Human-Computer Interaction Based Only on Auditory and Visual Information

Hui Sha and Arvin Agah

Abstract: One of the research objectives in the area of multimedia human-computer interaction is the application of artificial intelligence and robotics technologies to the development of computer interfaces. This involves utilizing many forms of media, integrating speech input, natural language, graphics, hand pointing gestures, and other methods for interactive dialogues. Although current human-computer communication methods include computer keyboards, mice, and other traditional devices, the two basic ways by which people communicate with each other are voice and gesture. This paper reports on research focusing on the development of an intelligent multimedia interface system modeled based on the manner in which people communicate. This work explores the interaction between humans and computers based only on the processing of speech (words uttered by the person) and processing of images (hand pointing gestures). The purpose of the interface is to control a pan/tilt camera to point it to a location specified by the user through utterance of words and pointing of the hand. The system utilizes another stationary camera to capture images of the user's hand and a microphone to capture the user's words. Upon processing of the images and sounds, the system responds by pointing the camera. Initially, the interface uses hand pointing to locate the general position which user is referring to, and then the interface uses voice command provided by user to fine-tune the location, and change the zooming of the camera, if requested. The image of the location is captured by the pan/tilt camera and sent to a color TV monitor to be displayed. This type of system has applications in tele-conferencing and other remote operations, where the system must respond to a user's command, in a manner similar to how the user would communicate with another person. The advantage of this approach is the elimination of the traditional input devices that the user must utilize in order to control a pan/tilt camera, replacing them with more "natural" means of interaction. A number of experiments were performed to evaluate the interface system with respect to its accuracy, efficiency, reliability, and limitation.

Keywords: Human-computer interactions, intelligent interfaces, multimedia systems, speech processing, image processing

I. Introduction

1. Motivations and applications

During the entire process of human-computer interface design, the objective is to achieve a more natural way of communication between the human and the computer. Recent advances have given rise to a number of novel human-computer interaction (HCI) multimedia systems, with the capability to process speech (Rabiner *et al.*, 1993), gestures, etc. Multimedia interface research is an essential component of human-computer interaction (Sharma *et al.*, 1998), covering issues that include:

- How can human-computer interaction be made clearer and more efficient?
- How can interfaces offer better support for their users' tasks, plans, and goals?
- How can information be presented more effectively?
- How can the design and implementation of good interfaces be made easier?

In face-to-face conversations, humans frequently use dedicated gestures (e.g. the index finger points at an object), parallel to verbal descriptions for referent identification. Such multimedia forms of communication are of great importance for intelligent interfaces (Sullivan and Tyler, 1991) and they can improve human interactions with computers and machines, because they simplify and speed up reference to objects in a visual context. The reason is the fact that such models are necessary prerequisites in order for a system to be capable of exhibiting a wide range of intelligent and cooperative dialogue behaviors. These models are required for identifying the location to which the dialogue partner is referring. Analogously, in HCI area, it means that the computer is required to identify the location to which the human user is referring. Successful embodiment of these modalities into an interface has the potential of easing the HCI bottleneck (Pavlovic *et al.*, 1997) that has become noticeable with the advances in computing and communication. It has also become increasingly evident that the difficulties encountered in the analysis and interpretation of individual sensing modalities may be overcome by integrating them into a multimedia human-computer interface. Therefore, it is important to incorporate naturalness in the design of interfaces, with the capability of handling continuous natural gesture and speech inputs. The use of features from both speech and hand gesture results in a multimedia interface where different forms of media complement each other in a more "human and natural" communication style.

The interface program reported on in this paper allows a human user to operate and control a pan/tilt camera completely, by only speaking a natural language and using hand gestures. In terms of ease of use, efficiency, and user-friendliness, the interface system can provide an ideal way to communicate with the computer. The multimedia interface system can be expanded to other application areas; for example, a user can use this interface technology to command a robot to move to a specific location to perform certain tasks for the user. Video-conferencing is now

Manuscript received: June 27, 2000., Accepted: Nov. 6, 2000.

Hui Sha: Tellabs, Lisle, Illinois U.S.A.
Arvin Agah: Department of Electrical Engineering and Computer Science The University of Kansas, Lawrence, Kansas 66045 U.S.A.

* All correspondence should be addressed to Arvin Agah.

* Work performed while Hui Sha was at the University of Kansas.

※ This work was supported in parts by a grant from The University of Kansas General Research Fund (KU GRF) of Summer 2000.

another popular area both in research and in application. The developed interface program is very applicable to video-conferencing. During a presentation, the presenters can use the system to operate the pan/tilt camera to show a specific slide or some interesting object to audience at the other end. In addition, this work can be applied to design of smart rooms and how people can interact with the rooms in an effective and productive manner.

2. Research methodology

In this project, a user interface is developed, using Microsoft Visual C++ (Microsoft, 1999), which interacts with the user through speech and gestures, or using a graphical user interface for debugging and training purposes. In the graphical user interface dialog box design several standard controls are used, such as *push button*, *check box*, *list box*, *edit box*, and *static control* (McGregor, 1996) (Schildt, 1998) (Yao and Yao, 1995). The system uses the DragonDictate speech recognition software (Dragon Systems, 1999) in order to implement the functionality of recognizing the voice commands uttered by user. A Cognachrome vision board (Newton, 1999) and a PULNiX TMC-7 series color camera (PULNiX, 1999) are used to capture and process the images of user's hand gestures. Combining the voice and hand pointing as inputs to the interface, the user can continuously control the movements of a Sony EVI-D30 pan/tilt camera (Sony, 1999). The control of the pan/tilt camera is done without the computer keyboard or mouse being involved. Initially, the person uses a hand gesture to points to a general direction that is preferred. Next, the pan/tilt camera points to the general location, and then the user can fine-tune the position by using voice commands such as *up, down, left, more, closer*, etc. Using voice commands, the user can also pause and resume the operations of the interface or completely exit the interface. A series of tests were performed to measure the performance of the interface system based on its accuracy, efficiency, reliability, and function restriction.

3. Paper organization

This paper is organized into five sections. After the introduction in Section 1, background and related work in the areas of multimedia human-computer interface design and application are presented in Section 2. Section 3 describes the multimedia interface system including the hardware devices and the designed software. Section 4 discusses the experimental setup, initial calibration and training of the system for the users, and the experimental results. In Section 5, the contributions are presented, followed by the limitations and the future work of this approach.

II. Background and related work

1. Human-computer interactions

There is currently no agreed upon definition of the full range of the research area of human-computer interaction. A general working definition is (ACM SIGCHI, 1999): "HCI (human-computer interaction) is the study of how people interact with computers and to what extent computers are or are not developed for successful interaction with human beings". Some of the special concerns in the research area of human-computer interactions are:

- The joint performance of tasks by humans and computers
- The structure of communication between humans and computers
- Human capabilities to use computers, including the use of interfaces
- Algorithms and programming of the interface
- Engineering aspects of designing and building interfaces
- The process of specification, design, and implementation of interfaces

Because human-computer interaction research focuses on humans and computers in communication, it draws from supporting knowledge on both the computer and the human side. On the computer side, techniques in computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social science, cognitive psychology, and human performance are relevant. Moreover, engineering and design methods are relevant. Interrelated aspects of human-computer interaction are the nature of human-computer interaction, the use and context of computers, human characteristics, computer system, interface architecture, and the development process. A variety of technologies have been developed for supporting interactions with humans, namely, the input and output devices connecting the human and the computer (Jacob, 1999). These tools are used in a variety of techniques for organizing a dialogue, whereas complex dialogues lead to considerations of the systems' architecture to support features such as interconnection of application programs, windowing, real-time response, and multi-tasking of dialogue objects.

