Architecture of a Teleconference System based on Minimum Audible Angle

N Harikrishnan Potty, Rajbabu Velmurugan and Preeti Rao
Indian Institute of Technology Bombay, Mumbai 400076, India
< {hkpottyn, rajbabu, prao} @ ee.iitb.ac.in >

***Abstract*— Several researchers have established the superior performance of spatial audio over monaural audio in multi-talker environments. The advantage provided by spatial audio can be incorporated in an audio teleconferencing system for an enhanced user experience. In our earlier work, a novel scheme for the spatial rendition of audio in an audio teleconferencing situation was proposed based on the concept of Minimum Audible Angle (MAA). We provide the basic architecture for a multi-talker teleconferencing system that uses the proposed scheme with independent clients that facilitate speaking, listening or both. The software stack that runs on each client is capable of receiving, sending and processing of audio signals and operates in real-time. The implementation is done in Java, using the Java Media Framework (JMF) for capturing and playback of the audio signal. RTP (Real-time Transport Protocol) protocols are used to ensure real-time transmission and reception over the network.**

Keywords-teleconferencing system; MAA; multi-talker environment ; JMF; RTP

I. INTRODUCTION

Today's teleconferencing technology gives users a broad range of options when it comes to communicating and collaborating with people at distant locations. One of the major challenges in designing a teleconferencing system is to make sure that it performs efficiently in multi-talker environments so that the participants' experience is better. The multi-talker environment can be compared to a cocktail party situation where sounds from different directions are received simultaneously. The binaural advantage helps humans to focus attention in one direction even in such a scenario. This special capability of human beings has been investigated by several researchers for long and motivates new ideas for improving existing audio teleconferencing systems.

It has been concluded that spatial rendition of audio provides remarkable improvement in intelligibility over monaural rendition in a multi-talker environment. The spatial separation between the speakers provides a release from masking which helps the listener to focus with more ease on a specific target [1][2]. It has been suggested in [2][3] that listeners are able to localize speech signals when two competing speech signals were spatially separated in azimuth.

Many a research had been made in the past regarding the effect of spatial audio in a typical teleconference system. Studies conducted in [4] claim that providing a stereo signal instead of a mono signal improves the teleconferencing experience for the listener. According to the results of [5], spatial rendering helps the listener to easily identify a change of the active speaker via the direction of the incoming audio.

Once the advantage of spatial separation in a teleconferencing system has been identified, the next challenge is to find an optimal mechanism to spatially locate the multiple talkers from the perspective of each listener. Several researchers conducted experiments to improve speech display in the multi-talker environment [6]. The design of an audio conferencing tool was proposed in [7]. It was implemented based on architecture similar to client-server architecture, which required a server. In our implementation, we dispense with the server and make a scalable system by connecting identical, independent client systems. Another implementation of the teleconferencing system is described in [8]. It uses Java Media Framework (JMF) for transfer of audio. But none of the above implementations talk about a criterion by which the speakers can be placed. Our implementation of the teleconference system takes into consideration the efficient placement of speakers in addition to the transfer of audio over the network. We proposed a speech display with a spatial configuration that exploits the dependence of minimum audible angle (MAA) on azimuth [9]. In this paper we focus our attention on developing the architecture of a real-time and scalable teleconference system that places the speakers in accordance with the speech display proposed in [9].

In Section II the algorithm that is used to spatialize the talkers is described. The third section describes about the system requirements, both hardware and software. In Section IV the overall architecture of the system is described.

II. ALGORITHM FOR PLACING THE TALKERS

In the teleconferencing setup we have designed, each participant system can manually select the spatial location of other participants. If no such selection is made, then by default an algorithm will be run that will automatically select the spatial location for other participants' audio streams. This algorithm is based on the result that we have derived in [9] which placed the speakers according to the MAA (Minimum Audible Angle). MAA is the minimum angular separation that provides release from masking in a speech over speech

978-1-4673-0816-8/12/$31.00 ©2012 IEEE

scenario. The MAA thus derived at various azimuths is given in Table I. These MAAs have been found based on subjective experiments where two talkers are simultaneously speaking. So this teleconference system is designed assuming that a maximum of only two speakers are active simultaneously at any instance. However, the overall number of participants can be much higher than this.

