A SPATIAL APPROACH TO SPEECH AND GESTURAL CONTROL IN COLLABORATIVE VIRTUAL ENVIRONMENTS

STEVE BENFORD
Department of Computer Science, The University of Nottingham,
Nottingham, NG7 2RD, UK.

CHRIS GREENHALGH
Department of Computer Science, The University of Nottingham,
Nottingham, NG7 2RD, UK.

ABSTRACT

This paper explores issues relating to the use of broadcast media such as speech and gesture for interacting with non-human (i.e. application) objects in collaborative virtual environments. These issues arise from a recognition that broadcast media such as these are inherently social in nature and that, in order to join in such media, an object should possess a degree of social awareness. Put in simple terms, in addition to being smart enough to understand the content of the input they receive, objects which are controlled by speech and gesture must also understand when they are being addressed and, conversely, when utterances they "overhear" are intended for others. This issue becomes particularly significant in environments were multiple people simultaneously attempt to control multiple objects. Our paper begins by clarifying the nature of the problem and outlining three examples of possible techniques for resolving it. We then focus on one of these techniques, the use of spatial awareness mechanisms. We explore one specific example of this technique and present two demonstrators that have been implemented in the MASSIVE collaborative VR system.

1. Introduction - The Nature Of The Problem

The idea of using gesture and, to a lesser extent, speech to improve the ease of interaction with virtual environments has already received considerable attention. For example, the DataGlove is an essential part of many immersive systems [Sturman and Zeltzer, 1994; Blau et al., 1992; Wexelblat, 1994; Appino et al., 1992] - the "default" configuration for immersive use being a head-mounted display, glove-type input device and spatialised audio output. Gestural input has also received considerable attention in less (classically) immersive systems such as Artificial Reality [Kreuger, 1991] and tele-robotics [Papper and Gigante, 1993]. While audio output has long been a feature of virtual reality systems, speech input and control has only recently made an appearance outside the realms of pure speech-recognition and natural language research (with the notable exception of [Bolt, 1980]). Recent contributions addressing speech-input for virtual reality include [Karlgren et al., 1995] and [Godereaux et al., 1994].

The basic premise of this paper is that, although such techniques are useful, they need to be developed within the context of the media involved. In particular, our paper is concerned with *collaborative virtual environments* which might be populated by many participants who are simultaneously interacting with each other and with many different application objects. Two factors need to be considered in such environments:

- The relationship between the use of speech and gesture to command application objects and their use for conversing with other humans. In the everyday world and in teleconferencing or collaborative virtual worlds speech and gesture are transmitted over the inherently broadcast media of sound and vision respectively[1]. Being broadcast, speech and gestures might be received by many objects in a virtual environment, including both humans and non-humans.

- The general problem of synchronising commands issued by potentially many people to potentially many objects. A more limited version of this problem arises in single user systems where many objects may be able to respond to any given utterance or gesture.

The following are specific examples of problems that might arise:

- How does an object distinguish speech and gestural commands directed at itself from overheard conversation?
- What happens if a command is received by several candidate objects which could all respond to it?
- What happens if several people simultaneously try to command the same object?
- How can a user simultaneously control multiple objects?

These problems arise from the nature of broadcast media, as used by speech and gesture, which provide a relatively open channel for multi-party communication and social interaction. We argue that, in order to participate in such media, speech and gesture controlled objects must be *socially aware*. In other words, objects must not only be smart enough to understand the content of speech or gestural input, they must also be able to understand (at least to some degree) the social context in which it occurs and be able to recognise when they are being addressed. Human beings have evolved a range of social skills to support this task, including sophisticated understanding of the context in which utterances are made and use of spatial cues such as gaze direction and body position to manage turn taking and to discriminate the intended recipient(s) of utterances.

Another way of expressing this problem is that by moving the control of objects away from direct manipulation using our hands (the traditional way we apply tools) to speech and gesture (the way we interact socially) we are overloading the latter and must be prepared for additional complications. This doesn't mean that speech and gestural control is intrinsically a bad idea; but it is a more subtle problem than just determining the surface content of speech and gestures (which is a hard problem in itself).

The aim of this paper is therefore to identify techniques to help address this problem and to explore one of these, the use of spatial awareness models, in some detail. We will concentrate on speech in our discussion but the same considerations will apply to gestures if they are made (publicly) visible.

2. Basic speech interaction

We first sketch out the typical structure of a speech-recognition and control system in a single user-single application context before generalising to multiple users and applications.

2.1. Speech interaction

Figure 1 shows a typical speech input/output system for a single user and a single application.