2. Intelligent multimedia user interfaces

The growing emphasis on multimedia user interface design is fundamentally inspired by the aim to support natural, flexible, efficient, and powerfully expressive means of human-computer interactions that are easy to learn and use. Since the advent of the computers, the users have been forced to conform to the interfaces dictated by the computers. The arrival of interactive text terminals in the 1970s was a noticeable leap, but soon, even typing was seen as a burden, and a more efficient interface was developed. Graphical operating systems of the 1980s introduced the mouse, a simple pointing device for the user. In the 1990s, with increases in computational power, basic speech recognition and pen-based computing had become a reality. All such interfaces are efficient for a trained user. However, they are typically inefficient as human-centric forms of communication. With recent advances in human-computer interface technologies, intelligent interaction has become more feasible to create interfaces that resemble forms of human-computer communication.

One aspect of intelligent multimedia user interfaces is devoted to the application of artificial intelligence and robotics technologies to the development of human-computer interface systems. The interface is the means by which a user communicates with a system, whether to get it to perform some function or computation directly, to find and deliver information, or to provide ways of interacting with other people. The three layers of communications are:

1) *Language Layer*. Natural languages include complex syntax and complex semantics (e.g., whatever a human can say). Restricted verbal languages consist of limited syntax and constrained semantics (e.g., operating systems command language). Direct manipulation languages include objects that are noun-like (e.g., verb equivalents from manipulations).

2) *Expression Layer*. Statements in most of the above languages can be expressed using different types of realizations, including: speaking (continuous speech recognition, isolated-word speech recognition); writing (typing on a keyboard, handwriting); drawing; gesturing (e.g., American Sign Language); pick-from-set (various forms of menus); pointing; clicking; dragging; three-dimensional manipulations (e.g., stretching, rotating); manipulations within a virtual reality environment (three dimensions with broader field of view); and manipulations unique to virtual reality environment (e.g., locomotion or flying through/over things as a means of manipulating them or at least looking at them).

3) *Devices*. Hardware mechanisms provide a way to express a statement. Again, more than one technology at this layer can be used to implement items at the layers above. These include keyboards (many different kinds of typing); microphones; light pen/drawing pads, touch-sensitive screens, whiteboards; video display screens and mice; video display screens and keypads (e.g., automated teller machine); touch-sensitive screens (touch with pen, touch with finger); telephones (audible menu with keypad and/or speech input); push-button interfaces with different buttons for each choice (like buttons on an appliance); joystick; and virtual reality input gear (glove, helmet, suit, body position detectors, etc.).

Although it is still impossible to create an interface that can handle all forms of human communication, it is possible to create a small multimedia subset (Bradski *et al*., 1998). Research and development of artificial intelligence (AI) multimedia systems for human-computer interfaces have focused on natural language text, speech, and graphics primarily in isolation, rather than in integrated interfaces. Some related work has been done in this area. Certain works rely on single mode of interaction such as hand gesture (Baudel and Baudouin-Lafon, 1993) (Segen and Kumar, 1998) (Lamar *et al*., 1998) (Morguel and Lang, 1998). Recently, other works have utilized multiple modes, such as using speech with other communication methods (Vo and Wood, 1999). Some research efforts combine hand gesture and speech recognition (Buxton, 1999) (Poddar *et al*., 1998) (Bradski *et al*., 1998). They integrate speech input, graphics and pointing gestures for interactive dialogues between humans and computers. These dialogues are modeled on the manner in which two people naturally communicate in coordinated multiple modalities when working at a graphics device.

In the domain of intelligent multimedia human-computer interface design, the interaction modalities includes three or six-degree-of-freedom techniques, gesture interfaces, voice input, audio output (both speech and non-speech audio), touch/touch-screen, two-handed interaction, multi-modal interaction, head tracking, and ubiquitous computing. The individual modalities for HCI can be classified under two categories of human-action modalities and computer-sensing modalities (Pavlovic *et al*., 1997). Multiple human actions, such as facial expressions and hand or eye movement, can be sensed through the same devices and used to infer different information. Figure 1 depicts how the two categories relate to each other. A particular human-action modality (e.g., speaking) may be interpreted using more than one computer-sensing modality (e.g., audio and video). One research project provided the notion that voice and gesture inputs at the graphics interface can converge to provide a natural user modality (Bradski *et al*., 1998). The work involved the user commanding simple shapes about a large screen graphics display surface. Since voice could be added to simultaneous pointing, the free usage of pronouns was possible. Thus, gesture combined with voice enabled the system to understand the user while referring to a location (Bradski *et al*., 1998).

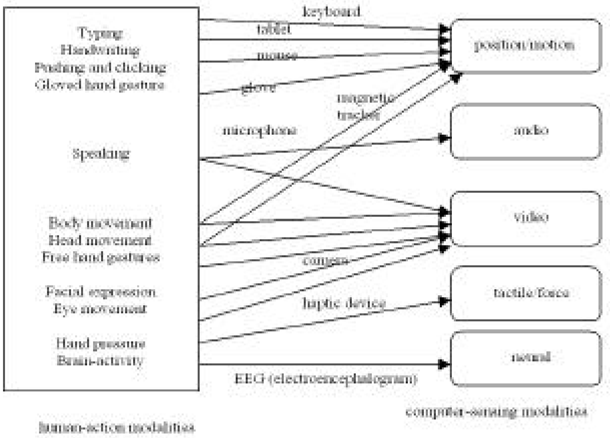


Fig. 1. Mapping between human-action modalities and computer-sensing modalities.

3. Image processing and hand tracking

Classes of techniques based on the movement of the human arm and hand, or hand gestures have been the focus of an area of research for recent years. Human hand gestures are the means of non-verbal interaction among people. They range from simple actions of using a hand to point at and move objects around, to the more complex ones that express our feelings and allow us to communicate with others (Poddar *et al*., 1998). In order to exploit the use of gestures, in HCI it is necessary to provide the means by which they can be interpreted by computers. First attempts to solve the problem resulted in mechanical devices that directly measured hand angles and spatial position. This group is best represented by the so-called glove-based devices (Fels and Hinton, 1993) (Berry *et al*., 1998). Glove-based gesture interfaces require the user to wear a somewhat cumbersome device, and generally carry a load of cables that connect the device to a computer. This hinders the ease and naturalness with which the user can interact with the computer-controlled environment. This had spawned active research efforts toward the design and development of more natural HCI techniques.