The basic flow of the algorithm is shown in Fig.1. The first client will be placed at 0^0 and the position of the next client will be decided based on the MAA at that azimuth. The MAAs for azimuths other than given in the table can be identified from Fig.2, which is a result from [9]. From Fig. 2, it is clear that the MAA varies almost linearly as we move away from the frontal plane. Also, it is clear from Fig.2 that at a fixed azimuth, the achieved separability decreases as the angular separation decreases. As the number of clients that needs to be accommodated in the teleconferencing setup increases, the angular separation decreases and the achieved separability decreases. An example of speaker placement in the case where six participants are present is shown in Fig.3. This speaker placement is as given in [9].

TABLE I. MAA FOR VARIOUS AZIMUTH CENTRES [8]

Azimuth	MAA
0^0	5^0
-30^0	14^0
-45^0	18^0
-60^0	22^0

III. SYSTEM REQUIREMENTS

A. *Hardware Requirements*

The hardware requirement for the implementation of the system is very minimal. We consider a teleconferencing setup where participants use a network of PCs as the communication platform. The PCs should have at least 128Mb RAM for running Java. The audio capture device is a microphone and the audio playback is happening through headphones. We assume that all the connected systems use the same sampling rate.

B. *Software Requirements*

The operating system we have used is Microsoft Windows 7 Professional. The software stack has been implemented entirely in Java. The latest available version of Java (jdk1.7.0) was used [10]. We have chosen Java for our implementation because of the various advantages it possesses over its counterparts, especially its platform independence. Hence the application we have developed can be run on several platforms without any major modifications.

Another feature that Java supports is multithreading. Multithreading is the capability for a program to perform several tasks simultaneously using multiple threads. In Java, multithreaded programming is smoothly integrated into the language, while in other languages, the operating system-specific procedures have to be called in order to enable multithreading. Several tasks in our application are implemented using multithreads. More details of this will be given in the next sections.