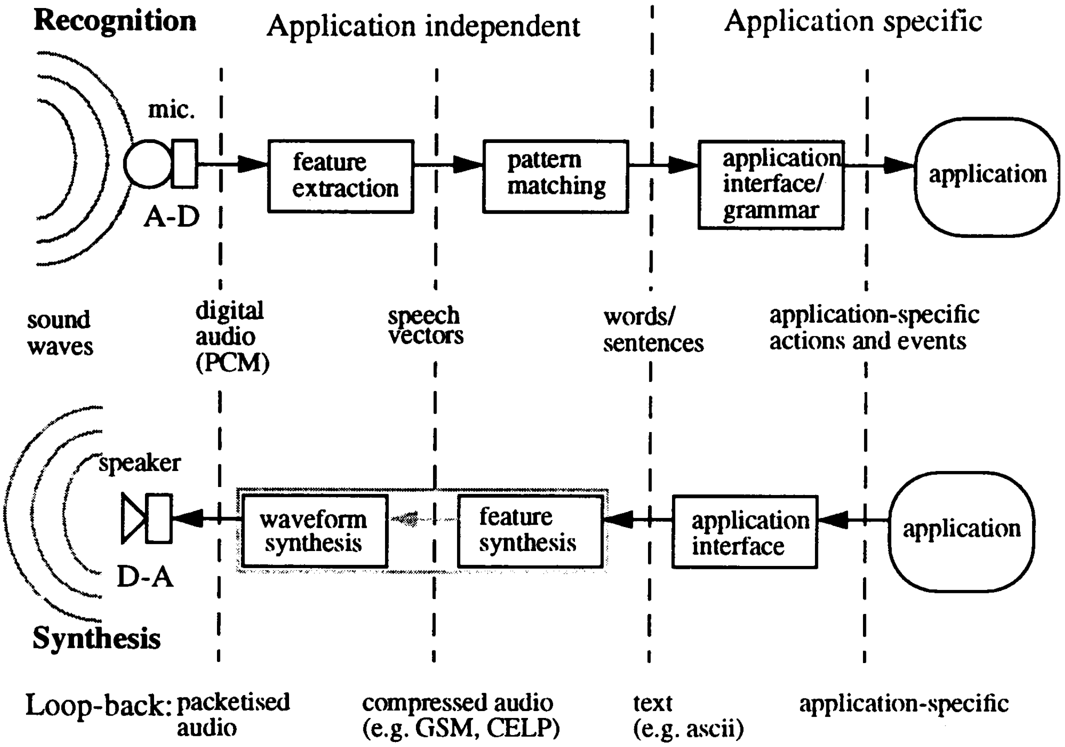


Figure 1: Speech I/O system

On the recognition side, utterances are captured by a microphone and fed to an analogue-to-digital converter to create a PCM digital audio waveform. This is subject to analysis and feature extraction, the result of which is a time-series of parameter values which encapsulate the significant characteristics of the waveform. Some form of pattern matching (e.g. dynamic time-warping (DTW) template matching or hidden markov models (HMMs)) is applied to this time-series to identify the words or sentences most likely to comprise the utterance. These may be used as input for syntactical or semantic analysis before being mapped to application-specific events or triggers which affect the operation of the application.

On the synthesis side these stages are reversed. First application-specific events are mapped (optionally via a production grammar) to synthesisable items (words, sentences). These are used to generate a corresponding digital audio waveform, possibly via an intermediate time-series of characteristic speech vectors.

At each level (digital audio, speech vectors, words, application-specific events) the recognition chain may be looped back to the synthesis chain to get out (an approximation) of what went in. This is not particularly useful for a single user systems, but in a multi-user scenario these channels can be used to support direct audio communication between users.

3. Interaction in multiple user applications

The previous section described the basic interaction facilities employed for speech interaction between a single user and a single application. In this section we generalise this to cater for multiple users and multiple applications. Again we will concentrate on speech interaction as the more general example though the results are equally applicable to gestural interaction.

Consider a system, e.g. a collaborative virtual environment, in which a number of users are simultaneously interacting with each other and with a number of applications which are embodied as virtual objects[2]. Between each user and each application are (at least nominally) speech recognition and synthesis chains comparable to those in figure 1. Responsibility for implementing the different components of these complete chains can be split arbitrarily between the user and the application processes, the collaborative system infrastructure providing an appropriate communications bridge between them for digital audio, speech vectors, words or application-specific actions as appropriate. This controlled bridging functionality is the basis for supporting multiple users and applications and is the main subject of this paper.

Suppose one of those users says something (or makes a gesture). The problems which must be addressed by the collaborative system are:

- which users should hear what is said?
- which applications should be aware of what is said? (e.g. for future reference or as context for future utterances)
- which applications should respond to what is said?

