THE CONTRIBUTION OF 3-D SOUND
TO THE HUMAN-COMPUTER INTERFACE

by

Mark Aaron Vershel

S. B., Massachusetts Institute of Technology
(1980)

Submitted in Partial Fulfillment
of the requirements for the
Degree of
Master of Science in Visual Studies
at the
Massachusetts Institute of Technology
June, 1981

Signature of Author . . . .
Department of Architecture
May 8, 1981

Certified by. . . . .
Nicholas Negroponte, Professor
of Computer Graphics
Thesis Supervisor

Accepted by .
Professor Nicholas Negroponte, Chairman
Departmental Committee for Graduate Students

THE CONTRIBUTION OF 3-D SOUND TO THE HUMAN-COMPUTER INTERFACE

by

Mark Aaron Vershel

Submitted to the Department of Architecture on May 8, 1981, in partial fulfillment of the requirements for the degree of Master of Science in Visual Studies.

ABSTRACT

Sound inherently has a spatial quality, an ability to be localized in three dimensions. This is the essence of 3-D, or spatial, sound. A system capable of recording sounds as digitized samples and playing them back in a localized fashion was developed in the course of this research. This sound system combines special hardware and interactive software to create a system more flexible and powerful than previous systems.

The spatial qualities of 3-D sound contribute to man's ability to interact with sound as data. An application which capitalized on these qualities was developed, allowing the user to interact with 3-D sound in a spatial environment. This application, called the Spatial Audio Notemaker, was not unlike a bulletin board, where the paper notes were recorded messages and the bulletin board was the user's environment. Using the Spatial Audio Notemaker, exploration into the manipulation of 3-D sound and the necessary interaction (using voice and gesture) and feedback (both visual and audio) to aid in this manipulation was accomplished.

Thesis Supervisor: Nicholas Negroponte

Title: Professor of Computer Graphics

The work reported herein was supported by the Advanced Research Projects Agency of the Department of Defense, under contract number MDA 903-77-C-0037.

CONTENTS

ACKNOWLEDGEMENTS

I would like to thank: Nicholas Negroponte, my thesis advisor, for his many suggestions throughout the project; Andy Lippman and Steve Gregory, for their suggestions in the writing of this document; Chris Schmandt and Eric Hulteen, for their criticisms and friendship; Bill Kelley, for his work on the sound box hardware; and the rest of the Architecture Machine Group, for providing a fruitful atmosphere for learning.

INTRODUCTION

In a world where research continues into making computers faster and smaller, there is some attention devoted to making computers easier to use. This is the thrust of those few involved with the human-computer interface, research which deals with making computer input and output more compatible with human senses. The interface between man and computer should deal less with printed data and more with data which is transmitted with visual cues, gestures, and sound.

Imagine an interface which is spatially oriented. The user can point within his environment and the visual cues center around a two-dimensional image of a three-dimensional world. This study is an attempt to integrate sound into that environment. Sound can be put into two categories: 3-D sound and simple sound. Sound with spatiality is referred to as 3-D sound. Spatial sounds can be localized in space using an eight speaker system (with one speaker in each corner of a room). In contrast, simple sound has no spatiality. Although it is possible to localize simple sound in a line or a plane (using two or four speakers), this localization does not reflect the spatiality of a three-dimensional world.

Realism is accomplished using 3-D sound. Simple sound does not reflect the spatiality of sound found in the real world. Man deals iwth 3-D sound in that world constantly, where the sources of sound are localized in one's environment. It is not difficult to determine the source of a sound when asked to do so. The use of 3-D sound at the human-computer interface is necessary, therefore, to make the transmission of audio data more realistic. This realism allows the user to assimilate data more easily than if simple sound were used because the characteristics of the real world are reflected using 3-D sound.

When dealing with localized sound, the use of 3-D sound allows faster processing of information. Using simple sound, the user has to receive localized sound with two senses. The user hears a sound and also has to be given the location of the sound. For example, the sound of a car approaching from behind the user would be represented as the sound of a car and a visual telling the user that the car is behind him. Using 3-D sound, the user is given the location information concurrently with the information within the sound. For example, the user would hear the car actually coming from behind him.

Thus, 3-D sound aids the human-computer interface by allowing the user to process both the information given by the sound and its location at the same time, using only the sense of sound. Using simple sound necessitates the user coordinating information from two sources. Since the user would not have to process two information sources when using 3-D sound, it is faster to process.

Localized sound can be used to divert the user's attention to a point in space. This is possible because of man's ability to localize a sound's source in his environment.

Therefore, 3-D sound can be used to get a user's attention (e.g., "look here"), while simple sound can only describe where to look ("look to your left"). When using simple sound, the user must pay attention to the sound as well as process it ("where should I look?"). The use of 3-D sound gets the user's attention not by requiring processing of information but by actually diverting his attention to the source of the sound. Once again, the advantage of spatial sound over simple sound is seen.

Man also has the ability to process audio information coming concurrently from different sources by "tuning" in one source and ignoring the others. This is the so-called "cocktail party" effect (named for an ability to pick out conversations from several occurring at the same time), and it accounts for the user's ability to process streams of audio data by shifting attention. Simple sound cannot capitalize on this effect to the degree that 3-D sound can, because it does not have spatiality. However, 3-D sound can be used to place sounds in the user's environment, thus allowing the user to process information from several sounds at once or to "skim" the information presented in order to find what is most important. This skimming process is a large contribution to the user's ability to interact with his environment.

Another contribution of 3-D sound is that it helps the user to organize audio information. Simple sound allows little ability to organize data in the user's environment, since it does not reflect that environment's spatiality. Using 3-D sound, the user can organize the space in which he is working as well as annotate that space. 3-D sounds can surround the user; they fill the spatial environment with information that can be processed concurrently. The user can organize that environment to any degree he wishes and with great ease; this could not be done with simple sound since it is so limited.