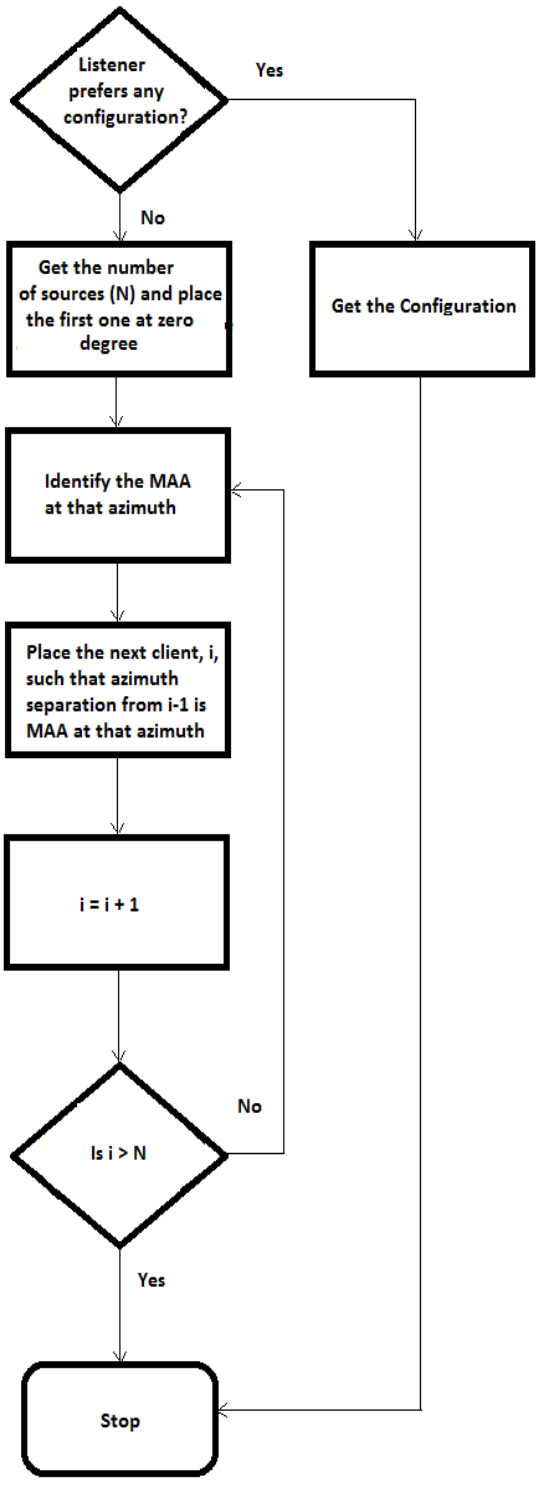

Figure 1. Flow chart representation of the speaker placement algorithm

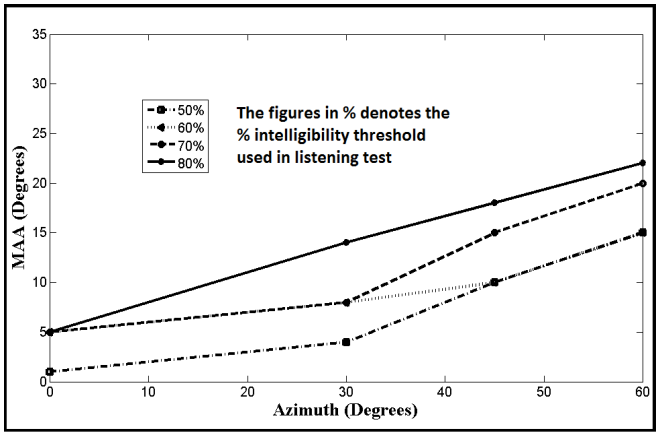

Figure 2. Variation of MAA with azimuth

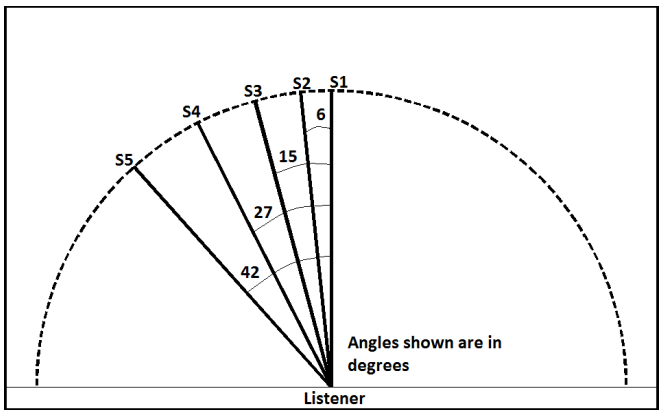

Figure 3. Position of speakers in six participant case [1]

Another reason for which Java was used in our system is the availability of the package named Java Media Framework (JMF). It helps Java programs to handle audio, video and other time-based media. JMF is very simple to use and is well integrated with Real-time Transmission Protocols (RTP) which ensure real-time data transfer across the network. We have used the latest available JMF (JMF2.1.1e) in our software stack [11]. The software stack in each of the system is almost identical and hence it easily scalable to accommodate more number of participants.

IV. ARCHITECTURE

A. *Overall Architecture*

The overall architecture of the system is simple and scalable. It is simple in the sense that it is easily implementable, and scalable in the sense that the number of users can be increased or decreased without significant effort.

The system consists of several clients that can function independent of others. The number of clients will be equal to the number of participants. Each client is capable of audio transmission and reception over the network. Apart from this, they are also capable of capturing and playing back the audio. The overall architecture, for the three participant case, is shown in Fig.4. Each client sends and receives a single (mono) stream of audio to/from every other client in the teleconferencing system. At each client's system, the various received audio streams are individually spatialized based on the MAA algorithm (or, based on the listener's own preference by a setting in the available GUI). Thus each received mono audio stream is converted to stereo form based on the selected angle before being presented to the listener.