To answer these questions the system may be able to use information about:

- the source of the utterance (who or what said it?);
- the context of the utterance (what else was the person doing, what had they been doing?);
- the content of the utterance (what was actually said?); and
- the potential audience for the utterance (who is there to hear it?).

Note that, within this shared framework, audio output from applications can be considered in the same terms as user utterances: it may be heard by users and other applications and may be interpreted as instructions by other applications.

For each utterance, the collaborative system must therefore deliver it to some subset of the available destinations (applications and users) and provide a context in which applications can decide whether or not to act on received utterances. One can imagine a nearly infinite range of negotiation models and strategies which might be adopted to make these control decisions.

For example, consider the three contrasting approaches outlined below:

- The user might explicitly switch between different modes of interaction or using different applications (e.g. [Allport et al., 1995] uses a foot pedal to activate a gesture-based navigation mode). Or the user might switch on an application through direct manipulation before subsequently commanding it through speech and gesture.
- The system can examine the content of utterances for clarifying information. The most obvious example is the use of a unique name associated with each application to be commanded. However, more subtle verbal or gestural triggers might be possible.
- The system can make use of spatial cues within a collaborative virtual environment such as spatial proximity and orientation, to decide where utterances are being directed (similar to the way in which humans use spatial cues to manage turn taking). For example, an application might decide that it was being addressed only if the speaker was sufficiently close by and facing it. In order to chose from among multiple candidates, applications might also reason about each other's spatial relations to the speaker.

The first approach (out-of-band selection) has the advantages of simplicity and removal of ambiguity, but only at the cost of the user having to explicitly recognise when they wish to use

[2] We will use the general term *application* to refer to an application object which has been directly embodied in a virtual environment.

speech and gesture for commands and when for conversation. This approach places the responsibility for dealing with the difference in social capabilities between humans and non-human applications onto the human alone. Also, there is a loss of flexibility in that users will be unable to speak or gesture conversationally while commanding applications (e.g. one couldn't wave to a friend while "driving" past).

The second approach has the advantage that it uses the same underlying mechanisms as understanding the speech and gestural input and avoids cross-modal coordination. However, the use of any content based approach will be subject to the usual ambiguities of natural language (e.g. what if someone uses an application's name as part of conversation - perhaps when talking about it?). Also, the issues of choosing unique triggers or of knowing what trigger to use when first encountering an application may become problematic for large scale exploratory environments in which users may encounter previously unknown kinds of application. Finally, the use of any complicated or non-obvious trigger will mean extra work for both the user and the collaborative system.

The third approach has the advantage that it aims to employ the same natural techniques that people use to manage turn-taking in the everyday world to the control of applications; we suggest that this will result in a natural, intuitive and flexible model of control. Indeed, support for the spatial aspects of turn taking and conversation would seem to be an appropriate extension to the general philosophy behind speech and gestural control. It also involves minimal interference with the timing or content of users natural conversation. However, the inherent ambiguities involved in using spatial cues might cause problems and successful discrimination between applications may become difficult in densely populated and highly dynamic spaces where many applications are moving about. It may also be difficult to control applications over long distances in this way.

Given the potential advantages and disadvantages of each of these three example techniques, and the range of alternatives which we have not explored here, it is clear that there is no single correct approach: the technique chosen will depend on range of applications and other user-specific factors. Indeed, a very general combination of approaches (including, but limited to those mentioned here) is likely to offer the best eventual solution.

4. A spatial awareness technique in more detail

We will now focus in detail on the use of spatial awareness techniques, as this is an area in which we have expertise and which has received little attention in the context of speech and gestural input. We are also particularly interested in this approach because it exploits the spatial characteristics of virtual reality in a way that is not possible with most other kinds of computer system (e.g., 2D windowing systems).

The essence of spatial awareness techniques is that each object (human or application) in a shared virtual environment may have a different quantifiable level of *awareness* of each other object. Levels of awareness represent the mutually negotiated importance that objects attach to one another and can be used to control their mutual perception. For example, awareness levels might drive the volume of an audio channel, the level of detail of a graphical rendering or the quality of service (e.g image size, resolution and frame rate) of a video channel. Awareness in spatial systems will most often depend on spatial properties such as proximity and orientation (although might also involve non-spatial properties too). For example, one object moving or turning towards another may increase the other's awareness of itself and so cause it to be heard at greater volume. Thus, objects move around in space in order to negotiate mutual levels of awareness and so control their interaction.