Potentially, any awkwardness in using gloves and other devices can be overcome by using video-based non-contact interaction techniques. This type of approach suggests using systems of video cameras and computer vision techniques to interpret gestures. Other factors that may have contributed to this increased interest include the availability of fast computing--making real-time vision processing feasible--combined with recent advances in computer vision techniques. Many of those approaches have been chosen and implemented to focus on one particular aspect of gesture, such as hand tracking, hand posture estimation, or hand pose classification. One such example is using gesture recognition control of a virtual environment research testbed (Berry *et al*., 1998). In another system, visual images of gestures were acquired by one or more video cameras, and processed in the analysis stage where the gesture model parameters were estimated (Pavlovic *et al*., 1997). Using the estimated parameters and some higher level knowledge, the observed gestures were inferred in the recognition stage. Recognition may also influence the analysis stage by predicting the gesture model at the next time instance. A major motivation for the reported studies on gesture recognition is the potential to use hand gestures in various applications, aiming at a natural interaction between the human users and various computer-controlled devices and displays.

4. Speech processing

As computer power grows each year, the boundary of the man-computer interface can move from interaction that is native to the computer toward communication that is natural to human, that is, speech recognition. Tremendous progress had been made in speech recognition, and several commercially successful speech interfaces have been deployed (Mane *et al*., 1996) (Rozmovits, 1996) (Feldman, 1999). Speech recognition and processing concepts, technologies, and products include:

- *Speech Synthesis* is the creation of speech electronically. In other words, making a computer talk.
- *Speech Recognition* is computer comprehension of speech. There are several broad classifications of speech recognition, including discrete speech versus continuous speech, speaker-dependent versus speaker-independent, and context-sensitive versus context-insensitive.
- *Discrete Speech Recognition* requires that each word be an individually identifiable unit. During normal speech, words are run together, and even slurred. To make speech recognition easier, many systems require a pause between words, with a typical requirement of 100 milliseconds.
- *Continuous Speech Recognition* is the ability to recognize words exactly as they are spoken, including slurs. One technology based on Hidden Markov Model separates the words into phonemes (individual sounds), and then reassembles them into words.
- *Speaker-dependent* systems are trained for a single voice. The system is trained to understand the user's pronunciations, inflections, and accents, and can run much more efficiently and accurately because it is tailored to the speaker.
- *Speaker-independent* systems are designed to deal with any speaker. They need to determine what parts of speech are generic, and which ones vary from person to person. A spin-off of this speech recognition technology is that the speaker-dependent parts have now been programmed into security systems, which are designed to respond only to a specific individual's voice.
- *Context-sensitive* systems increase their accuracy by anticipating or limiting what can be said at any given time. Context-sensitive systems may actually have large vocabularies, but only a small portion of that vocabulary will be activated at a time.
- *Context-insensitive* systems allow the user to say anything, at any time. Typically, they have dictionaries in the neighborhood of 20,000 words.
- *Speech recognition* engines are comprised of PC dictation and PC recognition engines.
- *PC Dictation* systems include DragonDictate, a dictation engine that can be adapted for specific vocabularies (Dragon Systems, 1999); IBM ViaVoice for PC-based dictation applications (IBM, 1999); and FreeSpeech98, a continuous speech dictation package from Philips Speech Processing for Windows PC (FreeSpeech98, 1999).
- *PC Recognition Engines* include the AT&T Advanced Speech products group's WATSON, with SAPI-compatible speech recognition and speech synthesis, as well as speaker verification technologies (AT&T, 1999); continuous speech recognition technology used for OEM software on the PC platform and Listen for windows by Verbex Voice Systems, and utilized by the UPS, L'Eggs, Canada Post, and Medical Labs (Verbex, 1999); and Apple's PlainTalk that incorporates speech recognition and synthesis into the Mac Operating System (Noon, 1999).

III. Intelligent multimedia interface system

This section introduces the intelligent multimedia interface system, the hardware devices, and the developed software used in the project. The system integration and implementation are also covered in this section.

1. System overview

The interface system provides a continuous human-computer interaction using hand pointing and voice commands using a personal computer running Windows NT 4.0 (Microsoft, 1999), with the specifications listed in Table

1. Microsoft Visual C++ 6.0 (Yao and Leinecher, 1997) was used to develop an interface, which interacted with the user through one or more modalities. A PULNiX CCD color camera (PULNiX, 1999) was used to capture the images of the user's hand. A Cognachrome vision board with its ARC development environment (Newton, 1999) were used to provide the functionality of processing the images of the user's hand. The DragonDictate speech recognition software (Dragon Systems, 1999) and a microphone were used to implement the functionality of processing the voice commands uttered by the user. A pan/tilt color video camera (Sony, 1999) was used to show the image of the location to which the user was referring. Combining the voice and hand pointing as inputs of the interface, the user could continuously operate the pan/tilt camera--without the use of the keyboard or computer mouse. The system was started using a voice command, followed by the user pointing his/her hand to specify the general direction that was preferred. After that, the user could fine-tune or modify the precise location using voice commands. In addition to the control of the pan and tilt movements of the camera, using this interface system, the user could also adjust the speed of the camera movements and the zooming of the camera using voice commands. The interface program provided a continuous interaction between the user and the computer. However, to allow the user to pause and resume the operation conveniently, additional voice commands were provided. The system could also be terminated using a voice command. The complete interface system was the result of the proper integration of a number of interacting modules, as displayed in Figure 2.

Table 1. Specifications for the personal computer component of the interface system.

SYSTEM COMPONENT	SETTINGS
Operating System	Windows NT 4.0
CPU	Intel Pentium II 266MHz
RAM	64,948 KB
Video card	Matrox Graphics Millennium II, memory size: 4MB
Audio card	16-bit audio

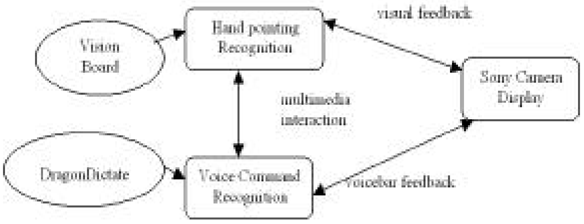


Fig. 2. Interacting modules of the multimedia user interface system.

2. Image processing

2.1 The CCD camera

The capturing of the images of the users' hand gestures was done using a PULNiX TMC-7 CCD (Charge Coupled Device) color video camera (PULNiX, 1999). A number of specifications of the camera are listed in Table 2, and a photograph of the camera is included in Figure 3.

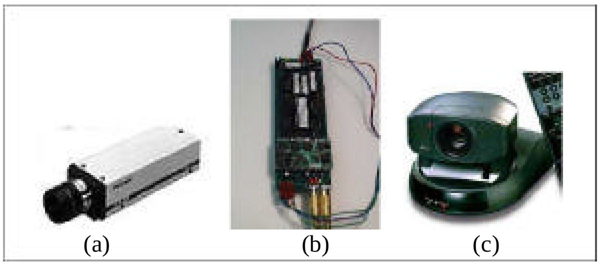


Fig. 3. Photographs of (a) PULNiX TMC-7 CCD color video camera (PULNiX, 1999), (b) Newton Research Labs Cognachrome vision board (Newton, 1999), and (c) Sony EVI-D30 pan/tilt color video camera (Sony, 1999).