The architecture of individual client is similar and is shown in Fig.5. In this figure, the bottom layer (4th layer) denotes the hardware layer which consists of the devices for capturing and playing back the audio. The third layer is the native implementation of various Java packages that are platform dependent. This is available as free to download for most of the widely used platforms. The second layer contains the classes and the functions exposed by Java for the programmers. Some relevant packages that we have used are shown here. The JMF and the RTP forms a part of the javax.media package. The graphical and event handling sections are implemented with the help of javax.swing and java.awt packages. The application written by us, to accomplish the azimuth dependent mono to stereo audio conversion, sits on top of these various packages.

B. *Software Architecture*

The software stack in each client can be diagrammatically represented as shown in Fig. 6. The audio transmission part is explained in Fig. 6(a). The audio data is captured by the hardware which can be either a webcam or a microphone. The JMF acts as the interface for the program to control the capture of data. RTP enables the transmission of real-time media streams across the network.

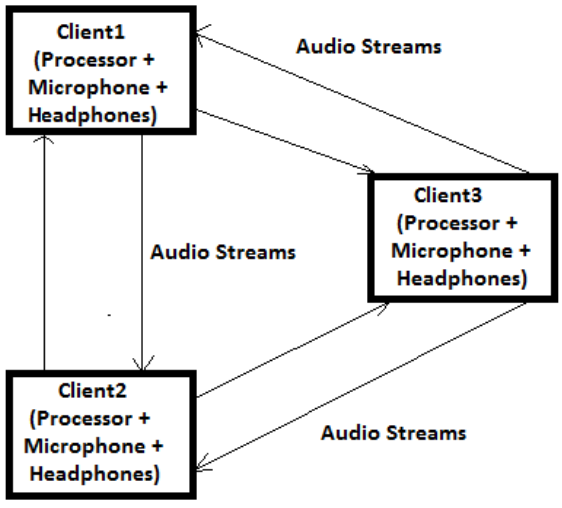

Figure 4. Overall architecture for the 3 participant case. Mono audio streams are transmitted between clients.

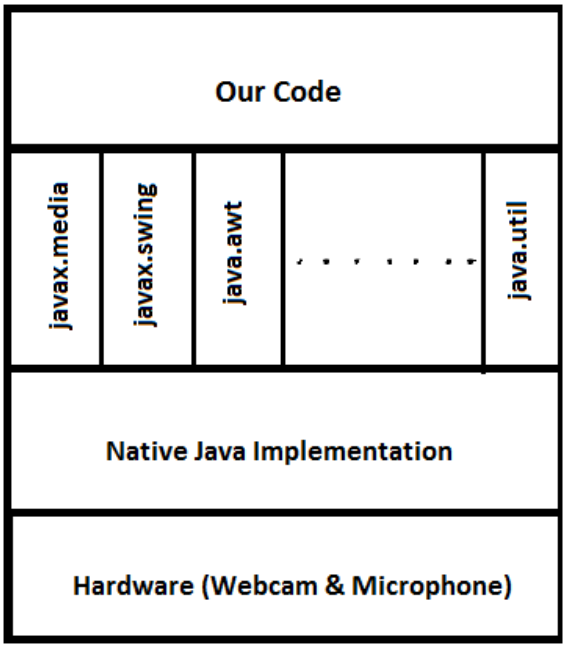

Figure 5. The layered architecture of each client

The transmitted audio is received at the receiving end in real-time which is ensured by the RTP. A single client will be receiving data from all other clients in the network. To handle data from different clients, a thread is created to process the data from each client. This will ensure that the reception as well as the processing happens in parallel.

The received data has to be processed such that a sense of direction is associated with the streams coming from each of the other clients. For this, the virtual azimuths of clients are obtained from the listener's input or from the algorithm described in Section II. Once the azimuths are obtained, Head Related Impulse Responses (HRIR) corresponding to each azimuth is obtained from the database. The database contains both standard as well as interpolated HRIRs. Standard HRIRs are those which are provided by MIT Media Lab [12]. The HRIRs for azimuths which are 5^0 apart starting from 0^0 to 360^0 are available in this database. The interpolation technique used to obtain the HRIR for non-standard azimuths is given in [13]. The incoming mono stream from each speaker is convolved with the corresponding HRIR and a stereo stream is obtained. The convolution is performed by adding it as a plug-in to the JMF. With the help of JMF, the processed data is given to the hardware for playback.

