press.

[4] 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.* NY: ACM Press.

[5] Fussell, S., Setlock, L., Parker, E., & Yang, J. (2003). Assessing the value of a cursor pointing device for remote collaboration on physical tasks. *Proceedings of CHI '2003*. NY: ACM Press.

[6] Fussell, S. R., Setlock, L. D., Yang, J., Ou, J., & Parker, E. M. (under review). Gestures over video streams to support remote collaboration on physical tasks.

[7] Gutwin, C., & Penner, R. (2002). Improving interpretation of remote gestures with telepointer traces. *Proceedings of CSCW 2002*. (pp.49-57). NY: ACM Press.

[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] Jin, X., Liu, W., Sun, J., & Sun, Z. (2002). On-line Graphics Recognition. *Proceedings of PG'02* (pp. 256-265).

[10] Jorge, J., & Fonseca, M. (1999). A Simple Approach to Recognise Geometric Shapes Interactively. *Proceedings of GREC'99* (pp. 266-276).

[11] Kraut, R. E., Fussell, S. R., & Siegel, J. (2003). Visual information as a conversational resource in collaborative physical tasks. *Human-Computer Interaction, 18,* 13-49.

[12] Kraut, R. E., Miller, M. D., & Siegel, J. (1996) Collaboration in performance of physical tasks: Effects on outcomes and communication, *Proceedings of CSCW'96* (57-66). NY: ACM Press.

[13] Kuzuoka, H., & Shoji, H. (1994). Results of observational studies of spatial workspace collaboration. *Electronics and Communications in Japan, 77*, 58-68.

[14] Kuzuoka, H., Kosuge, T., & Tanaka, K. (1994) GestureCam: A video communication system for sympathetic remote collaboration. *Proceedings of CSCW 94* (35-43). NY: ACM Press.

[15] 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). NY: ACM Press

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

[17] Ou, J., Chen, X., Fussell, S. R., & Yang, J. (in press). DOVE: Drawing over Video Environment. *Multimedia 2003 Demonstrations.*

[18] Rubine, D. (1991). Specifying gestures by example. *Computer Graphics, 25*, 329-337.

[19] Sezgin, M., Stahovich, T., & Davis, R. (2001). Sketch based interfaces: Early processing for sketch understanding. *Proceedings of PUI-2001*. NY: ACM Press.

[20] Tang, J. C. (1991). Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies, 34*, 143-160.