Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Research on fast real-time adaptive audio mixing in multimedia conference

FAN Xing (樊 星)[†], GU Wei-kang (顾伟康), YE Xiu-qing (叶秀清)

(*Department of Information and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*)

[†]E-mail: starfan@mail.hz.zj.cn

Received Apr. 16, 2004; revision accepted Jan. 25, 2005

Abstract: In multimedia conference, the capability of audio processing is basic and requires more for real-time criteria. In this article, we categorize and analyze the schemes, and provide several multipoint speech audio mixing schemes using weighted algorithm, which meet the demand of practical needs for real-time multipoint speech mixing, for which the ASW and AEW schemes are especially recommended. Applying the adaptive algorithms, the high-performance schemes we provide do not use the saturation operation widely used in multimedia processing. Therefore, no additional noise will be added to the output. The above adaptive algorithms have relatively low computational complexity and good hearing perceptibility. The schemes are designed for parallel processing, and can be easily implemented with hardware, such as DSPs, and widely applied in multimedia conference systems.

Key words: Multimedia conference, MCU, Real-time, Adaptive, Audio mixing, Aligned-to-self, Aligned-to-energy
doi:10.1631/jzus.2005.A0507 **Document code:** A **CLC number:** TP309

INTRODUCTION

As one of the main applications in Packet-based network (PBN) environment, multimedia communication is developing rapidly, with many service providers starting to offer new services. Audio interactive operation is one of the most basic components in multimedia conference. Because of absence of QoS mechanism in most application environments of PBN, network blocking will cause problems such as data loss and audio jitter in end-to-end communication (Yang *et al*., 2001). The burden of network transport and the instability of data transmitting and receiving significantly increase when several endpoints mutually and simultaneously send data. The needs for real-time characteristics of audio interactivity are much greater than those of any other factors. Longer delay or jitter in video or data interactivities is easier to be tolerated by users in practical use, while short delay or jitter in audio stream will cause apparent break and noise, which leads to failures in understanding the concepts conveyed by the audio. In order to solve the problem, audio mixer is introduced to mix up all the input audio steam, and decreases the burden of network transport and the endpoint calculation consumption. Rangan *et al*.(1993) provided architectures and algorithms for media mixing, which mostly concentrated on the transmission. And Daigle and Langford (1986) analyzed the queuing model in audio communication. Generally, the mixed stream output will frequently go beyond the limitation of quantization, so we have to use saturation operation to handle it by adding noise. This article provides several schemes for solving the problem, which do not add any other noise, can maintain the acoustic characteristics of human speech better, have low computational complexity, and can be applied in real-time multimedia communication system.

All the implementation work was done for the centered multipoint conference defined in H.323v4 (ITU-T, 2000). All the experimental results were collected in the multimedia multipoint conference system prototype we implemented.

MODEL OF AUDIO MULTIPOINT PROCESSOR

According to H.323v4 Standard Recommendation from ITU-T (2000), a Multipoint Control Unit (MCU) can be divided into two core modules: Multipoint Controller and Multipoint Processor (MP). MP can be categorized into Video MP, Audio MP (AMP) and Data MP. We concentrate on AMP, which consists of Audio Encoder, Audio Decoder and Audio Mixer (AM). Fig.1 describes the general model of AMP. In an AMP, K independent conferences are supported. Each conference has a corresponding Audio Processing Unit (APU), the ith APU has M_i input and N_i output, as shown in Fig.2.