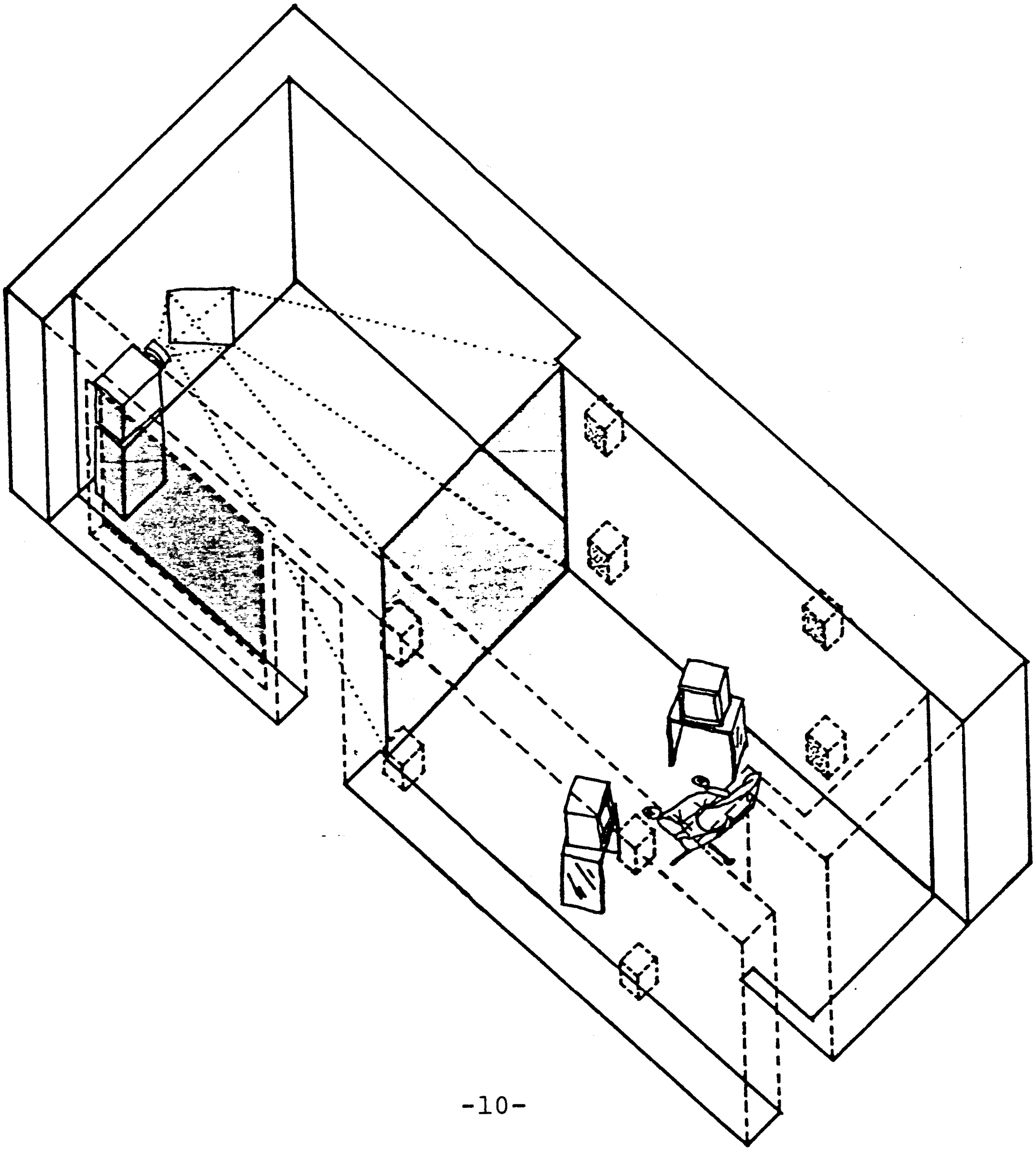
Not only does 3-D sound contribute to the user's ability to organize his environment, but it also allows the user to treat sound as a randomly accessible object. To treat simple sounds as objects, the user would need some sort of random access graphic which would give each sound an identity. The user would have to interact with this graphic to work with the sounds. However, 3-D sounds have identity by themselves since they have a virtual source. Rather than manipulating simple sounds with a graphic, the user can interact directly with 3-D sounds in the environment in which the information exists.

In summary, 3-D sound has many characteristics which make it more usable at the man-computer interface than simple sound. They key to these contributions is the spatiality which characterizes 3-D sound. This spatiality means more realism at the interface, faster processing of information, an ability to act as an attention getter, an ability to process multiple inputs, better organization of audio information, and an ability to treat sounds as data objects. In order to explore these contributions, an application must be developed which uses 3-D sound in a spatial environment and allows interactive control of the audio data.

In this study, the environment is the "media room." It is a room with a chair in the center, monitors to each side of the chair, speakers in the eight corners of the room, and a rear projected screen which takes up the entire front wall (see figure 1). Note the absence of computer terminals. The user doesn't sit in front of a terminal to control the manipulation of data, but rather sits in an environment in which the data to be explored exists.

The data is 3-D sound data and the control of this data involves voice and gesture recognition. Voice recognition is accomplished using the NEC connected speech recognizer, which allows a 120 phrase vocabulary which must be trained by the user. Each phrase must be no longer than about two seconds. Gesture recognition is accomplished by using a radiator-sensor system (made by Polhemus Navigation Systems) which locates the sensor in space using the magnetic field transmitted by the radiator, which is stationary. Both the position and the attitude of the sensor, which the user wears on his wrist, are returned to the user. This voice and gesture recognition at the human-computer interface has recently been discussed by Dick Bolt (1) in a paper concerning another project (called "put-that-there") done here in the Architecture Machine

FIGURE 1: A sketch of the media room



Group at M.I.T. Further documentation on the devices mentioned above can be found there.

Before any study of 3-D sound can be attempted, there must be a method for presentation of 3-D sounds. The current sound system as well as its historical progress should be explored to put the study of the contribution of 3-D sound to the human-computer interface in context with the environment in which 3-D sound will be examined.

BACKGROUND

The Sound System

The sound system is made up of a multitude of software routines which control specialized hardware and manipulate data located on magnetic disk.

The software for the sound system runs on an Interdata 7/32 minicomputer with 512K bytes of primary memory. The operating system is Magic6, an in-house system which includes a PL/1 compiler, an assembler, an editor, and various system routines for doing system i/o. The operating and file system are located on a CDC Trident disk; the sound data is located on a 2314 disk. Additionally, there is a piece of equipment called the "sound box."

The sound box is a hardware device, built and designed in-house, consisting of a group of digital-to-analog (D/A) and analog-to-digital (A/D) converters. The A/D converters are used to sample the input sound and convert it to digital data (record mode). The D/A converters are used to take this digital sound data and convert it to an analog signal (playback mode).