Table 2. PULNiX TMC-7 CCD camera specifications (PULNiX, 1999).

SPECIFICATION	VALUE
Imager	0.5 inch interline transfer CCD
Pixels	768 (H)×496 (V)
Cell size	8.4 (H)×9.8 (v) microns
Sensing area	6.41 (H)×4.89 (V) mm
Scanning	525 lines, 2:1 interlace
TV resolution	570 (H)×485 (V) lines
Power requirement	DC 12V, 2.5W
Dimensions	46mm (W)×40mm (H)×61.3mm (L)

2.2. Vision board

This project used a Cognachrome vision system 2000, developed by Newton Research Labs (Newton, 1999), which is a small video-processing computer system with special hardware to track multiple objects in the visual field--discriminated by color--at a full 60 Hz frame rate, with a resolution of 200x250 pixels. The board can also perform more general-purpose processing on low-resolution 24-bit RGB frame grabs from 64x48 to 64x250 pixels in resolution. Input to the Cognachrome vision system is standard NTSC, providing flexibility in video input source and viewing the real-time video image. The vision board also outputs a 60 Hz NTSC image of the objects that it is tracking to help in debugging. Figure 3 includes a picture of the vision board, and its technical specifications are listed in Table 3.

Using provided pre-programmed algorithms, the system can be set up to output tracking data over serial ports to another computer, or the user can write his/her own vision software, making use of the built-in libraries, and to make this data available to a host system through the serial port. In this project, the board communicated with serial port of

Table 3. Technical specification of the cognachrome vision board (Newton, 1999).

SPECIFICATION	VALUE
Size	64 mm x 160 mm x 32 mm
Weight	230g
Power	5v (digital): 400 mA 5v (analog): 110 mA 12v (analog): 20 mA
Tracking resolution	200 x 250 pixels
Tracking statistics, per object	Centroid (x , y) Area Angle of orientation of major axis Aspect ratio (square root of ratio of second moment around major and minor axes)
Number of tracking channels	3 color tracking channels (each channel can track multiple objects of the given color)
Maximum number of simultaneous objects tracked	Approximately 25 (tracking performance falls below 60 Hertz after 5 to 7 objects, depending on size and statistics computed)

host computer. The ARC development environment is a serial interaction program, which allows users to use a host computer to communicate with the Cognachrome vision system (Newton, 1999). In this project, ARC was used to interact with the Cognachrome vision system for configuration. By default, when the Cognachrome vision system starts operation, it goes into object tracking mode. In this mode, the incoming video stream is digitized to a resolution of 200 (250, and each pixel is checked for membership in the three independent color look-up tables. Connected regions of recognized pixels are merged, and various statistics about the regions can be calculated, such as, centriod (center of gravity of the color), area, orientation of major axis, and relative sizes of major and minor axes. A protocol string provides a flexible way to configure the data that is calculated and transmitted by the board.

In deciding what colors to use when marking objects for tracking, the main issue is finding colors, which are different from those typically in the environment. The highly saturated, neon colors often works well. Compared with glossy surfaces, matte surfaces get better tracking results. Glossy materials are much more affected by the angle the target's surface presents relative to the light source and camera. In this project, the user's hand needed to be tracked, and a blue matte material board was set behind the hand to serve as a background. This was done to eliminate the need for the user to wear special gloves on his/her hand. In some related projects, the user has to wear special gloves of certain colors to distinguish the hand color from the background color (Segen and Kumar, 1998) (Soh et al., 1997). However, the general goal of this research was to seek a more natural human computer interface. Not requiring the user to wear special clothing or devices, increases the ease of use and acceptability of the interface.

3. Speech processing

In this project, the user's verbal commands were captured using a microphone and processed by a discrete-speech and user-dependent recognition system, DragonDictate (Dragon Systems, 1999). DragonDictate allows the users to enter commands and to dictate text by speaking into a microphone. Th user can use DragonDictate with a keyboard and mouse, or can use it completely hands free. The total number of words in the active vocabulary (all with full phonetic and language models) is 30,000. Training helps DragonDictate recognize user's particular speech patterns. DragonDictate learns from everything that the user says. The amount of time that a person spends on training greatly improves speech recognition and saves time later. The built-in quick training program takes about 25 minutes to complete. To make working with speech easier, users can create their own commands, called macros, to automate tasks. A macro can be a sequence of keystrokes, including printable characters and non-printing keys.

4. Pan/tilt camera

The Sony EVI-D30 is a high speed, wide-angle pan/tilt video camera that can be controlled by RS232C serial control using VISCA (VIdeo System Control Architecture). A picture of the camera is shown in Figure 3. The pan/tilt ranges of the camera are listed in Table 4.

Table 4. The pan/tilt ranges of the camera (Sony, 1999).

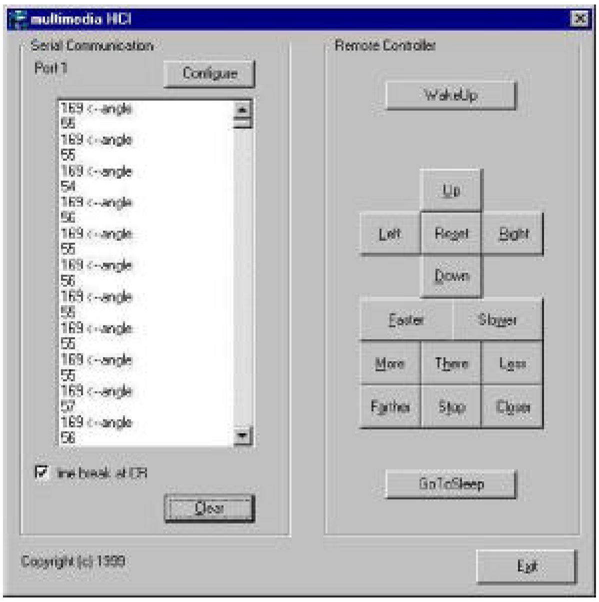
ORIENTATION	MAX SPEED	DEGREE
Horizontal (pan)	80 degree/sec	-100° ~ +100°
Vertical (tilt)	50 degree/sec	-25° ~ +25°

5. Graphical user interface

The general goal of this project is to incorporate naturalness in the design of human- computer interface handling continuous natural gesture and speech inputs. Fast reaction and response are important during the interaction between the user and the controlled system--pan/tilt camera. The utilization of Cognachrome vision system and DragonDictate speech recognizer makes it possible to obtain fast performance from the interface. This section discusses how the speech interface with hand gesture interfaces were integrated together in order to build a fast and efficient user interface. The interface system also includes a graphical user interface (GUI), through which the user can interact with the computer or pan/tilt camera. It should be noted that this GUI is not required to control the program, since only speech and image processing are needed. Instead, the objective of the GUI is to visually present the functionality of the interface system, in addition to the GUI being used for debugging and the initial setup, training (calibration) of the system. Some related projects in human-computer interface design area are designed and implemented in workstation under UNIX system to get fast and appropriate performance. However, this project applies and implements a natural interaction between a person and a computer under Windows NT system, achieving reasonably fast performance. The multimedia interface GUI in the form of a dialog box window was developed using Microsoft Visual C++ MFC (Microsoft Foundation Class) (McGregor, 1996), as shown in Figure 4. The interface dialog box is composed of two parts, the serial communication component, and the remote controller component. The software algorithm is depicted in Figure 5.