There are many possible ways to use spatial relations to calculate mutual awareness. The most widely used technique in virtual environments is *distancing* for controlling graphical level-of-detail (although this is not usually formulated this way). Alternatively, at VRST'94 we presented our own more powerful awareness model [Benford et al., 1994] such that:

- objects only become mutually aware on *aura* collision, i.e. upon the intersection of regions of space which define the scope of their presence and interest in the virtual environment;
- the receiving object of any utterance influences its awareness of the transmitter through its *focus* - a multi-valued spatial field (a function over space) that models its allocation of attention across space;
- the transmitting object influences this awareness through a complementary field called *nimbus* which models the projection of its utterances into space;
- various additional *adapter objects* can be used to change the shapes of focus and nimbus and so influence awareness (e.g. a virtual podium acting as an aura and nimbus enhancer).

In general an object will have a separate aura, focus and nimbus for each medium in which it interacts. Thus, objects move around space using their aurae, foci an nimbi to influence mutual awareness of other objects and so control interaction.

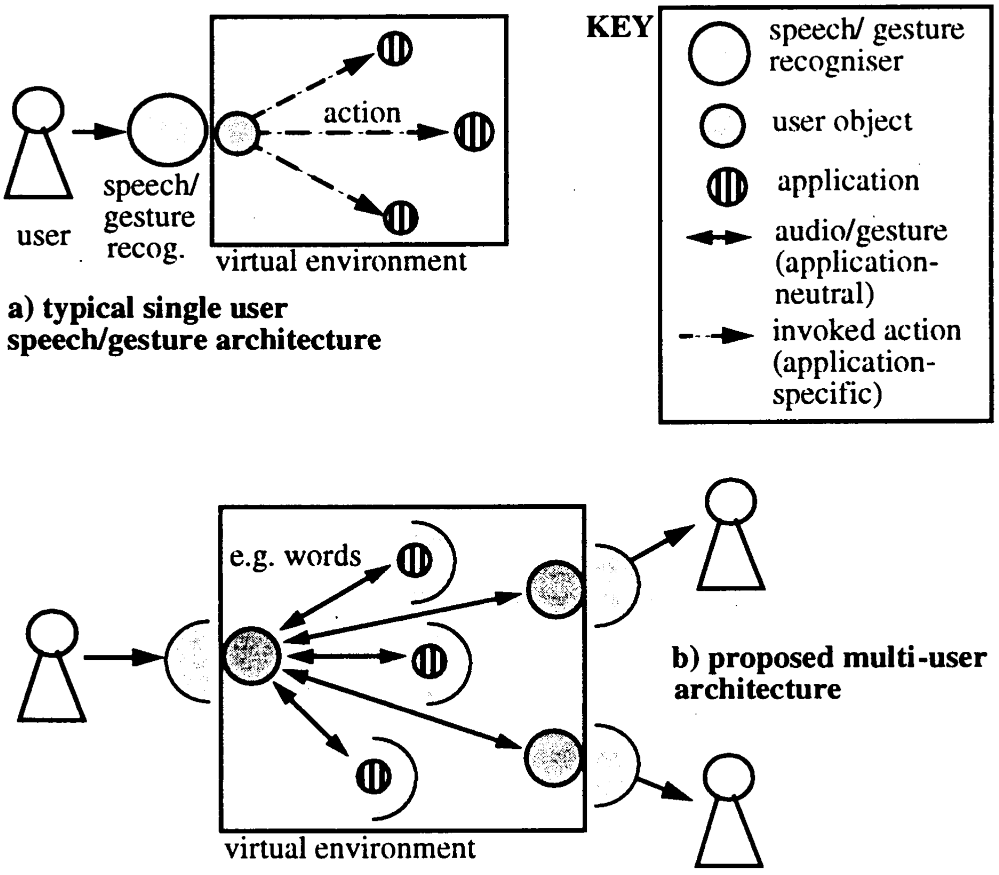
Applying such a spatial awareness model to the control of application objects by speech and gesture involves five main steps:

- defining a suitable awareness model and providing the infrastructure to support it.
- specifying and coding *policies* for applications which describe how they are engaged, shared and disengaged according to awareness levels and relationships.
- providing mechanisms so that an application can give feedback to humans about when its attention has been engaged and disengaged; this is in much the same way that turntaking in conversation involves natural (and often subtle) mechanisms for confirming the engagement of attention between humans.We argue that such explicit representation of the engagement of attention will be a key aspect of embodying non-human applications in virtual environments.
- providing mechanisms or protocols so that applications can determine and respond to other applications' awarenesses of a given speaker so that they can decide which applications should respond when there is more than one alternative.
- placing the application-specific parts of the code (or data) which interprets speech and/or gesture inside individual applications as opposed to within each users input agent at their interface. In other words, speech and gestural input should be broadcast to humans and applications alike in an application-neutral form, and its recipients should decide how to handle it.