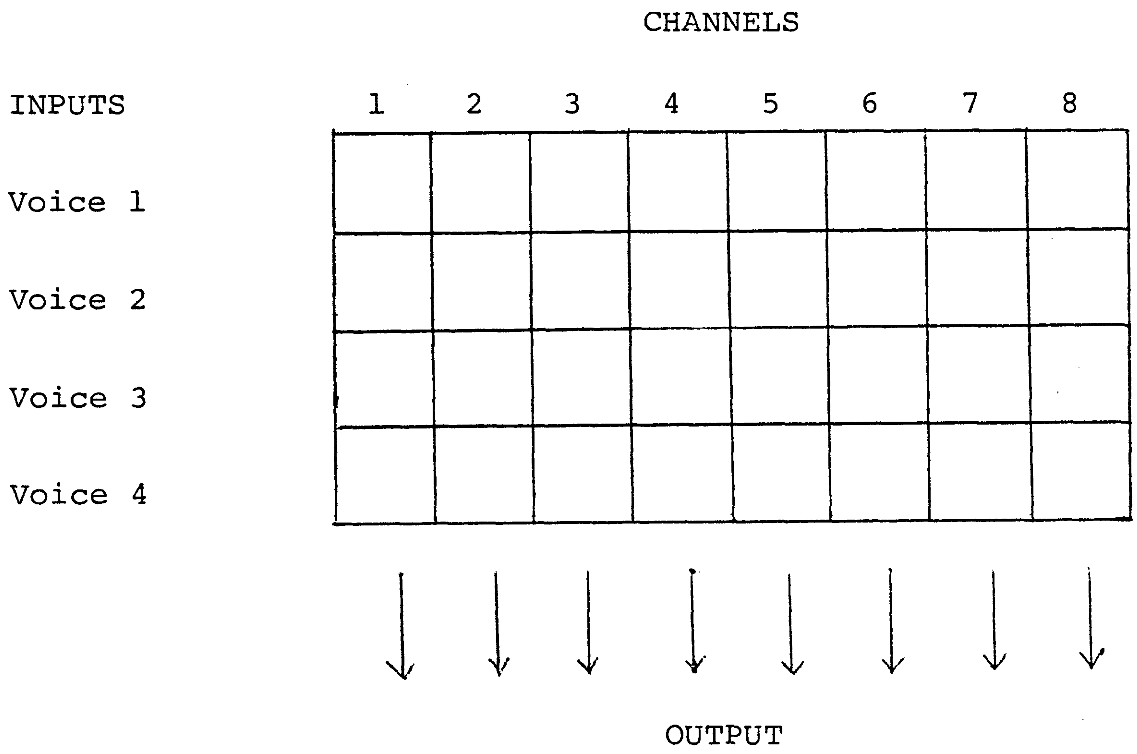
The input signal is sampled at discrete intervals in time at a rate dictated by the user. The sampling theorem states that a bandlimited analog signal may be recovered exactly from its samples if the sampling rate is at least twice the highest frequency contained in that signal. In the sound box, the sampling rate is normally 8000 Hz, thus requiring a low-pass filter on the input signal that has a cutoff frequency below 4000 Hz. Reconstruction is accomplished by D/A conversion of the samples followed by an interpolation filter (a filter that takes the impulse output of the D/A converter and interpolates between impulses to create a smooth signal) with a similar cutoff.

Each sample produces an 8-bit byte which will provide a value of 0 to 255. A grounded dc signal leads to bytes of value 128. The use of 8 bits per sample leads to a signal-to-noise ratio of about 50 dB.

A total of four sounds can be played concurrently through the sound box. Each sound is played through a "voice"--a hardware device in the sound box. Each voice can be played through 8 channels, leading to a volume array of 32 amplitudes (see figure 2). Each amplitude can be set to a value of 0 to 255 (full off to full on). For each of the eight channels, the sound box sums the weighted digitized signal for each voice and produces a voltage for that channel which is passed to the amplifier driving that channel's speaker.

Each voice has a voice clock. These clocks are controlled by a 1 MHz master clock. Controlling this clock allows control of the voice clocks in that turning off the master clock will disable all of the voice clocks. Thus, if the user wanted to start voices 2 and 3 simultaneously, he would enable the clocks for voices 2 and 3 and turn on the master clock.

The voice clocks are used to specify the rate at which the digital sound data is fed into the D/A converter for



Each column is added and sent to output amps.

FIGURE 2: Configuration of the amplitude array

each voice. The normal rate is 125 microseconds, which translates to a sampling rate of 8000 Hz. Changing this clock rate will speed up or slow down the sound. For example, a clock period of 100 microseconds will make the sound speed up. If the digitized data represents a human voice, then the played voice will have a "Mickey Mouse" effect. The reverse is true with clock periods greater than 125 microseconds.

Communication between the sound box and the processor is under interrupt control. For both the record and playback modes, an interrupt is generated after 64 bytes have been transferred. In both cases, the transmission of data is between a core buffer in the processor and the internal memory of the sound box, which consists of two 64 byte buffers.

The reasoning behind the use of two 64 byte buffers is that the interrupt handler can do transmission of data in bursts of 64 bytes. If this was not the case, then a new byte would have to be transmitted to or from the sound box every 125 microseconds to avoid a loss of coherency in the sound data. This would not allow the processor to do anything else but process the sound box data requests. With the use of two 64 byte buffers, the processor can process the interrupt generated by the sound box by writing or reading 64 bytes within 8 milliseconds and still maintain the coherency of the data. This will allow the processing of other events concurrently with the sound box and thus allow the user to accomplish other tasks while running the sound box.

The processor uses a wired segment in core for storing data for transmission to and from the sound box. A wired segment is a segment in core that will not be swapped. It is a place to store data where the interrupt handler will always find it without having to swap that segment into core. This segment is 60K bytes. (Note that 1K byte of memory is equal to 1024 bytes while 1 KHz is equal to 1000 Hz.) Hence, with a clock rate of 125 microseconds, the core buffer of 60K bytes will contain enough sound data for 7.68 seconds (60K bytes/8000 samples per second) of sound. Obviously, a method for storing sound data on a disk is necessary for storing digitized sounds longer than 7.68 seconds.