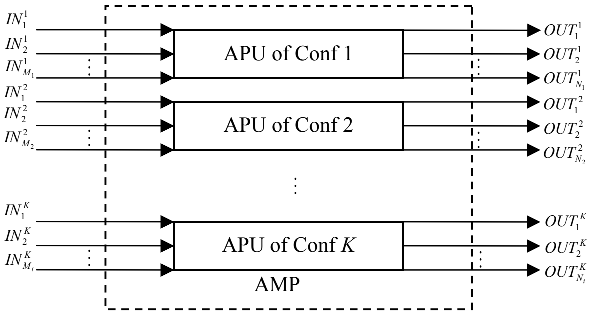

Fig.1 The structure of AMP

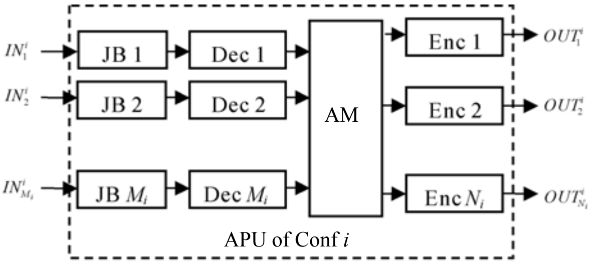

Fig.2 The structure of APU

SPEECH CODEC RECOMMENDATION IN MULTIMEDIA CONFERENCE

In a centralized conference, each terminal establishes unicast connection to the MCU, sends data to and receives data from the MCU. H.323 recommends several speech codec schemes. They are G.711, G.722, G.723.1, G.728 and G.729. We know that all their input and output are mono and 16 bits-sampling. For G.722 the sampling rate is 16 kHz, and those of all the others are 8 kHz.

The input of AM is the raw audio data after decoding, and the output is the raw audio data after audio mixing.

We assume that there are M_i endpoints in the ith conference. At time t, the jth (j=1, 2, …, M_i) speech decoded output is indicated by $a_j^i(t)$, whose value domain is $[-2^{Q-1}, 2^{Q-1}-1]$, where Q is the number of sampling bits. We want N_i+1 output after audio mixing, in which the kth (k=1, 2, …, N_i) output is indicated by $b_k(t)$. In practical application, we take N_i=M_i. Then $b_{N_i+1}(t)$ indicates the output of mixing of all the N_i input audio streams and all the others are the output of mixing of all the input audio streams excluding the corresponding input. Apparently, after mixing, we have N_i of N_i–1 mixing output and only one of N_i mixing output. All $b_k(t)$ must have the value domain $[-2^{Q-1}, 2^{Q-1}-1]$, so we have to use saturation operation to handle the case when $b_k(t)$ is out of bounds.

AUXILIARY PROCESS FOR AUDIO MIXING

We use the Real-time Transport Protocol (RTP) (Schulzrinne *et al.*, 1996) to encapsulate the data when the encoded audio streams from participants are sent to the MCU continuously. Because there are data loss and later-sent-earlier-arrive problems, and the ununiformity of the audio stream source, there must be data loss, jitter and wrong order of data clip received in the audio data on the MCU. We use the technique of Jitter Buffer (JB) to solve the problem. In Fig.2, the JB Modules are implemented by using this technique. The buffered encoded audio stream is decoded by the speech codec and passed to the mixer. The mixed audio stream must be encoded by the codec and then sent to the terminals.

In the models provided by this article, different kinds of encoders and decoders can be used at the same time in an application system. So we can see that the APU can do both audio mixing and transcoding.

Because there are two sampling rates in the speech input, we should do resampling before the mixing process.

ADAPTIVE REAL-TIME AUDIO MIXING SCHEMES

Mixer is the core module in AMP. In practical occasion, if there are only two participants in a conference, we just make it do Full-Duplex Transmission. If more than 3 people take part in the conference, switched transmission or mixing should be introduced.

We have two strategies for switched transmission. One is to transmit all the audio streams from other endpoints to the destination; the other is to set up a kind of regulation to determine which stream should be transmitted. Both of the strategies have obvious defects. The first one adds much burden to network transmission. The problem can be solved by using multicast transport, which is not supported in many cases. The second one cannot support many participants in one conference, because if there are many participants speaking simultaneously in a discussion, the system performance will be slow and the audio streams will be switched so frequently that the listeners cannot distinguish the speeches.

Real-time audio mixing is a good way out. We know the audio mixing process is simply the linear combination of each source pressure wave (González and Abdel-Wahab, 1998). However, the digital audio signals have the quantization limitation, so we can see there may be overflow after doing the sum. Generally, we check the overflow, and do saturation operations. That is, if the result is greater than the upper boundary, it will be set to the upper boundary; if it is less than the lower boundary, it will be set to the lower boundary. However, the operation affects the characteristic of the audio stream in time domain, so new noise is added. This is the reason why the noise occurs. What is more, while the number of participants increases, the noise increases. Experiments proved that when more than 4 participants are in one conference, the mixing result is unacceptable.