5.1 Serial communication module

The function of this module is to continuously display the result of serial port communication from a port that is connected with the vision system board. The list box continuously shows the direction by which the user's hand is pointing to, with the angle ranging from 0(- 180(. It also shows the aspect ratio of the image, ranging from 10 to 255, and defined as the square root of the ratio between of the second moments around the major and minor axes, times 10. For example, for a circle (or a square), the aspect ratio would be 10. For an oval (or rectangle) that is twice as long as it is wide, the aspect ratio would be 20. In tracking the user's hand, the angle of the major axis of the hand is only meaningful if it is elongated. Thus, if the detected image of user's hand is not elongated, the data cannot correctly represents the direction pointed by hand. Therefore, the interface used the aspect ratio to determine the reliability of the angle, assuming that if the aspect ratio was greater than 11, the angle was considered valid and the interface could use it to determine the direction. Otherwise, the system ignored the angle and the user could retry.



```
169 <--angle
56
169 <--angle
55
169 <--angle
54
169 <--angle
56
169 <--angle
55
169 <--angle
56
169 <--angle
56
169 <--angle
55
169 <--angle
55
169 <--angle
57
169 <--angle
56
```

Fig. 4. Graphical user interface component of the interface system.

The data format shown in the list box was set according to the protocol string code of the vision system. By doing this, the serial communication module provided a real-time feedback to the user during the process. It helped the user to identify the exact angle he/she was referring to and to have the user possibly modify or fine-tune the direction by adjusting the hand gesture. In addition, by doing this, it assisted the user to locate the general position. Since the list box has limited space, under the list box, a pushbutton is provided for the user to clear the list box in order to read the current angle easily. When the user's hand was not placed in the viewing area of the camera, the tracking mode was paused, and the list box in the interface dialog box stayed on the last data. Whenever the user's hand appeared in the view area of the camera, the interface system automatically started the tracking mode, and the current angle was displayed. Above the list box, a pushbutton is provided to allow the user to configure the system, by setting the parameters for the communication protocols between the vision board and the serial port.

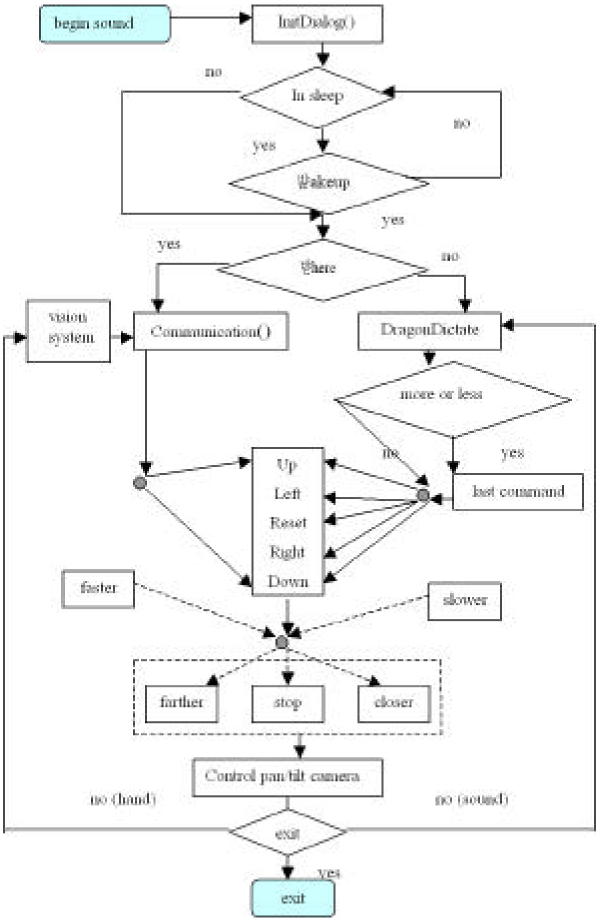


Fig. 5. Software algorithm of the interface system.

5.2 Remote controller module

The remote controller module is comprised of 15 buttons for the control of the pan/tilt camera, all of which can also be invoked via speech. By uttering a button's name, the system will prompt the function of the corresponding button, since all the button names are included into the system vocabulary. The command buttons include:

- *Orientation* buttons: including *left*, *right*, *up*, and *down* to control the basic movements of the pan/tilt camera.

- *Reset* button: it reset the pan/tilt camera to its fixed home position.
- *Speed adjust* buttons: the *faster* and slower buttons provide the function of adjusting the pan and tilt speed, by slightly increasing and decreasing the speed, respectively.
- *Position memory* buttons: after using one of the orientation buttons, if the user wants to continue using the same orientation button to adjust the camera to point to a location, the user can choose to use position memory buttons *more* and *less*. The interface remembers the orientation button, which was used last time. Using the *more* button will move the camera in the same direction as previous one, while using the *less* button will move the camera in the opposite direction. For instance, the sequence of commands *left, more, more* will results in the camera moving to the left three times.
- *Hand pointing* button: the first step of using this interface is to use hand pointing gesture to locate the general position. User can issue the command *there* to active the hand pointing interface. User can also use this function to relocate to a different position during the entire operation process of the interface.
- *Pause and resume* buttons: Since this project provides a continuous user interface, it is necessary, and important to allow the user to pause and resume the interface whenever necessary. The *wake-up* and *go-to-sleep* buttons provide these functions. When the *go-to-sleep* button is activated, the interface will enter the pause status. It will ignore all the speech commands, button hits, and hand pointing, until it is interrupted by the *wake-up* command. These two buttons can be used any time during the operation process.
- *Zoom* buttons: the *farther*, *closer* and *stop* buttons provide zoom functionality. These commands can be used to zoom in and out with the camera.
- *Exit* button: using the *exit* button, the user can terminate and exit the entire interface.

IV. Experiments

In order to evaluate the performance of the interface system, series of experiment were performed, as described in this section, including the experiment setup, the pre-experiment calibrations and training of the system, and the experimental results.

1. Experiment setup

The hardware devices and the software components used in this project as part of the multimedia interface system are listed in Table 5, and the experiment setup is shown in Figures 6 and 7. The host computer serial communication port 1 is connected to Cognachrome vision system board serial. The output of PULNiX camera is hooked up as video input of the vision system board. The video output of the vision system board is connected to the color TV monitor video input for hand tracking training. The Sony pan/tilt camera is connected to the host computer serial communication port 2. Video output of the pan/tilt camera is connected to the TV monitor video input to provide video feedback picture. The distance between microphone and the user's head is fixed. A blue board is placed behind the user's hand to serve as background. The position of PULNiX camera is fixed. The images or pictures shown on TV monitor's screen are checked to get video feedback. The words (voice commands) on the computer screen are checked as a feedback of the voice recognition. The user must have all fingers placed together when using hand pointing, and valid words must be used when using voice commands. The interface program is paused and resumed when necessary.

