Presented at: Intelligent Environments Symposium, American Assoc. for Artificial Intelligence Spring Symposium Series, March 23-25, 1998 at Stanford University

Visualization Space: A Testbed for Deviceless Multimodal User Interface

Mark Lucente, Gert-Jan Zwart, Andrew D. George
IBM Research
lucente@alum.mit.edu
Route 134, Yorktown Heights, NY 10598 U.S.A.

Abstract

The Visualization Space ("VizSpace") is a visual computing system created as a testbed for deviceless multimodal user interfaces. Continuous voice recognition and passive machine vision provide two channels of interaction with computer graphics imagery on a wall-sized display. Users gesture (e.g., point) and speak commands to manipulate and navigate through virtual objects and worlds. Voiced commands are combined with several types of gestures – full-body, deictic, symbolic and iconic – to allow users to interact using only these natural human-to-human communication skills. The system is implemented on a single (high-end) IBM PC, yet provides comfortably interactive rates. It allows for rapid testing of voice/vision multimodal input and rapid prototyping of specific multimodal applications for natural interaction.

Introduction

Humans discover and understand their world through interactive visual sensations. We speak, point, gesture, or move things around, but our information machines do not use these rich and intuitive communications channel. To accommodate humans, computers must be designed to hear, see, and (eventually) understand natural human communications. We have created an intelligent environment – the "Visualization Space" (or "VizSpace") – that employs deviceless voice and vision multimodal input. The VizSpace provides a testbed for unencumbered, user-independent, deviceless interfaces, and is directed toward a number of future applications: scientific visualization, medical imaging, entertainment, training and education.

The VizSpace "hears" users' voice commands and "sees" their gestures and body positions, allowing users to collaborate in a shared workspace using interactive visual computing. Interactions are natural, more like human-to-human interactions. Users are free to focus on virtual objects and information and understanding and thinking, with minimal constraints or distractions by "the computer", which is present only as wall-sized (3D stereoscopic) images and sounds (but no keyboard, mouse, wires, wands, etc.). VizSpace employs continuous speech recognition for voice input (IBM ViaVoice), machine-vision input of full-body gesture, and a high-bandwidth network for access to 3D data. It is a deviceless descendant of the Put That There system (Bolt 1980) at MIT and of VIDEODESK and other work by Krueger (Krueger 1990). VizSpace takes the deviceless approach used in "Perceptive Spaces" by Wren et al. (Wren et al.1997a) and emphasizes voice/gesture input fusion, as in the work by Koons, Sparrell, and Thorisson (Koons, Sparrell, and Thorisson 1993; Thorisson 1997) at MIT. The VizSpace is implemented relatively inexpensively using common hardware and much off-the-shelf software.

Description of Visualization Space

Our Visualization Space is a room (about 4x8 meters) where one wall is a rear-projection display (with an image of 2.5-meter in width). The users see no computer boxes, no keyboards, no pointing devices. A camera is hidden above the display, and (for now) a user wears a wireless microphone. The user can walk into the room and immediately begin to interact with VizSpace without the need to sit, type, or to strap on any technology (with the current exception of clipping on the microphone). Users navigate through virtual worlds and manipulate virtual objects that are displayed on the large display.

Figure 1 shows a schematic of the VizSpace. The two input channels are the voice and vision. Voice and live video are each digitized and processed using a single IBM PC, also responsible for integrating these inputs with application software (e.g., a VRML browser) and rendering computer graphics for visual feedback.