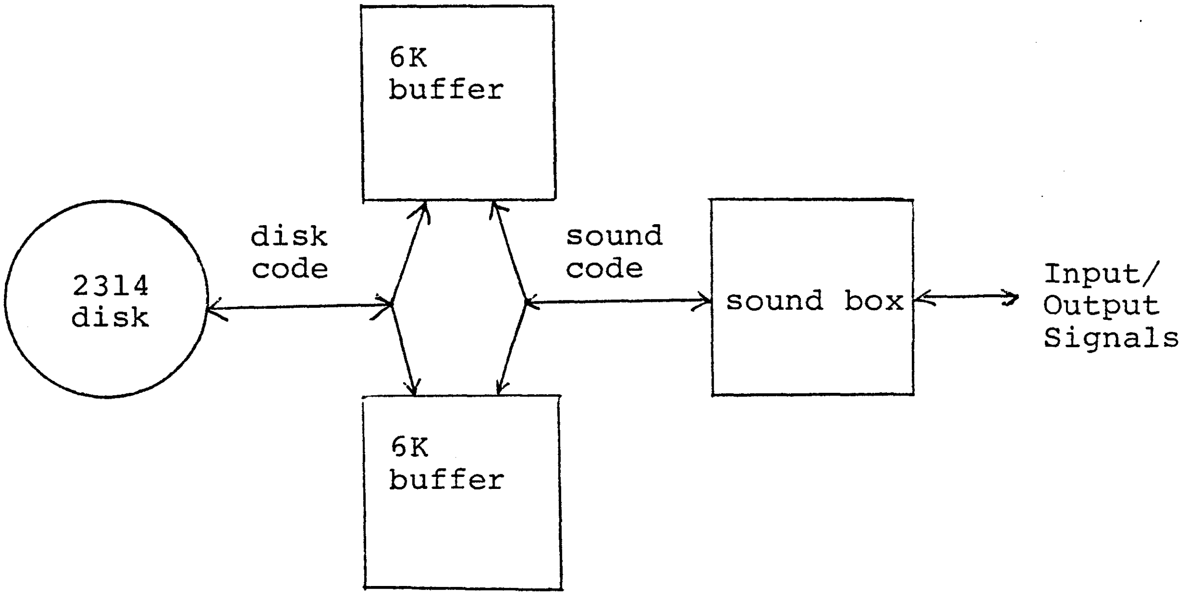
In order to play sounds from disk, the wired segment must be divided into four buffers, since there are four voices. Each of these buffers must be a "double buffer"--a buffer of two equal parts (each part is 6144 bytes). In the same fashion as with the two 64 byte buffers of the sound box, data from the sound disk is written into one half of the double buffer while the sound interrupt handler writes the data from the other half into the sound box. When one half of the buffer is emptied to the sound box, that buffer is filled while the other one is played, and so on. This double buffering scheme allows the user to process other tasks while playing sounds from disk because the writing of data from disk to core can be accomplished in writing one block asynchronously, similar to the methods used in transferring data from the wired segment to the sound box mentioned above. Recording to disk is accomplished with the same double buffer and the data moving in the opposite direction. Figure 3 provides an overview of the sound system.

Further and more detailed documentation concerning the sound system, the data bases used, and the routines which control the sound system and its data bases can be found in the Sound System User's Manual (Vershel, (5)).

Previous Work

The first work done on the sound box keyed on localization of sound in a plane using four speakers. This work was done by David Gorgen (2). Gorgen determined that although there were several methods of localization, the most appropriate one for use with loudspeakers was localization by varying intensity. The then popular method of varying interaural time delay (the difference in time between the arrival of a sound at each ear) was found to be inappropriate due to the critical positioning of the user's head in a loudspeaker environment.

Gorgen made several assumptions about localization, the most important being: (1) vertical localization is independent of horizontal localization, (2) once a sound is localized, increasing the volume of the speakers involved (keeping the proportional weightings of the volumes constant) will lead to the sound being localized in the same place but seem closer since it will be louder, (3) using multifrequency ("white") noise for calibration will make the



12K byte buffer
within wired segment

FIGURE 3: An overview of the sound system

localization method valid for arbitrary frequencies.

With these assumptions, Gorgen accomplished localization of sound using a calibration table made by adjusting the volumes of the speakers until a sound was localized at each calibration point, and then saving the weightings of the speakers for each point. Hence, to localize a sound, the user would look in the stored table and read out the weightings of the speakers for that point. Interpolation was used for localization points which fell between calibration points but Gorgen found that it was difficult to sense differences in direction of less than about one foot.

Jonathan Hurd (3) expanded this system to eight speakers so that localization could be accomplished within three dimensions rather than just within a plane. Hurd's major contribution was to deal with the sound system as a whole rather than to key in on localization. He created a system which allowed recording to magnetic disk rather than just to core.

In this environment, he explored the tradeoffs between the sampling frequency and the amount of data generated. The faster the sampling frequency, the higher the bandwidth the resulting output signal (once it was digitized and played back) will have. However, the higher sampling frequency means that proportionally more data will be generated. For example, a ten second speech sampled at 8000 Hz will generate 80,000 bytes of digital data. This speech will have a bandwidth of 4000 Hz, roughly that of a telephone circuit. However, 4000 Hz is quite poor for music. A higher-fidelity 10 second recording at a sampling rate of 20,000 Hz (a bandwidth of 10,000 Hz) will generate 200,000 bytes of data, two and a half times that at the 8000 Hz rate. Since the speed of the processor is limited, four voices could not be played concurrently at a clock period of 50 microseconds (20,000 Hz sampling frequency). However, this can be accomplished easily with a clock rate of 125 microseconds (8000 Hz sampling frequency). Most of the work done here is with voice data, so high fidelity recordings would be nice, but usually unnecessary. Music recorded with the current system sound much the same as music played over a telephone--disappointing.

Hurd created a disk system using the double buffer scheme mentioned previously. By breaking the 60K wired segment into pieces, he was able to play sound from and record sounds to disk. However, his original system had several problems and proved to need revision. Although the basic ideas remained, the new system, which was described in the beginning of this section, proved to be more reliable and more flexible.

Dave Moosher (4) expanded on Hurd's work by actually using the sound system in several interactive applications. He used an analytic model of localization rather than a calibration table like Gorgen's. This model treated sound with similar assumptions to Gorgen's, but dealt with eight speakers rather than four. Moosher also dealt with localization at a distance by applying the inverse square law to the decrease in volume as a sound is located further from the user. However, Mooser's applications dealt primarily with localized sound in a plane (needing only four speakers). Although the potential was available for localizing sound within the entire media room, no applications were developed.

Moosher did set the way for using sound as a data type. He created a library of sounds and showed that sounds fall into several categories which included both generated sounds (like sine waves) and recorded sounds. By treating sound as data, he was able to begin exploring the user's interaction with that data.

PROBLEM

In the introduction to this thesis, the reasons for using 3-D sound rather than simple sound were discussed as contributions of 3-D sound to the human-computer interface. Using the sound system explained previously, an application should be developed which allows the user to examine 3-D sounds and to explore the validity of these contributions.

There are three issues to be explored in the context of that application. The first is the ease of interacting with and controlling sound data which is spatially organized. The second is how the spatiality of sound is important in using sound as data; i.e., how does the user organize an environment consisting of spatial sounds. The third, although part of the first, is what feedback is necessary to aid in using 3-D sound. This issue of feedback is important to any human-computer interface, and thus deserves separate consideration.