Table 5. Hardware and software components of the interface system.

HARDWARE AND SOFTWARE	FUNCTIONALITY
Personal computer	Provide the computing power, and running the interface
PULNiX TMC-7 CCD color camera	Capture the images of user's hand
Sony pan/tilt camera	Focus on the location specified by the user
Cognachrome vision system board	Track user's hand using color tracking
ARC development environment	Configure the vision board
TV monitor	Provide feedback of the location to which the camera is pointing
Microphone	Capture user's voice
DragonDictate	Speech recognition engine
Microsoft Visual C++	Develop the graphical user interface dialog box

2. Pre-experiment calibrations

Prior to using the interface, the system must be calibrated and trained in order for it to be effectively utilized by the user. This system training is user-specific and is required both in hand tracking and in speech recognition. Hand tracking training is used to make the vision system recognize the hand color of the specific user under the ambient lighting and background condition. Speech recognition training helps the system recognize user's particular speech patterns. Proper training of the system results in better performance of the interface program.

It is important for each user to calibrate the vision system. It helps adapt the vision system to the user's hand in the environment. The goal of the vision system is to classify each point in the image as being interesting, meaning that it may be part of a target to track, or uninteresting, meaning that it probably is not part of a target. It is best to train the vision system to recognize certain targets in certain situations from empirical measurements. Hand tracking training process begins by placing the user's hand in the center of the camera's viewing area, after which the color is sampled. Persistent parameters are set as different values to get good result, as the region is grown and shrunk. Finally, after obtaining acceptable results, the training and parameters are saved in the system. Even though for a new user the first training takes about one hour, after becoming familiar with the system, it would take the user approximately 10 to 25 minutes to finish the training process.

In the speech recognition process, it is important to note that the "words" that make up the vocabulary that can be recognized by the speech recognition system need not be words in the normal sense. Rather, such systems typically work by matching the acoustic pattern of an acoustic signal with the features of stored templates. Within the confines of the system's resolution, these signals could be words, short phrases, or any other discernible acoustic signal or utterance. Because the acoustic pattern for each different user is different, a speech recognition system is a speaker dependent system. It requires to be trained separately for each different user. DragonDictate learns from the user. The time that the user spends on training greatly improves speech recognition and is time efficient. The built-in training program helps to recognize user's particular speech patterns and takes about 25 minutes to complete (Dragon Systems, 1999).

3. Experimental results

3.1 Overview

Once the interface system is started, the user can operate the pan/tilt camera using hand pointing and/or voice command. It is recommended for the user to provide the general direction by using hand gestures since the range of the direction is the entire experiment room (laboratory). After locating the general location, the user can fine-tune the position using voice commands; at the same time, the user can rest his/her arm. The operation speed of the pan/tilt camera is very fast. Generally, the user can endure the response time (time-delay) between uttering the voice commands and the exact movement of the pan/tilt camera. The user also can adjust the pan and tilt speeds at any time during the operation process using associated voice commands. The interface program can be continuously operated. The no-keyboard, no-mouse interface provides correct, precise, and fast performance under three conditions:

- Correct voice command recognition
- The user keeping his/her hand in vertical level
- Only one image shape is captured by vision board under lighting and other conditions

Under some extreme situation (for example, when one of the three conditions above is violated), the interface program is likely to provide wrong information and the pan/tilt camera could possibly move to an incorrect position. Under this type of situation, the user needs to change hand orientation or utter new voice commends to adjust or correct the movements of the pan/tilt camera. The experiment process is shown in Figure 6 with the TV monitor showing hand tracking image during the setup and training. Figure 7 shows the experiment in process with the TV monitor showing the location selected by the user.

3.2 Image processing interface

One of the restrictions of the hand pointing is that it is two-dimensional. The user can only provide direction information in the vertical level, in this specific experiment, the direction of up and down. For the horizontal level (left and right) direction, user can only use voice commands to control the camera. The hand tracking results obtained from vision system can possibly be "fragile". Many changes in the scene, including changes in the lighting and changes in hand orientation can break the training. In addition, for different users, the hand complexion varies. The matte and glossy features of the hands also vary. When the user changes his/her hand orientation or position, the lighting on the hand might also change. Because of the white light reflection caused by the glossy feature of the hand, in worst case, after the user changes the hand pointing direction, the vision system might provide wrong orientation information. The vision system catches the object as an image segment (blob). Under some lighting condition, when the user places his/her hand in some position or orientation, the vision system will detect more than one object. The angle of the direction is meaningful if and only if the tracked hand is caught as one elongated object, and the hand is pointing to a direction. Therefore, to get the correct direction information, it is important and necessary to keep the entire hand as one unique image by placing all fingers together.

3.3 Speech processing interface

Although simple vocabulary provides fast response and performance, and makes the interface program easy to operate and maintain, the limited vocabulary used in the interface command does set restrictions on the application. During the experiments, if a voice command uttered by the user is correctly recognized by the system, the interface program shows the correct result. The voice part in the interface program causes problems when the interface does not recognize the user's voice command. These situations include a few possible cases:

1) The user utters a valid command in vocabulary (e.g. left, right, etc.), but the interface does not recognize it.

2) The user utters a valid command in vocabulary (e.g. left, right, etc.), but the interface recognizes it as a different word.

3) The user utters an invalid word, not in the vocabulary.

4) The system is influenced by the environment's noise and provides wrong recognition.

These types of cases are dealt with by the user, as retry attempts will correct the situation, guiding the interface to perform the actual commands issued by the user.

3.4 Guidelines for experiments

Throughout the performance of numerous experiments, a number of guidelines were developed. These guidelines refer to the experimental characteristics that can yield a more effective and productive multimedia interface:

- Speech recognition is devoted to a specific user. Different users should perform their own speech recognition training. Other users' voice or environmental noises will influence the performance of the interface.
- Foam muffler should be used in microphone to dampen

background noises, such as breathing. The microphone should be positioned the same way every time, about a thumb's width from the corner of user's mouth. If a desktop microphone is used, it is best for user's head to be kept in the same position relative to the microphone every time.

- The result of color tracking by the system is influenced by many factors such as lighting, background color, position of the CCD camera, and system configuration. The lighting, background color and camera's position should be fixed, and the same persistent parameters should be used during the experimental processes.
- The user's hand should be kept at a vertical level, with all the fingers kept close together when using the hand tracking mode.
- The interface program should be processed under same conditions of microphone position, lighting, and environmental noise. Major changes in the environment can potentially adversely affect the performance of the interface system.