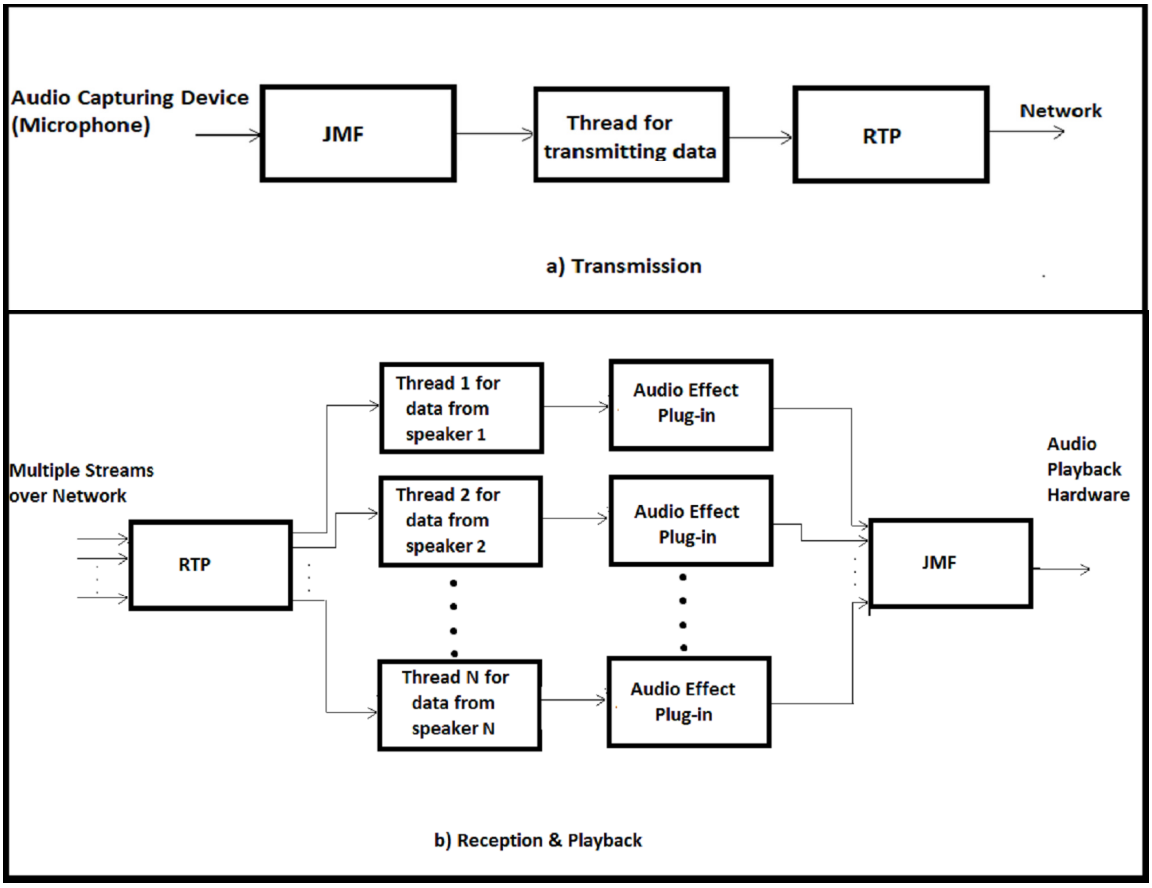

Figure 6.a) Block diagram for for the transmittion part
b) Block diagram for the reception part

The HRIRs can be considered as a sequence of numbers of type double. The number of samples in any HRIR depends on the sampling rate used. If sampling rate is 44.1 KHz, then the number of samples is 512. At any instant the number of samples from each stream that is convolved is 6912 bytes. With systems having reasonably good computatinal capability, this process of convolution could be performed in real-time without much delay. The buffering is entirely taken care by JMF and does not need any special handling.