To deal with the first issue, a system must be developed to explore manipulation of 3-D sound data. This system is the Spatial Audio Notemaker, a system which manipulates messages stored as 3-D sounds in the spatial environment of the media room. The modes of interaction are voice and gesture. Voice interaction will serve to tell the system what to do to the messages, while gesture interaction will serve to indicate which messages the system should address. With these modes of interaction, the Spatial Audio Notemaker should be useful in exploring the interactions of 3-D sound and the user.

The importance of the ability to locate sounds in space will be explored as well. There are two elements of spatiality of sound; one is the position of the sound in the room and the other is the distance of the sound from the user. The thrust of this exploration will be to see if the user positions sounds as a function of their characteristics (e.g., the topic of the message) or positions them in a random manner.

The issue of what feedback is necessary at the interface in order to use 3-D sound deserves special attention. Since the messages do not exist as physical entities, the user must be given some sort of visual cue for where the messages are located if the user is to manipulate them. Playing a localized sound will give the user a feeling for the position of that sound, but not all manipulations should involve playing the messages within the room.

There must also be feedback as to where the user is pointing within the room. This involves the creation of a 3-D cursor. Further visual feedback must include a method for indicating which notes are playing, which notes are being addressed by pointing, etc.

Besides visual feedback, there must be auditory feedback. The system should ask questions when it does not understand what the user is trying to do, as well as inform the user if something illegal is being attempted. Thus, the system and the user may have a conversation to clarify those commands that may need clarification. Additionally, the sense of localization achieved by playing each message is a feedback mechanism, one which hopefully plays some part in the ability of the user to work with 3-D sound.

Before these issues can be addressed, an effective method for localizing sounds must be found. As explained previously, some work has already been done on this subject, but no application really used localization to its fullest degree. Since localization of sound is essential to the workings of the Spatial Audio Notemaker, research must be devoted to localization.

LOCALIZATION

Past work here at the Architecture Machine Group has shown that the best method of localization of sound in an environment involving loudspeakers is to weight the volumes of the speakers proportionally for any point in space (Gorgen (2), Hurd (3), and Moosher (4)). Some analytical model is desired to set these proportional weightings so that no calibration method like Gorgen's is necessary. The calibration process would be too complicated when the number of speakers is increased to the current eight from Gorgen's four.

There are two methods that have been explored in this study. The first is a linear method which involves weighting the speakers (or output channels) such that the sum of the weightings is one. For example, a sound localized in the center of the room will have all the speakers on at 1/8 volume. The calculation involved with this method is simple; each speaker's weighting is the product of the differences of the position of the localized sound and the speaker in all three directions (left to right, top to bottom, and front to back).

Although this method is somewhat crude, it allows the user to localize sound in a general way. Sounds which are placed at the corners of the room are localized easily by the user. The user also gets a general sense of left to right, top to bottom, and front to back. However, this localization method is not accurate beyond a general localization. For example, sounds which are localized at a distance from the corners of the room do not maintain the same volume as those at the same distance from the user but near a corner. Thus, a sound which "orbits" the user does not maintain a constant volume. This proved to be somewhat confusing since this does not occur in the real world.

The second method capitalizes on the logarithmic characteristics of sound. This method involves much more calculation but since it reflects sound more realistically than this first method, it should be much better. Essentially, this model assumes the sum of the weightings is a constant. However, unlike the previous method, the power levels (measured in dB's) of each speaker are summed to maintain a constant power level for the sound. Hence, at given distance from the user, the total output power of the localized sound is constant. Although this method involves more calculation and is therefore slower, the increased realism gained by dealing with the actual characteristics of sound is quite sufficient to merit this increase. This method of localization is used throughout this project.

Note that, so far, both methods deal only with direction. In other words, the first goal of the localization routine is to place the sound in a direction from the user. Once this direction vector is determined, the volume of the sound needs to be adjusted to give the user a sense of where the sound is along that vector. This is accomplished by multiplying the proportional weightings by a factor which represents the decrease in volume as a sound is located further from the user. This factor should mimic the actual characteristic of sound as it moves away, i.e., the volume of a sound falls off as the square of the distance from the listener. Hence, moving a sound twice as far from the user will result in the sound having 1/4 volume. Note that this is only true in open air situations. Within the confines of a room the decrease will not be as dramatic due to the sound bouncing off the walls of the room.

This ability to localize sound along a vector is dependent on the user having some method of calibration. The user must know how loud a sound should be at a given distance. Otherwise, given another sound with lower volume, the user will not know if this sound is at the same distance but of lower volume, or of the same volume at a larger distance. For this reason, moving sounds are easier to localize (assuming the sound maintains its volume) because the user can compare one instance of the sound to the previous one.

Due to the limitations of dynamic range in the sound system, the fall off in volume is purposely lessened. The signal-to-noise ratio at larger distances would otherwise be unacceptable. In the case of the Spatial Audio Notemaker, the fall off is slight but enough so that the user can get a sense of whether a sound is close or far. The fact that all the sounds are at constant volume (since they are recorded by the user himself) makes this possible.

Localized sound has been the subject of several experiments in the course of this study. This research has shown that moving sound is easier to localize than stationary sound (especially when augmented by doppler shifts), that localization of tones at a signal frequency is very difficult (multifrequency tones should be used), and that voice data can be satisfactorily localized in a variety of applications (due to the fact that the human voice is made of many frequency components).

THE SPATIAL AUDIO NOTEMAKER

In order to explore the contribution of 3-D sound to the human-computer interface, a system must be designed so the user can manipulate 3-D sound using voice and gesture. Such a system is the Spatial Audio Notemaker (SPAM), which allows the user to record messages (of up to 20 seconds in length) and position them in the space of the media room. Once positioned, the messages or "notes" can be manipulated within the environment of the user. This manipulation consists of a set of commands which will be explained shortly.

The control structure of SPAM is similar to that of "put-that-there", the subject of a recent paper by Dick Bolt (1). This is due to the fact that both systems deal with voice and gesture at the human-computer interface. However, "put-that-there" was a system that allowed manipulation of graphical data by detailed description or gesture. That system dealt with data in a planar environment. SPAM deals with data in a spatial environment; that data is not graphical but audio. Thus, although there are some similar commands, SPAM has capabilities which "put-that-there" did not. SPAM does not attempt to incorporate all the interactions of "put-that-there" because to do so would be redundant. That system was an exploration into voice and gesture at the interface; SPAM is an exploration into 3-D sound and that interface.