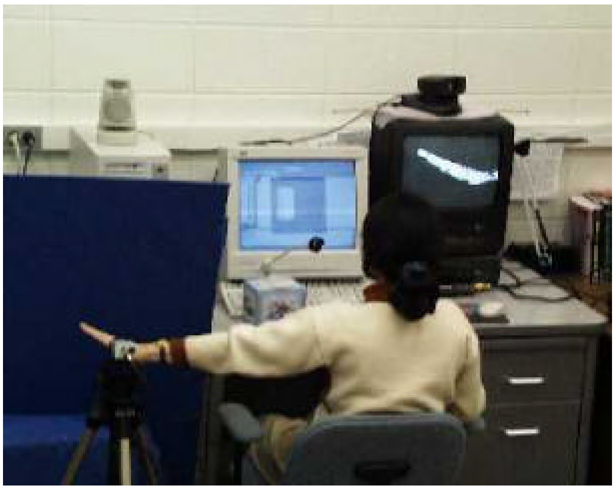


Fig. 6. Interface experiment process, with hand tracking image displayed on TV.

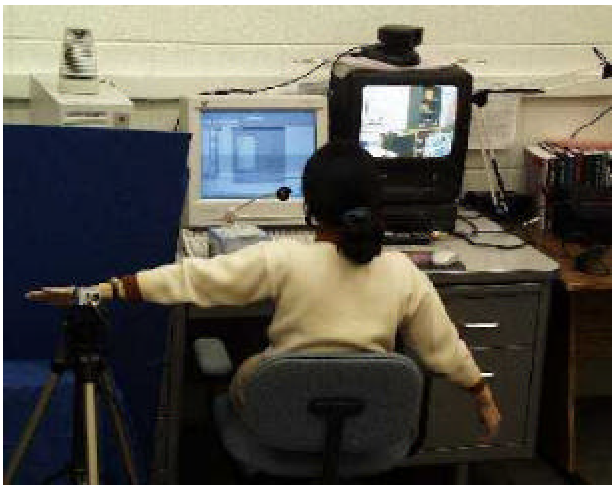


Fig. 7. Interface experiment process, with location feedback image displayed on TV.

V. CONCLUSIONS

1. Contributions

All aspects of human-computer interactions, from high-level concerns of organizational context and system requirements, to conceptual, semantic, and syntactic levels of user interface design, are ultimately funneled through physical input and output actions and devices. The challenge is to design and implement novel systems and types of dialogues that better fit and exploit the communication-relevant characteristics of humans. In doing so, naturalness is an important goal. In seeking naturalness, designers attempt to make the user's input actions as close as possible to the user's thoughts that motivated those actions, that is, to reduce the gap between the user's intentions and the actions necessary to input them into the computer. Many modern interfaces rely exclusively on a single mode of interaction such as speech or a computer mouse, but rarely attempt to use multiple modes. It is still relatively new in the area of the intelligent multimedia interfaces to have systems that integrate speech input, speech output, natural language text, graphics, and pointing gestures for interactive dialogues between humans and computers. This paper provided a multimedia and multi-modal interface system between users and computers to operate and control a pan/tilt camera. It combines three communication modes of speech, hand gesture, and graphic dialogues. The pursuit of knowledge in this area of research is potentially limitless, and much additional work remains to be done. This paper provided some information about how multiple modalities can be combined together to apply to a more "natural" human-computer interaction methodology.

2. Limitations

Although this project can successfully provide a "natural" multimedia human-computer interface, it has some limitations, including the hand tracking background. Among interface designs using hand gesture, some of the projects use gloves to make the hand tracking task easier and more efficient. Some glove technologies use mechanical sensors. These devices typically report a vector containing the bend angle of each of the joints of each finger of the hand. Most glove devices use a three-dimensional tracker, so that they can report the position and orientation of the hand, as well as the angle of each finger. At the beginning of the development process of this project, the user was required to wear a yellow matte glove in order for the system to gain better color tracking results. During the entire procedure, it was uncomfortable for the user to wear a glove and therefore, a blue board was setup behind the user's hand to serve as a background. Comparing the two methods, the background method without the need for gloves was more preferable and was consequently used in this project. However, it does cause a limitation at the same time when it brings the convenience. The final goal of human-computer interface is to allow the hand pointing to take place anywhere in the environment. In the current project, the valid range is limited to the places where the blue board can act as background.

Another limitation of this work is the training process. Before using the interface program, the user needs to participate in two training tasks of hand tracking training and speech recognition training. For a user who uses the system for the first time, it may take about 40 to 60 minutes to finish the hand tracking training. With practice, the user may get a much faster training speed. For the speech recognition training, voice training takes about 25 minutes to complete. Both trainings are user-dependent. Even for the same user, there may be different training results, according the specific environment conditions, such as lighting, camera position, background, microphone position, and environment noise. Valid vocabulary was also a limitation. Although simple vocabulary makes the training and the interface program to run faster, it also brings some restriction to the application. The user only has limited vocabulary to utter in order to operate the pan/tilt camera. This project needs more work to yield a more natural and continuous language interface. Additionally, feedback is another limitation. From the interface program, user can get feedback of the angle or direction to which he/she is pointing. With one television monitor, user can only get either location feedback or hand image feedback.

3. Future work

One possible extension of the presented research includes improving the hand pointing. In this project, with one CCD camera, it is only possible to get a two-dimensional image instead of a three-dimensional image. The interface can only provide direction information in vertical level. In future work, another camera should be added to allow the system to report angle information in horizontal level using stereo images. By doing this, the system can produce the necessary three-dimensional direction information. Expanding the vocabulary is another possible future work. In this project, the available voice commands are limited and non-continuous. In future work, more words and even phrases will be added to the interface vocabulary to provide a more "natural" speech for control of the pan/tilt camera. Improving tracking background is also needed. In order to expand the valid range of hand pointing, the background needs to be eliminated. The hand pointing working space should be expanded to the entire environment. Shortening the training time can also be helpful. Currently, the training time for a new user ranges between 45 minutes and 85 minutes. In future work, this needs to be shortened as much as possible. Incorporation of voice output feedback could also improve the system. The current system does not support natural voice output, and the user can only get feedback from the graphical user interface. Adding the capability to also generate voices will aid the two-way natural communication between the user and the computer.

REFERENCES

[1] ACM SIGCHI. *ACM SIGCHI curricula for human-computer interaction*, http://www.acm.org/sigchi/ cdg/ cdg2.html. 1999.

[2] AT&T *Watson Speech Recognition*, http://www.speech. cs.cmu.edu/comp.speech/section/Recognition/att.html, 1999.

[3] Baudel, T., and Baudouin-Lafon, M. "Charade: remote control of objects using free-hand gestures," *Communi - cations of ACM*, vol. 36, no.7, pp. 28-35. 1993.

[4] Berry, G. A., Pavlovic, V., and Huang, T. S. "Battle view: A multimedia HCI research application," *Image Formation and Processing Group*, Beckman Institute. 1998.

[5] Bolt, R. A. "Put-That-There: Voice and gesture at the graphics interface," *Computer Graphics*, Vol. 14, No. 3, 262-270. 1980.