V. SUMMARY AND FUTURE WORK

We have implemented the teleconferencing system with two participants. Each of the participants can enter the virtual location of the other participant. Both of them have a PC, Microphone and a Headphone. Audio was transmitted over an Ethernet network. The sampling rate used was 44.1 KHz. The performance will be better if we use lesser sampling rates. But 44.1 KHz was used to test for worst case scenario. The system performed quite well with this high sampling rate. The listeners reported that they could get the sense of direction associated with the incoming signals.

Currently we are placing the speakers on the basis of MAA. We are making use of static spatialization algorithm where the speakers will have fixed positions. This setup will not be able to accommodate large number of participants as the angular separation between them decreases as the number of participants increase. Also static spatialization will not be using the available soundscape efficiently, especially when only a few listeners are active and when they are closely positioned. Such scenarios can be dealt with by making use of dynamic spatialization techniques. In the case of dynamic spatialization, the positions of the speakers will not be static and hence the soundscape will be more efficiently utilized. Also, as a part of future work, we are planning to implement and test the above system with upto six participants.

ACKNOWLEDGMENT

The research work is supported by the project “National Program on Perception Engineering”, sponsored by the Department of Information Technology, MCIT, Government of India.

REFERENCES

[1] D. S. Brungart, B. Simpson, M. Ericson, and K. Scott, “Informational and energetic masking effects in the perception of multiple simultaneous talkers”, J. Acoust. Soc. Am., 110 (5), Pt.1, (Nov. 2001), 2527-2538.

[2] R. Freyman, K. Helfer, D. McCall, and R. Clifton, ”The role of perceived spatial separation in the unmasking of speech”, J. Acoust. Soc. Am., 106 (6), (Dec. 1999), 3578–3587.

[3] H. S. Colburn, B. G. Shinn-Cunningham, G. Kidd, Jr., and N. Durlach, “The perceptual consequences of binaural hearing”, International Journal of Audiology 2006; 45 (Supplement1): S34 -S44D.

[4] Nicole Yankelovich, Jonathan Kaplan, Joe Provino, Mike Wessler, Joan Morris DiMicco, “Improving Audio Conferencing: Why Two Ears are Better than One”, CSCW ‘06, November 4-8, 2006, Banff, Alberta, Canada.

[5] Mansoor Hyder, Michael Haun,Christian Hoene, “Placing the Participants of a Spatial Audio Conference Call”, University of Tübingen, 72076 Tübingen, Germany

[6] D. S. Brungart, and B. Simpson, “Optimizing the spatial configuration of a seven-talker speech display”, in Proceedings of the International Conference on Auditory Display (Boston, MA, USA, July6-9, 2003).

[7] Zhenkai Zhu, Sen Wang, Xu Yang, Van Jacobson, Lixia Zhang, “ACT: Audio Conference Tool Over Named Data Networking”, ICN’11 August 19, 2011, Toronto, Ontario, Canada.

[8] H. Abdel-Wahab, O. Kim, P. Kabore, and J. P. Favreau., “Java-based multimedia collaboration and application sharing environment.”, In Colloque Francophone sur L’Ingenierie des Protocoles (CFIP’99), Nancy, France, April 26–29 1999.

[9] N. Harikrishnan Potty , Dipti Sengupta, Rajbabu Velmurugan and Preeti Rao, “Azimuth-dependent Spatialization for a Teleconference Audio Display,” National Conference on Communications, January 2011.

[10] http://www.oracle.com/technetwork/java/javase/downloads/index.html (Last accessed on 08th October 2011).

[11] http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html (Last accessed on 08th October 2011).

[12] http://sound.media.mit.edu/resources/KEMAR.html (Last accessed on 08th October 2011).

[13] L. Chen, H. Hu, Z. Wu , “Head-related impulse response interpolation in virtual sound system”, Fourth International Conference on Natural Computation, 2008.