The user sits in the media room chair. The monitors to the left and right are not used but the rear projected screen to the user's front has a computer generated graphic which depicts the media room in perspective (either as a projection or a virtual mirror). Within the graphic (which is a wire frame model of the media room) are rectangles which represent the notes that the user has previously recorded. Additionally, there is a transparent cursor which moves coherently. A drawing of a snapshot of this image can be found in figure 4. Further discussion of this graphic can be found in the next section.

The user wears a microphone that is connected to the speech recognizer and wears two Polhemus cubes (sensors), one on the wrist and one on the shoulder. By using two cubes, the user can specify a distance along the vector along which he is pointing. The Polhemus cube on the user's wrist is used to determine in which direction he is pointing. The distance between that cube and the cube on the user's shoulder is used to determine how far along the vector the user wishes to indicate. This shoulder-wrist distance is scaled to represent the shoulder-wall distance. Using this method the user can position notes within the environment and not just at the walls of the media room.

The user is now ready to use the system. Typical manipulations of the notes using voice and gesture interactions are examined below.

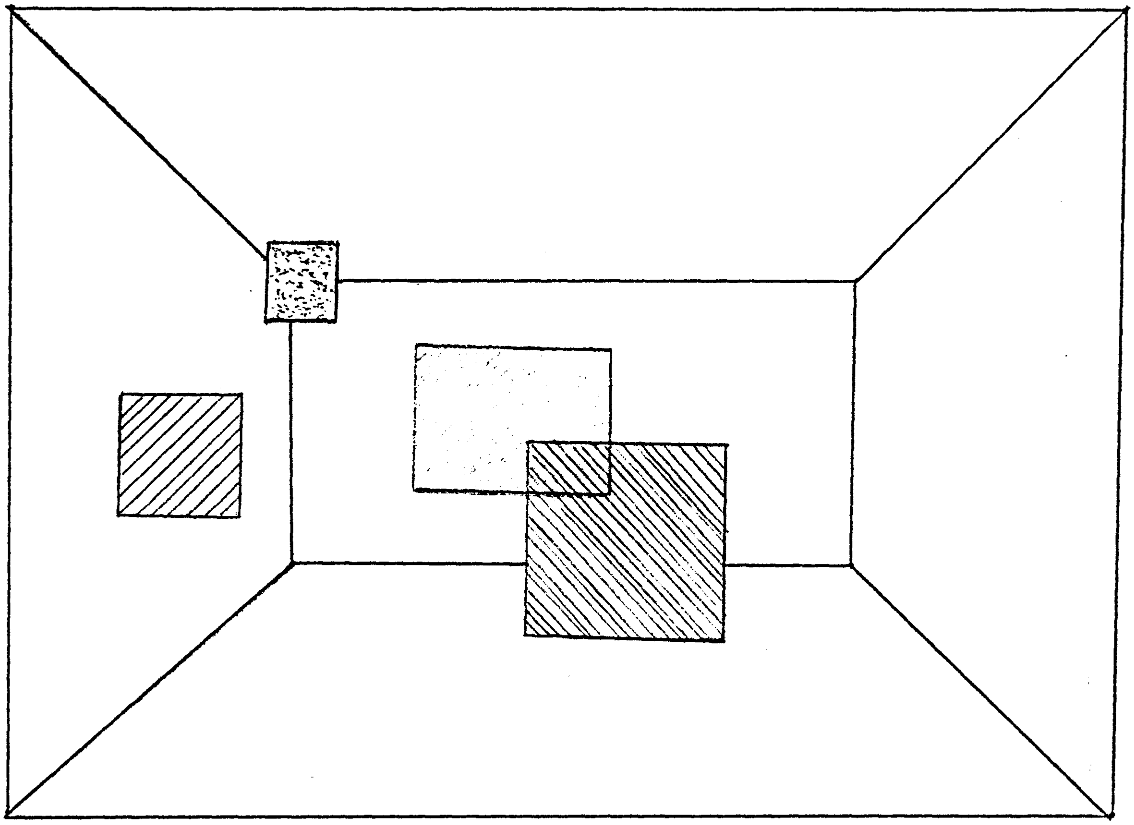


FIGURE 4: A sketch of the large screen display for the Spatial Audio Notemaker

The image shows the wire frame model of the media room, three notes (the different patterns represent different colors), and a transparent cursor (shaded rectangle).

"Peruse notes"

The command will put the user into a mode where whatever note the user points at will begin playing. Up to three notes can be played concurrently; each note will play once. This mode of operation (analogous to skimming) allows the user to quickly peruse all of the messages within the environment without having to separately command each note to play. Utilizing the "cocktail party" effect, the user may hear a note of interest, one which deserves more attention. In this case the user can give any other command to leave this mode, allowing the note to be played on its own.

"Play..."

When the user wishes to play a certain note, this command can be used. It has two forms:

"Play that note" or

"Play the red note", "Play the note to my left"

The first form assumes the user is pointing to a note while giving the command. If this is not so, the system will ask the user to specify which note it should play. The second form involves only voice specification of the note (either by color or position). If no note fits this description, then the system will again ask the user to specify a note (by description or pointing).

There are two types of interaction involved here. One assumes an interplay between voice and gesture, the other involves a vocal description. Both methods of specifying a note are used with the commands within SPAM. The user interchanges how a note is specified depending on the situation. This redundancy allows the user flexibility to express the commands in whatever form is best for that user. Sometimes, however, the user must point at the note; for example, if there is more than one red note (in the example above), then the user must point at one of them.

"Stop everything" or "Stop"

These commands allow the user to stop the notes that are playing. In the case of "stop everything", all the notes currently playing will stop. In the case of "stop", only the note specified will stop playing. If the user does not specify which note should be stopped and more than one note is playing, then the system will ask the user to indicate which note is to be stopped. If only one note is playing, then the system will assume the user wants to stop that note.

"Record..."

The user may choose to rerecord a note which already exists or record a new note. To record a new note, the user simply says "record" and dictates the message when the "recording" message appears on the screen. The recording process will automatically stop after 20 seconds. To stop recording earlier than that, the user pauses for a moment and then says "stop recording." That phrase will not be in the recorded message because the code which records for SPAM will adjust the byte count to a point before that phrase. (This is done by subtracting a certain number of bytes from the number of bytes actually recorded.) To rerecord a message, the user must simply specify which note to record, e.g., "Record that note."

The user may now check the recording by saying "play." If the note was rerecorded, then the recording will be localized at the correct position in space. If it is a new note, then it will be localized in the center of the room.