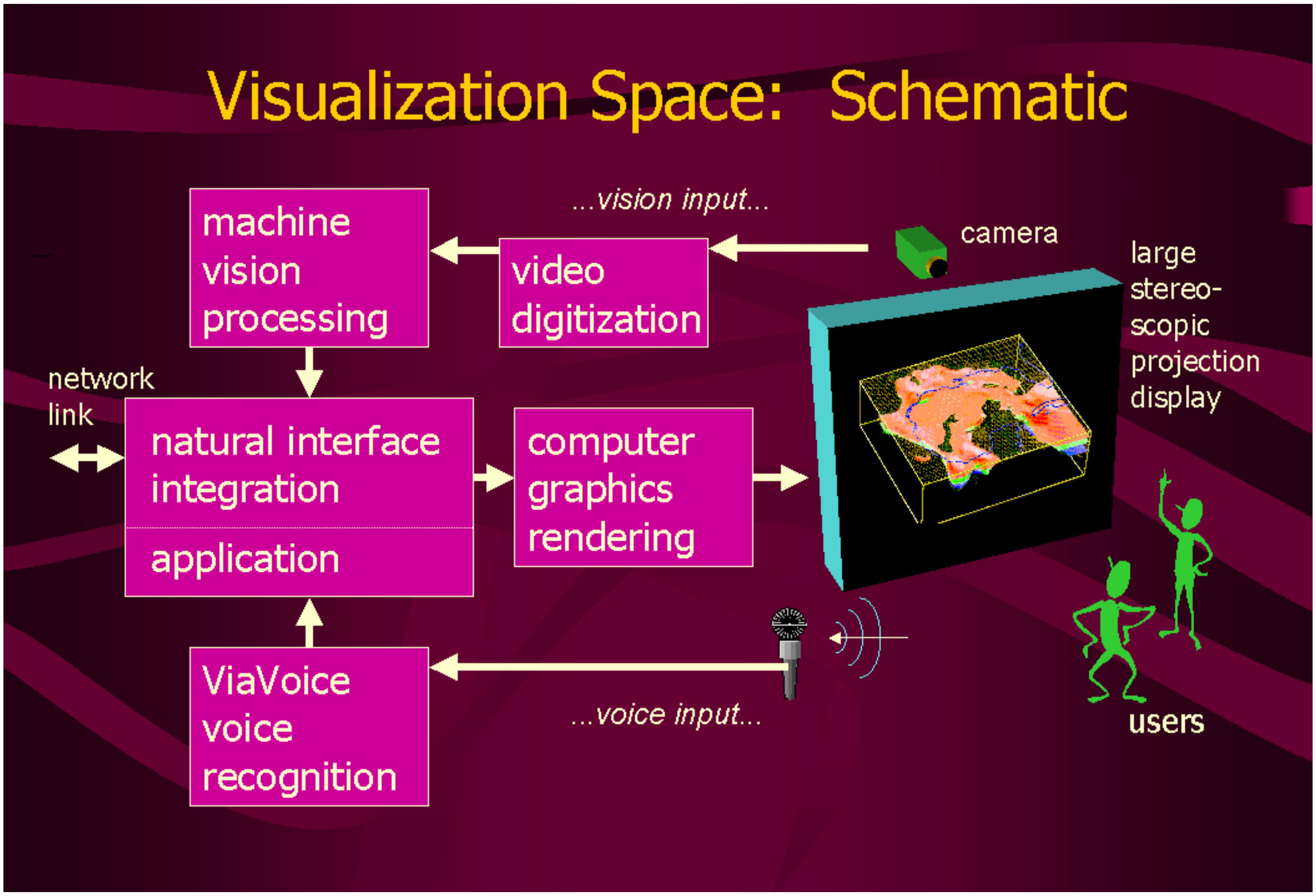
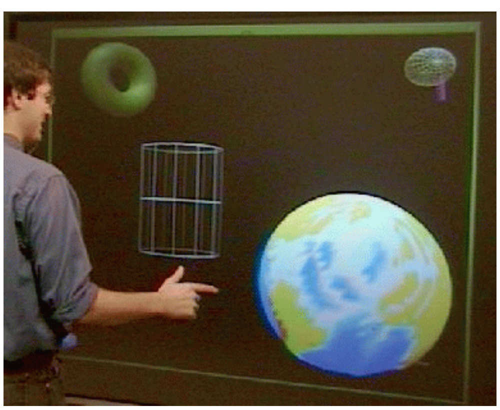
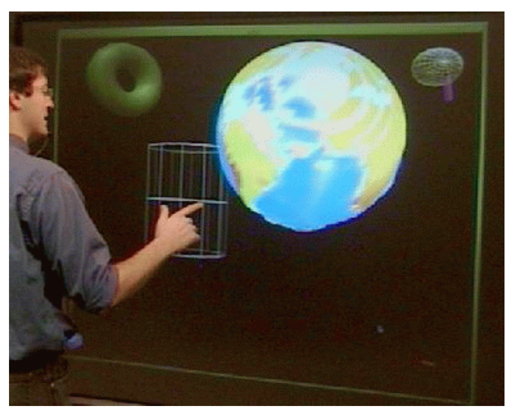


Figure 1: A schematic of the Visualization Space. The two input channels – voice and vision – are combined to control applications running on a single IBM PC. Visual output is displayed on a large projection display.



"grab that"



"leave it there"

Figure 2: A typical interaction with the VizSpace. The author (Mark Lucente) moves a virtual object (the Earth) by pointing and speaking.