To solve the problem, five real-time adaptive audio mixing schemes are introduced. They are Align-to-Average Weighted (AAW) scheme, Align-to-Greatest Weighted (AGW) scheme, Align-to-Weakest Weighted (AWW) scheme, Align-to-Self Weighted (ASW) scheme and Align-to-Energy Weighted (AEW) scheme. We use weighting methodology to do the process. Generally, we define $w_j^i(t)$ as the corresponding weight of $a_j^i(t)$. Then we have the general calculation formula,

$$\begin{aligned} b_k^i(t) &= \sum_{j=1,j\neq k}^{M_i} w_j^i(t)a_j^i(t) \Big/ \sum_{l=1,l\neq k}^{M_i} w_l^i(t) \\ b_{M_i+1}^i(t) &= \sum_{j=1}^{M_i} w_j^i(t)a_j^i(t) \Big/ \sum_{l=1}^{M_i} w_l^i(t) \end{aligned} \qquad (1)$$

For AAW, we can see from the name that we just define the weight as the same value as $w_j^i(t) = 1/M_i$, then we have,

$$b_k^i(t) = \frac{1}{M_i - 1}\sum_{j=1,j\neq k}^{M_i} a_j^i(t),\ b_{M_i+1}^i(t) = \frac{1}{M_i}\sum_{j=1}^{M_i} a_j^i(t) \qquad (2)$$

Though noise will not be added when this scheme is used, yet all the streams taking part in the mixing are attenuated (Tu *et al.*, 2002), so the output is consequently attenuated. If one stream taking part in the mixing has very low level, the result will be pulled down. If more than 4 streams take part in the mixing, the level of the output is weak. This is the limitation of AAW scheme.

In AGW scheme, the choice for $w_j^i(t)$ is based on the amplitude of the corresponding audio stream. The scheme provided in Yang *et al.*(2001) belongs to this category. We have done some modifications to the scheme to develop a new one. We define $\overline{TotalMax}^i$ as the maximum value of all buffers, and $\overline{MixedMax}^i$ as the greatest value of $b_k^i(t)$ resulted from Eq.(2). Then we have,

$$b_k'^i(t) = b_k^i(t)\mu^i\, \overline{TotalMax}^i \Big/ \overline{MixedMax}^i \qquad (3)$$

where μ^i is a factor used to adjust the amplitude value of $b_k'^i(t)$ and is chosen from a neighbor domain of the absolute value of $\overline{TotalMax}^i \Big/ \overline{MixedMax}^i$, and is in the range of a buffer. This scheme adaptively adjusts the mixing results. However, there are still cases of overflow. So after the process using Eq.(3), we should still do overflow-check and saturation operation. Though the results from AAW are exaggerated by a factor selected according to partial characteristics, new noise is still added from the point of view of the whole process. In case that one stream is very weak, the output waveform will be greatly distorted, as shown in experimental results.

The way AWW scheme behaves is opposite to that of AGW. The exaggerating factor is selected according to the weakest one. The advantage is to amplify the weaker streams in the mixing process, but there is still noise added. The processing procedure and computation time consumption of AWW is similar to that of AGW.

In ASW scheme, we consider the characteristics of each input audio stream separately. We choose the weight from their own amplitude, then we have,

$$w_j^i(t)=\left|a_j^i(t)\right| \Big/ \sum_{p=1}^{M_i}\left|a_p^i(t)\right| \tag{4}$$

We substitute Eq.(4) for the weight in Eq.(1), then we have

$$\begin{aligned} b_k^i(t)&=\sum_{j=1,j\neq k}^{M_i}\left(\left[a_j^i(t)\right]^2\operatorname{sgn}\left[a_j^i(t)\right]\right)\Big/\sum_{l=1,l\neq k}^{M_i}\left|a_l^i(t)\right| \\ b_{M_i+1}^i(t)&=\sum_{j=1}^{M_i}\left(\left[a_j^i(t)\right]^2\operatorname{sgn}\left[a_j^i(t)\right]\right)\Big/\sum_{l=1}^{M_i}\left|a_l^i(t)\right| \end{aligned} \tag{5}$$

where sgn(·) is the sign function. We just make sure that the sum of the energy of the input streams cannot be zero. If the sum of the energy is zero, it indicates that all the input streams have the level of zero, so it is not necessary to do mixing.