Figure 2 compares our proposed approach with "traditional" approaches. With the traditional model, a single user's speech or gesture is fed to a recogniser attached to their interface. The recogniser then outputs actions to be invoked on specific applications in the virtual environment using the techniques from figure 1. Under our approach, a user's speech or gesture may be part-processed and then transmitted (via the user object at the interface) to all sufficiently aware (and therefore interested) users and applications. These applications then complete the recognition process. Their internal policies together with awareness levels allow them to decide when and how to react to the input they receive.

We have introduced the term policies to refer to code which helps an application decide when and how to react to speech or gesture based on knowledge of its (and other applications') awareness of the speaker. A wide range of such policies are possible. For example, an application's policy may be:

- Respond to any utterance where the application's awareness of the speaker is above some threshold. In other words, only respond when you are confident that it is you who is being addressed (e.g. perhaps when you are in the speakers nimbus). Thus, with this policy, an application might be engaged by a user moving or turning towards it and disengaged by them moving or turning away. This policy would allow several people to control an application simultaneously.
- Respond only to the user of whom the application is most aware. This policy implements a simple floor control mechanism whereby the application can only be controlled by one user at a time. Users would then use space to negotiate access to such an application (e.g. people might vie to get closest to the application), resulting in natural behaviours such as queuing, scrumming and jostling (common forms of human negotiating behaviour for accessing resources in shared spaces).
- Respond if the application is most aware of the user (from several possible candidates). This provides a simple way for applications to decide among themselves which is the most appropriate to engage a user. For example, the nearest available application might respond.



a) typical single user speech/gesture architecture

b) proposed multi-user architecture

Figure 2: Comparison of traditional and proposed speech interaction models

These three examples demonstrate the use of spatial awareness techniques to deal with problems of mixing commands and conversation and also of multiple humans controlling multiple applications. In particular, we can address all four of the specific issues raised in section 1 above:

- An application distinguishes commands from overheard conversation through the association of different awareness values with each utterance.

- Different internal policies allow multiple applications to decide how to respond to a command (e.g. should all of them or just one of them?)
- Different internal policies might allow an application to be simultaneously commanded by many people (e.g. a "threshold" policy) or restrict its use to one person at a time (e.g. a "most aware" policy)
- The broadcast of commands allows a user to control more than one application at a time and the use of aura and nimbus to scope this broadcast allows the use to control the extent (at least spatially) of the applications in a group.

Of course, we can imagine extending these policies in many ways including the addition of non-spatial attributes representing history, temporal and security issues.

5. A demonstration implementation

We now present a demonstration of the spatial awareness technique realised within the MASSIVE collaborative virtual environment. MASSIVE is a VR tele-conferencing system which implements the spatial awareness model and its concepts of aura, focus, nimbus and adapters to manage conversation between multiple participants in a shared virtual space [Greenhalgh and Benford, 1995]. Users of MASSIVE can communicate using three media, each of which is awareness driven:

- The graphics medium allows people to see each other and the space which they inhabit. Awareness in the graphics medium is used to control the level of detail of an object's graphical rendering.
- The audio medium supports real time conversation between users. Awareness levels in the audio medium are used to control the volume of audio information.
- The text medium provides a 2-D map like view of a virtual space using only ASCII characters and supports the exchange of text messages between participants. Text awareness levels are used to control how a text message is displayed (from the full text to a notification that something was said).

Users can switch between preset forms of focus and nimbus: *narrow,* such that full awareness requires close-up face-to-face positioning; *normal,* such that maximum awareness is obtained in a generally conical area projecting in front of the user; and *wide,* such that maximum awareness is obtained in a generally spherical region surrounding the user. Users can also change both the distance and conical angle parameters of focus and nimbus fields while using the system.

Another interesting feature of MASSIVE is cross-medium embodiment of text and graphics users. More specifically, text-only users may appear in the graphics medium and vice versa, allowing users of radically different types of terminal equipment to interact within a shared virtual space. Users with full audio capabilities can share the same space as audio-less users; the following discussion of the demonstrations indicates how this incompatibility can be circumvented.

We now present two examples of the use of the spatial awareness model to support interaction with non-human objects in MASSIVE. As the focus of our work has not been directly on mechanisms for interpreting speech and gesture, all of the following examples use the text medium (typed text) for communication with objects. This is sufficient to demonstrate our approach and the techniques are easily extended to the audio and graphical media given suitable speech and gesture recognition systems.