Once the note is heard, the user must decide whether or not to save it. The recording is buffered, i.e., no change has actually been made to the data. Hence, even if the new note is a rerecorded one, the user can say "cancel" to restore the environment to its original state (before the recording). To change the environment, the user must say "save it" in the case of the rerecorded note or "save it there" for the new note. The "save" and "cancel" commands essentially allow the user to accept or not accept the edit to the environment.

When saving a new note, a position for that note must be specified. Hence, "save it there" or "save it to my left" are both acceptable. Note that the "it" in these phrases represents the recorded note. The color of the new note is chosen by default. There are a maximum of 15 notes, and each note has a default color associated with it.

Therefore, if the user is saving the fourth note, that note will be the fourth color. However, sometimes the user wants to change the color of the note. This is possible with a simple command.

"Color..."

In order to change the color of a note, the user simply specifies which note is to be changed and the color it is to be changed to. If either of these components are left out of the "color" command, the system will ask the user to clarify what is wanted. There are 15 colors and any number of notes can be the same color. As in other commands, there are several ways to specify the command. Examples are:

"Color that note red"

"Color the blue note green"

"Color it yellow"

Notice that "it" can be used when the system know which note the user is working with. Usually, the "it" will refer to the last note used. If "it" is not understood by the system (i.e., it does not know which note is being referenced), then the system will ask the user to specify which note is desired.

"Move..."

In order to move notes around the environment, a "move" command has been included in the command set. The user specifies which note is to be moved and the system will ask where the user wants to move the note. The user must respond by indicating a position by gesture ("over there") or by description ("to my left"); once moved, the note will be localized in the new position and the graphic will be updated to reflect the new environment.

This option is included so that the user can reorganize the environment as time goes on. As messages become more important with the passage of time, the user may want to move them around within the environment to reflect this fact.

After moving a note, the user has the option of immediately restoring it to its old position. This is accomplished by saying "restore it" after a "move" command. If other commands are executed by the user (such as "play" or "record") after the "move" command, then the option to restore a note is lost.

"Delete..."

This command allows the user to modify the spatial environment by deleting notes which are no longer needed. The note will be erased from the graphic. The user simply has to specify (with description or gesture) which note is to be deleted. As with the "move" command, there is a "restore" command that will restore the note to the data base. The restore function must be used immediately after deleting a note or the option to restore the note will be lost.

Added to the commands mentioned above are several that don't alter the spatial environment but are essential to the user while using SPAM. In order to tell the speech recognizer to "listen" to the user, the phrase "pay attention" must be said. To stop it, the user simply says "stop listening." These two commands allow the user to talk without having the system interpreting what is said as commands. The system will acknowledge both of these commands to let the user know whether the system is listening or not. There is a "clear" command which allows the user to reset the system if a problem in the speech recognition process occurs (i.e., a mis-recognition). Finally, the user can redraw the graphic by simply saying "redisplay."

It is important to point out that the vocabulary of SPAM allows the user to use more than one phrase to indicate a command. "Record" and "take a note", "clear" and "reset", and "color" and "make" are examples. This is done to give the user a rich vocabulary to choose from, thus increasing flexibility while using the system.

GRAPHICS AND 3-D SOUND

The graphic mentioned in the previous section deserves further discussion as it is important in SPAM (see figure 4). The graphical image projected on the screen is a three-dimensional model of the media room, which is represented as a wire frame. There are two modes of operation. The image can be a virtual mirror, where the image is a mirror projection of the media room, or the image can be a simple projection.

In the mirror image, the notes that are in the front of the room are larger than those in the back of the room. This is what would be expected if the user was looking into a mirror. Only the front to back direction is reversed in the image; the notes at the right are still to the right in the image, those to the top are still at the top. Additionally, the cursor (which will be explained shortly) is large when the user points forward and becomes smaller as he points backward.

In the case of the simple projection, the image is a three-dimensional image of the room where the point of view is outside the room. Thus, the user sees an image of the room as if he were standing behind the rear wall and that wall was glass. Thus, notes to the rear appear larger than those to the front; the cursor is large when the user is pointing backward, small when pointing forward.

The user has the option of which method of projection is to be used in the system. The mirror image is more realistic, since the point of view is actually at the user rather than behind him. The use of a virtual mirror seems best since the entire front wall can be used as if the screen really was a mirror. However, the mirror image is difficult to work with since the user must work with a mirror of the spatial environment. The ability to deal with mirror images is easily accomplished with certain tasks since the user learns to use mirrors in everyday life (driving a car, tying a tie, etc.). The user is not proficient at using a mirror to organize data since this is not a common experience. Thus, the simple projection, although not as realistic as the mirror image, is easier to work with. Therefore, although the user does have the option of which method to use, the remainder of this discussion will assume that the user chose the simple projection method.

In the image, notes are represented by colored squares. All of the squares are originally the same size, but when projected in three dimensions, perspective makes the ones to the front of the room appear smaller than those to the rear. The notes must be the same size before perspective if the user is to obtain any depth perception from the image. The user can reason that a smaller note is behind a larger note. If the note size could vary, then the user would not know if a smaller note was at the same distance and just smaller in size or at a further distance. When notes overlap in the image, the image of the note to the rear will overlap that of the note to the front.

This image gives the user a fairly good impression of the spatial environment in which he is working. Added to the image is a transparent rectangle that represents a 3-D cursor. The rectangle will show the user where he is pointing in the room at all times. The unique aspect of this cursor is that when the user is pointing behind a note, that part of the cursor that is behind the note will appear to be behind the note. The transparency of the cursor allows the user to see through it when he is pointing in front of a note. The cursor diminishes in size just as do the notes in the image. Thus, the cursor is computed in three-dimensional object space before it is converted to the two-dimensional image space. This unique cursor allows the user to easily interact with the graphical environment, even though the image is two-dimensional and the environment in which he is interacting is three-dimensional.

Since the notes have spatiality unto themselves, why should there be a graphical interface at all? The graphical interface augments the user's ability to localize the sounds. It is faster to scan the image than to listen to all of the notes each time the user wishes to find a note. By having the graphic, the user can associate a note not only with a location but also with a real object. Manipulating the notes will be facilitated by the user manipulating the graphical image as well.

Additionally, the use of the 3-D cursor is instrumental for the user to use the Polhemus cubes. Even though the user knows where he is pointing in the room, it is necessary for the system to show the user that it also knows where he is pointing. This graphical feedback must be present to efficiently use gesture interaction within SPAM.

INTERACTING WITH 3-D SOUND

The use of graphics is only one method used to facilitate the interaction between the user and 3-D sound. There are several features of SPAM that help the user to interact with the sound. Additionally, the spatiality of the notes themselves add to the user's ability to manipulate those notes.