Voice

VizSpace employs IBM's ViaVoice continuous-speech dictation system to convert user-independent speech into text strings. Voice commands are specified by the application as a list of grammars. When a user speaks, the ViaVoice engine (running on the PC's central processor) looks for phrases that match the command grammars, and passes each match (or lack thereof), including a time-stamp, to the integration program. ViaVoice has a 40,000-word vocabulary which can be expanded through training. The current system uses a single microphone worn by one user. We are working to move the microphone away from the users and embed it in the walls of the room. Even an untrained ViaVoice engine functions well as a speaker-independent command-phase recognizer, and it is used to provide voice input from multiple speakers. ViaVoice also provides built-in speech generation software, which we use to provide audible verbal feedback to the user.

Vision

Live video of the space in front of the display is digitized and processed using a machine-vision system which runs on a the PC's central microprocessor. Humans naturally work in environments with visible ambient light; machine vision provides non-invasive spatial sensing in such natural settings. We adapted the PersonFinder (or "Pfinder") software developed at the MIT Media Laboratory (Wren et al. 1997b). Pfinder uses a multi-class statistical model of color and shape to segment a person from a background scene, and then to find and track people's head, hands, feet, and upper and lower body. Our version of Pfinder runs on any IBM-compatible PC using a Win32-based OS. Using this Pfinder-based machine-vision code, the integration program monitors information about the three-dimensional location of the user's body parts: head, hands, feet, upper and lower body, and centroid. No chroma-keying or tracking devices are used. Typically, the color (YUV 4:2:2) images are downsized from 640x480 and processed at a resolution of 160x120.

The vision system operates asynchronously, but is coded to run with a fixed latency between the reading in of a video frame and the updating of person location information. This crude substitute for an absolute timestamp allows for synchronization with voice commands.

Integration

The interface integration program (called "Intergrate") combines voice and vision information and interprets it to control one or more visual applications. Our two categories of interaction – navigation through and manipulation of virtual objects – are relatively straightforward. Intergrate keeps track of the user's desired actions and state, using (at present) simple rules. Its primary duties include

- synchronizing voice and vision information,
- interpreting a user's gestures,
- determining the validity of certain voice commands,
- passing user location and gesture information to the application.

Synchronization is crucial to the multimodal input system. Voice information carries a time-stamp. As currently implemented, Intergrate assumes a consistent latency in the vision channel from moment of capture. Intergrate buffers approximately one second (an upper limit to the latency in the of voice channel) of vision information, selecting the moment that matches the voice timestamp. Synchronization is particularly important for a command such as "get rid of that" in which the user points to the undesired object briefly (during the speaking of the command).

Interpretation of gestures involves combining the spatial location and motion of the user's head and hands. Several such gestures are discussed as Examples of Interaction. An important example is to calculate the direction of the user's pointing gestures (for each hand) using a simple technique. The difference vector between the absolute coordinates of the head and (pointing) hand are used to bilinearly interpolate pointing direction. The direction is combined with the user's distance to the screen, and the lateral location of the head. At initialization, the user is asked to point to four targets (more reliable than only two) and to read in turn the word on each target. Intergrate uses the head-hand vector for each target to calibrate the system. We have found that users of different heights and different styles of pointing have remarkably consistent and extremely repeatable pointing gestures.

Interpretation of pointing gestures is typical of Intergrate's role in validating commands. For example, when a user uses deictic words (e.g., "there" or "that" as in "grab that" or "leave it there"), Intergrate first determines the pointing direction of both the left and right hands. In the usual case, one hand performs a valid pointing gesture (i.e., in the direction of the active area of the display's screen), and Intergrate passes the valid pointing information along with the specific instruction to the application. If neither pointing gesture is valid, the command is ignored, and the user is asked to clarify. If both are valid, Intergrate simply takes the average.

In addition to converting voice commands to instructions for the applications, Intergrate provides a stream of user location information. For example, the user is generally welcome to move about the room, and Intergrate converts user head location to an eyepoint location and instructs the application to update the rendered image appropriately. Hand locations and pointing directions (and their validity) are continuously provided so that the application can interactively move or alter objects. For example, the user may have asked to grab and object and is moving it by pointing to a new location. Or, the user may have asked to interactively resize an object by holding up both hands and adjusting their separation.

Implementation and Performance

To date we have implemented VizSpace both on Unix (AIX) platforms and on an IBM PC platform. Both are designed to work across the network, so that different processes can run on different machines. Because it is simplest, the single IBM-PC implementation is described here.

The Visualization Space runs on a four-processor IBM Netfinity 7000 PC Server (512 KB cache/processor and 512 MB of RAM) running Windows NT 4.0. A PCI-bus Snapper (Active Imaging) provides video digitization, downsizing, and conversion to YUV from the analog composite video provided by a standard CCD camera. A common sound card (Creative Labs SoundBlaster) provides digitization of sound from the wireless microphone. A GLyder MP (Symmetric, Inc.) graphics accelerator card provides rendering support, including geometry processing, for fast interactive graphics. All other functions – voice recognition, machine vision, input integration – are performed in the central processors. Interactive applications are written directly with OpenGL instructions, or are implemented using a VRML browser that is based on the Cosmo Player 2.0 for VRML 2.0 (Cosmo Software) and controlled as a slave process.