5.1. Text-to-graphics Adapter

The first demonstration that has been implemented is a graphical object which resembles a large chalk-board. This object has its own aura and focus in the text medium. When it becomes sufficiently aware of a user in the text medium it will take any messages which they type and scroll them onto the graphical board. The message selection policy is a simple awareness threshold - all high awareness messages (from whatever source) are copied to the board while all low awareness messages are ignored. Figure 3 shows a user compiling a meeting agenda on the board. The left-hand image is the user's graphical view while the right-hand image is their text view - text messages appear at the bottom of the window. The normal settings for focus and nimbus mean that the board will only respond to a user who is standing directly in front of the board and facing it. If they turn away - to address other meeting participants - then their text messages will be ignored by the board. Space-permitting, any number of users can engage the board simultaneously and their messages will be interleaved in the order they are received.

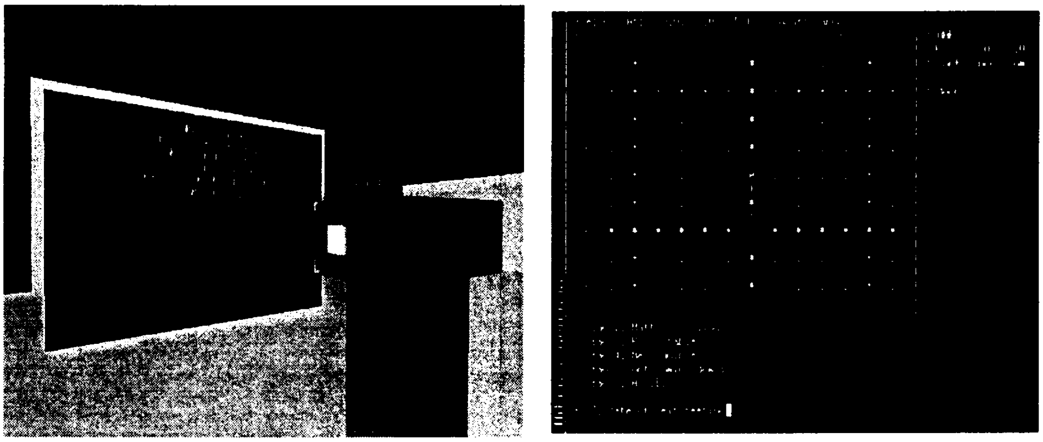
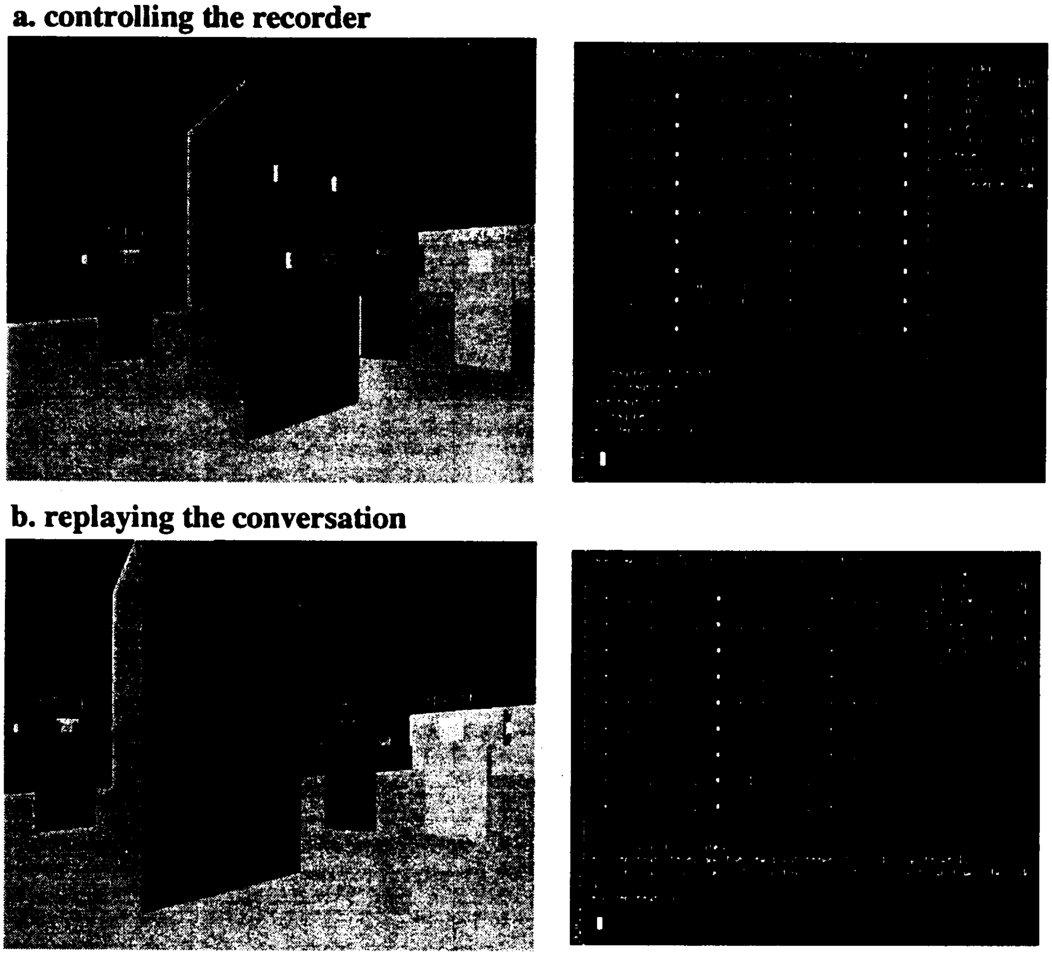