[6] Bradski, G., Yeo, B.-L. and Yeung, M.M. "Gesture and speech for video content navigation," *Microcomputer Research Labs*, Intel Corporation. 1998.

[7] Buxton, B. Speech, language and audition, http://www. dgp.toronto.edu/OPT/papers/bill.buxton/audio.html. 1999.

[8] Dragon Systems. http://www.dragonsys.com. 1999.

[9] Feldman, S. "Speech recognition software requires lots of training to work correctly," http:// www.finfacts.ie/ Private/cib/voice.htm. 1999.

[10] Fels, S.S. and Hinton, G.E. "Glove-Talk: a neural network interface between a data-glove and a speech synthesizer," *IEEE Transactions on Neural Networks*, Vol. 4, 2-8. 1993.

[11] FreeSpeech98. http://www.freespeech98.com/ 1999.

[12] IBM. *IBM ViaVoice98*, http://www.speechtech. net/ viavoice98.htm. 1999.

[13] Jacob, R.J.K. Input devices and techniques, Tufts University,http://www.eecs.tufts.edu/~jacob/papers/ crc.txt. 1999.

[14] Lamar, M.V., Bhuiyan, M.S., and Iwata, A. Hand gesture recognition using morphological PCA and an improved CombNET-((, Department of Electrical and Computer Engineering, Nagoya Institute of Technology. 1998.

[15] Mane, A., Boyce, S., Karis, D., and Yankelovich, N. "Design the user interface for speech recognition applications," A CHI 96 WORKSHOP, *SIGCHI*, Vol. 28, No. 4. 1996.

[16] McGregor, R.W. *Peter Norton's Guide to Windows 95/NT 4 Programming with MFC*, SAMS Premier. 1996.

[17] Microsoft. http://www.microsoft.com. 1999.

[18] Morguel, P. and Lang, M. "Spotting dynamic hand gestures in video image sequences using hidden Markov models." *In Proceedings of the 1998 International Conference on Image Processing*, ICIP, Part 3, vol. 3, 193-197. 1998.

[19] Newton *Newton Research Labs*, http://www. newtonlabs.com. 1999.

[20] Noon, B. PlainTalk speech recognition, http://snow. cit.cornell.edu/noon/ListenUp.html. 1999.

[21] Pavlovic, V. I., Sharma, R. S., and Huang, T. S. "Visual interpretation of hand gestures for human-computer interaction: a review." *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, vol. 19, no.7, July 677-695. 1997.

[22] Poddar, I., Sethi, Y., Ozyildiz, E, and Sharma, R. "Toward natural gesture/speech HCI: a case study of weather narration," Department of Computer Science and Engineering, Pennsylvania State University. 1998.

[23] PULNiX. PULNiX CCD camera. http://www.turnkey-solutions.com.au/tmc7.htm. 1999.

[24] Rabiner, L.R. and Juang, B. *Fundamentals of Speech Recognition*, Englewood Cliffs, N.J., Prentice Hall. 1993.

[25] Rozmovits, B.A. "The design of user interfaces for digital speech recognition software," http://www.digital.com/info/DTJM09. 1996.

[26] Schildt, H. *Window 98 Programming from the Ground Up*, Osborne McGraw-Hill. 1998.

[27] Segen, J. and Kumar, S. "Human-computer interaction using gesture recognition and 3-D hand tracking." *In Proceedings of the 1998 International Conference on Image Processing, ICIP*, Part 3, vol. 3, 188-192. 1998.

[28] Sha, H. *Human-computer interaction using only images and sounds*. M.S. Thesis, Department of Electrical Engineering and Computer Science, The University of Kansas. 1999.

[29] Sharma, R., Pavlovic, V.I., and Huang, T.S. "Toward multimodal human-computer interface." *In Proceedings of the IEEE*, vol.86, no.5, 853-869. 1998.

[30] Soh, J., Yoon, H.-S., Wang, M., and Min, B.-W. "Locating hands in complex images using color analysis." *In Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics*, Part 3, 2142-2146. 1997.

[31] Sony. http://www.sony.com/. 1999.

[32] Sullivan, J.W. and Tyler, S.W. *Intelligent User Interfaces*, ACM Press. 1991.

[33] Verbex Verbex VoiceSystems, http://voxweb.voxware.com/2.23.99.verbex.html. 1999.

[34] Vo, M.T. and Wood, C. "Building an application framework for speech and pen input integration in multimodal learning interfaces," http://www-cgi.cs.cmu.edu/afs/cs.cmu.edu/user/tue/WWW/papers/icassp96/paper.html. 1999.

[35] Yao, P. and Leinecher, R. C. *Visual C++ 5 Bible*, IDG Books Worldwide, Inc. 1997.

[36] Yao, P. and Yao, J. *Foundations of Visual C++ Programming for Windows 95*, IDG Books Worldwide, Inc. 1995.

Hui Sha

Ms. Hui Sha is an engineer at Tellabs in Lisle, Illinois. She was a graduate student during 1998 and 1999 in the Department of Electrical Engineering and Computer Science at the University of Kansas. During her academic study and research in the Master Program, her work was focused on research in interaction between humans and computers. Ms. Sha developed a multimedia human-computer interface which utilized the combination of graphics, speech input, and hand gestures. Before joining the University of Kansas, Ms. Sha was an assistant engineer working on local area network administration in China. She received her B.S. on Electrical Engineering in 1994 from Xi'an Jiaotong University, China.

Arvin Agah

Dr. Arvin Agah is Assistant Professor of Electrical Engineering and Computer Science at the University of Kansas. His research interests include human interactions with intelligent systems (robots, computers, and interfaces) and distributed autonomous systems (robots and agents). He has published two book and over 60 refereed articles in these areas. Dr. Agah has been a co-investigator on numerous projects funded by NSF, DARPA, and Sprint. He has taught courses in artificial intelligence, robotics, software engineering, computer systems design laboratory, and intelligent agents. He has served as the technical program committee member, conference session chair, and organizing committee member for various international technical conferences. He is a senior member of IEEE and a member of ACM. Dr. Agah received his B.A. in Computer Science with Highest Honors from the University of Texas at Austin (1986); M.S. in Computer Science from Purdue University, West Lafayette, Indiana (1988); M.S. in Biomedical Engineering from the University of Southern California, Los Angeles, California (1993); and Ph.D. in Computer Science from the University of Southern California (1994).

Dr. Agah has been a member of research staff at Xerox Corporation's Webster Research Center, Rochester, New York; IBM Corporation's Los Angeles Scientific Center, Santa Monica, California; Ministry of International Trade and Industr's Mechanical Engineering Laboratory, Tsukuba, Japan; and Naval Research Laboratory's Navy Center for Applied Research in Artificial Intelligence, Washington, D.C. He has been an instructor at Mansfield Business School, Austin, Texas; Purdue University's Department of Computer Science, West Lafayette, Indiana; and University of Tsukuba's Department of Engineering Systems, Tsukuba, Japan. He has also worked as a systems analyst and a software engineer for entertainment law firms and management companies in Century City and Beverly Hills, California.