The IBM PC provides the computing power required to perform interactions at a comfortable rate of interactivity – more than ten updates per second. The voice channel has a typical lag of 0.4 seconds on continuous speech. The vision channel runs at more than ten frames/second, with a latency of less than 0.2 seconds. Most of the computing power is used by the machine-vision system.

Examples of Interactions

The Visualization Space and its deviceless natural interactivity are ideal for many applications:

Education and Entertainment. In location-based entertainment (e.g., a virtual theme park), where interface hardware is often damaged and dirtied, the hands-off gadget-free interface of the VizSpace allows robust unobtrusive interactivity. Virtual adventures and walk-throughs are more fun and memorable. A user might take another user on a virtual tour of a historic site, the human anatomy, or another planet.

Scientific Visualization. Using the VizSpace, the interactive communication of complex visual concepts is as easy as pointing and speaking. Users can collaborate remotely with users at other workspaces and workstations via the network, sending voice and real-time video.

Retail: Vacations, Real Estate. Plan a vacation, or look at potential real-estate purchases for home or business, simply by walking, gesturing, and asking for information. Take a virtual visit to a tropical island to get a feel for your potential vacation site. Or check out that factory space that you are planning to buy. The natural interactivity of the VizSpace, combined one-to-one scale of the displayed imagery, provides a visceral effective experience.

These and other applications involve the navigation through and manipulation of virtual objects. Intergrate combines voice information with several types of interpreted gestures – full-body, deictic, symbolic and iconic – and facilitates interactions that utilize the voice modality, the body and/or gesture modality, or both modalities:

- Voice: Users may ask to visit a particular place (e.g. “take me to Athens”), or to control an object (e.g., “spin it”, “make it bigger”), or to navigate (e.g., “let’s go faster”).
- Body/gesture: Users may control navigation using only gestures or body location. For example, stepping to the right causes a change in the direction of forward motion – an example of full-body gesture input.
- Voice and body/gesture: Both object manipulation and navigation involve many such combinations.

This third class – the fusion of speech and body/gesture – is the most interesting. For example, if a user points and asks to “put a box there”, Intergrate calculates the user’s deictic pointing gesture information and instructs the application to create a box primitive at the desired location. The user can “grab that” object, selecting an object by pointing to it, and then move the object interactively by pointing to a new location. If an object has been selected (by pointing to it and saying “select that”), a user can say “make it this big”, holding hands apart by a desired width – a use of iconic gesture. Furthermore, an object can be interactively resized or rotated in a similar way, beginning with a command such as “change its size” or “rotate it like this”, followed by movement of both hands, and ending with a command such as “this is good” or “leave it like that”. Navigation also utilizes multimodal interactions. For example, while moving forward, a user can alter course heading by using a lateral waving gesture (as if clearing dust off of a tabletop) with either a left or right emphasis. This use of iconic gesture is particularly effective, as users often to automatically express themselves this way.

Current Focus and the Future

The first stage of VizSpace development involved finding and combining tools to perform the computationally intensive voice and vision input functions. Multimodal experiments followed, including a system with two simultaneously active users. We have three primary interrelated goals:

- rapidly explore new ideas in multimodal input;
- create prototype systems for specific applications and test their efficacy;
- begin to develop a layer of intelligence between the multimodal input channels and the computational tasks, i.e., create an interface that “understands” the users.

Our choices of hardware and software have been guided by the first two goals. At present, we can create a demo or application quickly simply by using a modeler to build a virtual world and then integrating a set of pertinent interactions. We continue to create a lexicon of interactions described by the relevant voice command grammars and spatial information. Sets of commands can be shared among several applications, and particular commands can be altered as needed. The second goal – to determine what specific applications benefit from this deviceless system – inspires us to generate new approaches and solutions, as well as stimulating broader interest from the computer industry. In addition to the wall-sized visual display, we are testing a number of different input and feedback scenarios. Visual feedback can be embedded into a table top (great for laying out Web pages, magazines, multimedia storyboards), or use small displays (for educational and entertainment applications on common desktop computers). Systems can utilize little or no visual feedback (appropriate for a smart-car dashboard or smart kitchen). Additional modes of input (e.g., sensors embedded in carpets, walls and furniture) will provide more robustness through redundant detection of user location. Additional important input will include the detection of gaze direction and hand pose, as well as user identification using biometrics such as voice and face recognition.