Figure 3: A user compiling an agenda on the message board

5.2. Text recorder

The second demonstration is a text recorder application. This is analagous to an audio tape recorder but, instead, records text messages (such as those displayed by the massage board, above) and can resend them at a later time. Additionally, unlike a normal tape recorder it is controlled using text commands. As for the message board, its policy is based on awareness thresholds. However in this case messages received with low awarenesses are recorded for subsequent playback (if in record mode) while messages received with high awarenesses are considered to be potential commands. Thus, the text recorder application is required to process text messages differently depending on whether it believes that they represent commands to be obeyed or conversation to be recorded.

So, in normal use, a user engages the recorder by standing in front of it and facing it. The recorder changes its (subjective) appearance according to whether it is engaged to accept commands or not: when it is ready to accept commands the user can see a simple face on the recorder. Figure 4.a shows a user starting the recorder while the other meeting participants look on. The user issues commands such as "play", "record" and "stop" to control the recorder (of course, these same words could easily occurr in the conversation to be recorded!). If the user then turns away to join the main conversation then the face disappears and their subsequent messages will be recorded. The recorder's colour indicates to all onlookers whether it is recording or not: it appears red when it is recording and green when it is not. The same user - or any other user for that matter - may subsequently engage the recorder and stop it recording or replay the recorded conversation, etc. Replayed messages are broadcast over the text medium in the same manner as they are received. Figure 4.b shows the recorder replaying a conversation.



a. controlling the recorder

b. replaying the conversation

Figure 4: Using the text-recorder in a meeting

5.3. *Discussion*

These demonstrators are quite simple: they use only the text medium and might have been implemented in other ways (e.g. using direct manipulation to control the recorder). However, they demonstrate the feasibility of using spatial awareness mechanisms to mediate natural language input for controlling application objects in collaborative virtual environments. In fact, closer examination of these examples reveals a further advantage offered by this approach. Consider the effect of moving the text recorder object sufficiently close to the board object so that the board becomes highly aware of the recorder. In this case, any recorded conversation that is played back on the recorder will be automatically displayed on the board. In other words, by positioning objects in appropriate spatial relationships we can cause them automatically to react to each other, thereby chaining them together - creating more complicated composite objects. This is illustrated in figure 5 - the replayed conversation is being displayed on the message board for graphical users to observe.

As a more interesting (hypothetical) example, imagine a general speech to text converter for MASSIVE (i.e. one that receives audio and broadcasts text). Just leaving this around for other people to use would immediately make it possible for any text driven object to also react to audio speech commands, without that object having to be at all audio capable and without the