Added to the 3-D cursor explained previously is the feature to make this cursor "gravitate" to the notes. Hence, as the cursor moves near a note, it is attracted to that note. This allows the user to roughly indicate a note's position rather than having to point exactly at it. Thus, the user remembers a note is to the left and points in that general direction to specify that note.

While a note is playing, the image of that note will blink. Blinking was chosen to indicate the dynamic quality of a playing note. Thus, the user can easily determine the number of notes playing and which ones are not. This feature is especially helpful while using peruse mode.

When the system knows which note the user is pointing to or describing, it will change the image of that note to a hollow square rather than a solid one. This feature lets the user know that the system understands which note is being referenced without having to play that note. This is helpful when the user is giving a command and part of it is misunderstood by the system.

If the system cannot decide what to do after the user gives a command, it will ask the user to clarify that command. For example, if the user just says "play", the system will respond "which note?" with a stored voice. This feature allows the system to communicate with the user on a human level. Rather than printing queries, the system vocalizes them. Thus, there is vocal interaction, a conversation between user and system. The system also warns the user if he is attempting to do something illegal as well as telling the user that it is "ready" after certain commands.

Not only the appearance of sound helps the user interact with the environment, but also the localization of sound. The user will remember a note's position after hearing it in peruse mode. The user will remember that the message concerning work is to his left, a message concerning home is behind him, and a group of messages concerning an upcoming report is in front. This is the function of peruse mode; it lets the user familiarize himself with the environment by playing the localized sounds. With the addition of the graphical interface, the user will relate these messages to the correct images in the graphic.

Along the same lines, a user will group messages which relate to each other in the same area. For example, those messages concerning the work of the day could be placed in front, those concerning the next week's work could be placed to the rear. The user may also color code these notes by changing their color from the default color to something more meaningful to that user.

As a note becomes more important, the user will move it in the environment to reflect this fact. If the user records a message as a reminder to finish a paper in two weeks, he may put that note off to the side. On subsequent days, after perusing the notes in his environment, the user will probably move that note forward to reflect its increased importance. After the due date has past, the user will simply delete that note.

The user deals with the messages in an audio environment. Manipulation is accomplished by voice as well as gesture and output represented as audio information. SPAM allows the user to interact with spatial sound in a spatial environment using an interface designed for that purpose.

FUTURE WORK

Although the work done in this study deals with many of the issues in managing 3-D sound, there is some future work that should be done in both the context of spatial sound and the sound box itself.

The sound box has the ability to localize external sound. This is sound that is not digitized with the sound box but rather sound from other media, such as videodisc, video tape, audio tape, and even live sound. This feature of the sound box has never been utilized, but should be. It would allow applications that could localize sound in real time without having to record to magnetic disk. Applications involving videodisc would be especially fruitful due to the high density of sound data that could be stored on such a disk.

A system similar to SPAM which could handle incoming messages would prove to be interesting. This system would allow the user to organize data which was being received in real time. The user would not record messages, but would rather manipulate information coming from the outside world. Hence, this new system would be analogous to a spatial notebook, where the user could organize incoming sound data such as from a lecture or a conference.

Additional research should be devoted to localizing sound with respect to the user rather than to the room. The use of a Polhemus cube attached to the user would be ideal for this purpose. For example, if a note is localized at the user's left, it would still be at the user's left independent of which direction the user was positioned. However, the localization coordinate system should move with the user rather than staying constant within the room. This would further allow the user to move about the room and still have sound localized properly. This would alleviate some of the problems of the user having to sit in the media room chair to use localized sound.

An interesting addition to SPAM would be to have the recorded messages move automatically in the user's environment as those messages took on more importance. Thus, the user would not have to move notes around but the system would. The system could also alert the user to notes that need attention. This would be as if the system was reminding the user about a note previously recorded. The user could therefore inform the system to remind him to do something at a later date.

Although not directly related to 3-D sound, it would be useful to have the graphical interface actually be in 3-D rather than a flat image representing a 3-D world. This could be accomplished using PLZT glasses, which allow the system to give each eye of the user a different image. Thus, the visual disparity between the eyes can be controlled such that the user will see the image in 3-D. This feature would greatly help the user to manipulate items in a spatial environment since the graphical interface would present that spatiality.

CONCLUSIONS

This project had two goals. One was to redesign and overhaul the sound system, and the other was to explore 3-D sound and its contributions to the human-computer interface.

The sound system as explained briefly in this paper and more completely in The Sound System User's Manual (5) is indeed better than the previous version. The system is more interactive, more flexible, and more powerful than the previous system. It allows sounds to be stored on magnetic disk in an efficient manner as well as allowing great flexibility in playing these sounds from disk.

Using the Spatial Audio Notemaker as an application which actually uses 3-D sound, the validity of the contributions of 3-D sound to the human-computer interface was shown. If 3-D sound was not a part of the Spatial Audio Notemaker, the user could not manipulate notes with the same ease and proficiency as with spatial sound. Thus, the spatiality of sound is necessary in order for sound to contribute to the interface.

This contribution can be summarized as providing the user with an ability to manipulate audio data in a manner that reflects the user's true spatial environment and allows flexibility in that manipulation. This ability is directly related to the spatiality present in 3-D sound. Without this spatiality, the manipulation would be contrived and difficult.

The work set forth in the previous section shows further applications of 3-D sound. This study has opened a new door in the world of human-computer interfaces, that of spatial sound. Hopefully, this new dimension of that interface can be explored further, leading to an interface between man and computer which allows even more transmission of data, as well as communication on a human level, than previously known. This will allow man to use computers more efficiently, more easily, and with less stress than ever before.

REFERENCES

1. Bolt, Richard A. "Put-That-There": Voice and Gesture at the Graphics Interface. SIGGRAPH '80 Conference Proceedings, 14 (3), July 14-18, 1980, Seattle, Washington, 262-270.

2. Grogen, David P. Computer Generation of Sounds with Localization in Three Dimensions. Master of Science Thesis, Massachusetts Institute of Technology, June, 1977.

3. Hurd, Jonathan A. An Interactive Digital Sound System for Multi-Media Databases. Bachelor of Science Thesis, Massachusetts Institute of Technology, June, 1979.

4. Moosher, David P. Digital Sound for the Spatial Data Management System. Bachelor of Science Thesis, Massachusetts Institute of Technology, June, 1980.

5. Vershel, Mark A. The Sound System User's Manual. M.I.T. Architecture Machine Group, Cambridge, Massachusetts, April, 1981.