The interface must be more intelligent – what does the user really want? what does this user usually mean by that? how can the user be prompted to make things clearer? Adding understanding to the interface is a long-term and difficult goal. However, attempts at artificial intelligence tend to succeed in focussed realms. Combining voice and vision information presents such an opportunity, beginning with the simple but important case of knowing when to listen to a user. By understanding the user (i.e., seeing where users are facing, knowing what activities have preceded, perhaps even listening to tone of voice) a truly multimodal input interface can determine when a user is talking to the computer versus talking with other users. Humans commonly use indistinct phrases such as “make it a little bigger” when manipulating objects. The Intergrate system must be progressively trained to the habits of a given user to interpret this phrase: how much bigger? does the user often ask for “bigger still” if the object size is increased by only ten per cent? what does the user mean when this type of object is being treated? Arbitration becomes critical for multiuser systems. If two or more users ask for different but operations, how are these commands interpreted? Selecting only the first or loudest or longest command represents a loss of important information. Therefore, a more intelligent interface would engage in compromise, arbitration, or perhaps active solicitation and negotiation – conversational interaction.

Conclusion

VizSpace has demonstrated that simple deviceless multimodal user interface can be implemented inexpensively, drawing on readily available hardware and software. We continue to use it to provide users with a natural interface, and quick easy access to information and communications channels. Users more quickly engage in their tasks as there is little or no distraction from interface hardware. Test subjects have responded by describing positive sensations of “control” and “tangibility”. Many simply feel that the computer “understands” the user, and on the user’s own terms – using natural language and communications skills.

Equal in importance to this natural human-to-computer interface is the natural feel of the human-to-human interaction. When one user points to an object and says “make it red”, other humans present in the VizSpace automatically know the active user’s intentions. Moreover, they immediately learn how they too can participate. And because the interface uses natural language, users are more inclined to express what they want, and to experiment with the possibilities. For example, one young visitor (who could not yet read an instruction manual) was able to interact with the VizSpace after only a one-minute demo that involved a few spoken commands. Later, unprompted, he casually experimented with new voice commands, many of which gave immediate results. These are encouraging observations as we attempt to create the last interface that any user will ever have to learn: a deviceless interface that uses the skills of speaking and gesturing that have been learned throughout life and are performed naturally when communicating with humans.

Acknowledgments

Jeffrey A. Kusnitz, Catherine A. Chess, David A. George, Robert E. Walkup, Thomas M. Jackman, Ulisses T. Mello, Robert F. Cook, Athicha Muthitacharoen, Christopher R. Wren, Flavia Sparacino, Alex (Sandy) Pentland, and many other clever people have helped to make our interactions more fun and efficient.

ViaVoice and Netfinity are trademarks of the International Business Machines Corporation (NY, USA). Likewise, Snapper is from Active Imaging Ltd. (UK); SoundBlaster is from Creative Labs, Ltd. (UK); GLyder MP is from Symmetric, Inc. (TX, USA); Cosmo Player 2.0 is from Cosmo Software (CA, USA).

References

Bolt, R. 1980. Put-That-There: Voice and Gesture at the Graphics Interface. In SIGGRAPH '80 Proceedings, Computer Graphics, vol. 14, #3 (July 1980), pp. 262-270.

Koons, D. B.; Sparrell, C. J; and Thorisson, K. R. 1993. Integrating Simultaneous Input from Speech, Gaze and Hand Gestures. M. T. Maybury (Ed.), *Intelligent Multimedia Interfaces*. Cambridge, Massachusetts: AAAI Press/ M.I.T. Press.

Krueger, M. W. 1991. *Artificial Reality II*. Addison Wesley.

Thorisson, K. R. 1997. Gandalf: An Embodied Humanoid Capable of Real-Time Multimodal Dialogue with People. First ACM International Conference on Autonomous Agents, Mariott Hotel, Marina del Rey, California, February 5-8, 1997, pp. 536-7.

Wren, C. R.; Sparacino, F.; Azarbayejani, A. J.; Darrell, T. J.; Starner, T. E.; Kotani, A.; Chao, C. M.; Hlavac, M.; Russell, K. B.; and Pentland, A. 1997a. Perceptive Spaces for Performance and Entertainment: Untethered Interaction using Computer Vision and Audition. *Applied Artificial Intelligence*, June 1997, vol. 11, #4, pp. 267-284.

Wren, C. R.; Azarbayejani, A. J.; Darrell, T. J.; and Pentland, A. 1997b. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, #7, pp. 780-785, July 1997.