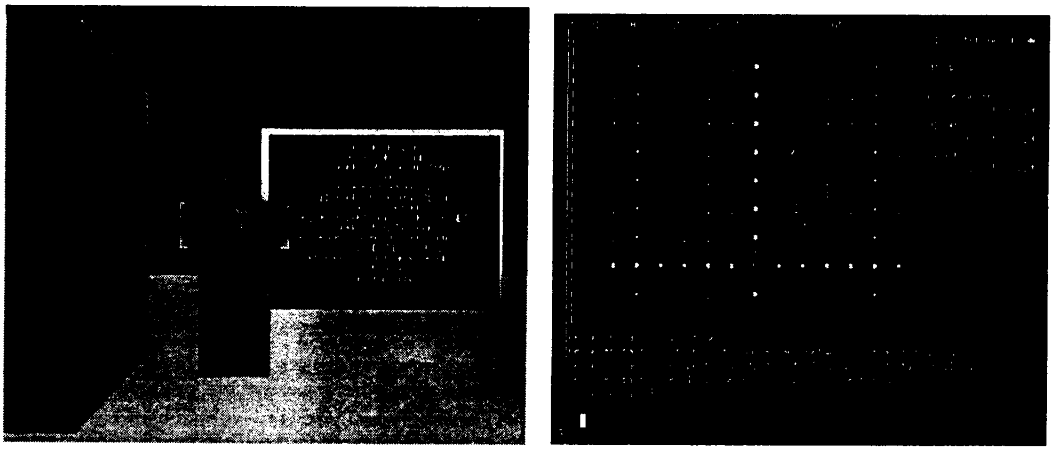


Figure 5: Combining the text-recorder and message board

needing to change the users own audio input code in their interface (indeed, absolutely no coding would be needed for the user. to make use of this converter in a powerful way).

6. Conclusions

The first conclusion of this paper is that the use of speech and gestural input techniques in collaborative situations needs to take account of the broadcast nature of visual and auditory media. More specifically, interesting issues arise in an environment where utterances may be received by many human and non-human objects:

- how do speech and gesture aware objects distinguish commands intended for themselves from overhead conversation?
- how do we resolve situations in which multiple people command multiple objects over a broadcast medium?

To address these issues, one must recognise that in order to participate in broadcast media objects must be empowered with a degree of "social awareness" in addition to the normal requirements associated with speech and gestural input.

We have picked out three particular approaches to this problem: the use of explicit selection mechanisms for switching between command and conversational modes; the use of content based triggers; and the use of spatial awareness mechanisms.

Given our previous experience and the highly spatial nature of VR systems, we have focused on the last of these. In particular, we have introduced our own spatial awareness model from previous work on spatially mediated conversation and have shown how, by associating objects with a variety of internal decision policies, even simple applications could support a range of control mechanisms. We then presented two simple demonstrators implemented in the MASSIVE collaborative virtual environment.

In conclusion, in complex multi-user virtual environments it is important to give non-human objects a sense of social awareness; we propose the use of spatial awareness techniques as one way of achieving this.

7. References

Allport, A., Rennison, E., and Strausfeld, L.," Issues of Gestural Navigation in Abstract Information Spaces", In *Human Factors in Computing Systems - CHI'95 Conference Companion*, (short paper), pp206-208, Denver, CO, May, 1995, ACM Press.

Appino, P.A., Lewis, J.B., Koved, L., Ling, D.T., Rabenhorst, D.A., and Codella, C.F., "An Architecture for Virtual Worlds", in *Presence*, 1(1), Winter 1992, MIT Press.

Benford, S., Bowers, J.,Fahlén, L.E. and Greenhalgh, C., "Managing Mutual Awareness in Collaborative Virtual Environments", *In Proc. ACM SIGCHI conference on Virtual Reality Software and Technology (VRST'94)*, (Singapore,1994).

Blau, B., Hughes, C.E., Moshell, J.M. and Lisle, C., "Networked virtual environments", *Computer Graphics 1992 Symposium on Interactive 3D Graphics*, 1992, p. 157.

Bolt, R.A., "Put-that-there": Voice and Gesture at the Graphics Interface, in *ACM Computer Graphics*, 14 (3), July 1980.

Godereaux, C., Diebel, K., El-Guedj, P.O., and Nugues, P., "Interactive Spoken Dialogue Interface in Virtual Worlds", in *Proc. of the CSCW SIG Seminar on Linguistic Concepts and Methods in CSCW*, (London, 1994).

Greenhalgh, C. and Benford, S., "MASSIVE: A Collaborative Virtual Environment for Tele-Conferencing", *ACM Transactions on Computer-Human Interaction (TOCHI)*, in press.

Karlgren, J., Bretan, I., Frost, N., and Jonsson, L., "Interaction Models, Reference and Interactivity in Speech Interfaces to Virtual Environments", in *Proc. 2nd Eurographics Workshop on Virtual Environments - Realism and Real-time*,(Montecarlo, 1995).

Kreuger, M.W., *Artificial Reality II*, Reading, MA.: Addison-Wesley Publishing Company, 1991.

Papper, M.J., and Gigante, M.A., "Using Gestures to Control a Virtual Arm", in *Virtual Reality Systems*, London: Academic Press Ltd., 1993, pp. 237-246.

Sturman, D.J., and Zeltzer, D., "A Survey of Glove-based Input", in *IEEE Computer Graphics and Applications*, 14(1), January 1994, pp. 30-39.

Wexelblat, A., "Natural Gesture in Virtual Environments", in Proc. *Virtual Reality Software and Technology (VRST'94)*, (Singapore, 1994)

Acknowldegments

This work has been sponsored by the UK Engineering and Physcial Sciences Research Council (EPSRC).