We have designed a processing model for the audio mixing algorithm shown in Fig.3. We can see that the scheme is well optimized. From $a_j^i(t)$ we first calculate $\overline{SG}_j^i(t)=\operatorname{sgn}\left(a_j^i(t)\right)$. Then, we have $\overline{AB}_j^i(t)=a_j^i(t)\overline{SG}_j^i(t)$. Afterward, we calculate $\overline{SQ}_j^i(t)=a_j^i(t)\overline{AB}_j^i(t)$. Using the results we got from the former computation, we have $\overline{SAB}^i(t)$ and $\overline{SSQ}^i(t)$ using Eq.(6). The result of dividing them is $b_{M_i+1}^i(t)$. $\overline{SAB}^i(t)$ and $\overline{SSQ}^i(t)$ respectively remove the corresponding value of the same audio stream. After the division of the two values, we have all the other mixing results whose echo is removed.

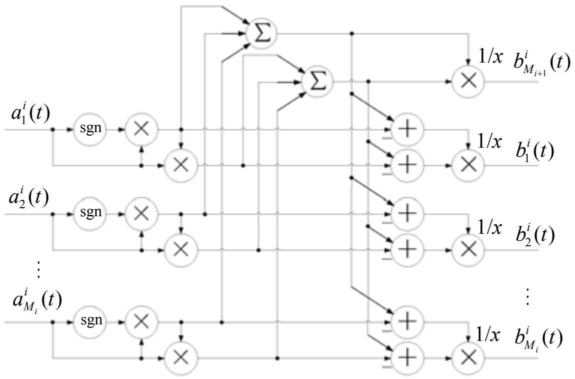

Fig.3 Optimized ASW model

$$\overline{SAB}^i(t)=\sum_{j=1}^{M_i}\overline{AB}_j^i(t),\ \overline{SSQ}^i(t)=\sum_{j=1}^{M_i}\overline{SQ}_j^i(t) \tag{6}$$

The scheme is designed for parallel computation and can be easily implemented by DSPs, and make full use of the parallel computing capability. If in IA32 platform, we can use SIMD instruction set to do the optimization, the system will have much better real-time performance.

If we choose the weight from the energy value of the input streams, then we have AEW scheme.

$$w_j^i(t)=\left[a_j^i(t)\right]^2\Big/\sum_{p=1}^{M_i}\left[a_p^i(t)\right]^2 \tag{7}$$

$$\begin{aligned} b_k^i(t)&=\sum_{j=1,j\neq k}^{M_i}\left[a_j^i(t)\right]^3\Big/\sum_{l=1,l\neq k}^{M_i}\left[a_l^i(t)\right]^2 \\ b_{M_i+1}^i(t)&=\sum_{j=1}^{M_i}\left[a_j^i(t)\right]^3\Big/\sum_{l=1}^{M_i}\left[a_l^i(t)\right]^2 \end{aligned} \tag{8}$$

This scheme has more computational complexity, and actual mixing results similar to those of the ASW scheme. Since there are cubic computations, the time consumption and storage consumption are greater than those of ASW scheme. So we tend to use ASW scheme in practical application.

EXPERIMENTAL RESULTS

In last section, we list the experimental results of the five schemes. Since AWW is similar to AGW, we do not make it an individual item in the experiments.

First, we analyze the output waveform and the influence on hearing perceptibility. Figs.4a and 4b are the input streams. Figs.4c~4f respectively show the output of AGW, ASW, AEW and AAW. It is obvious that Fig.4e is similar to Fig.4d, and Figs.4d and 4e are much better than Fig.4c. Fig.4f has the weakest waveform, since there is attenuating effect in the computation for average values. The outputs shown in Figs.4d and 4e are closer to the inputs. The output shown in Fig.4c is obviously distorted, because it depends on the partial statistical characteristics. Though Fig.4f is similar to the inputs in waveform, the amplitudes of the stream are attenuated.

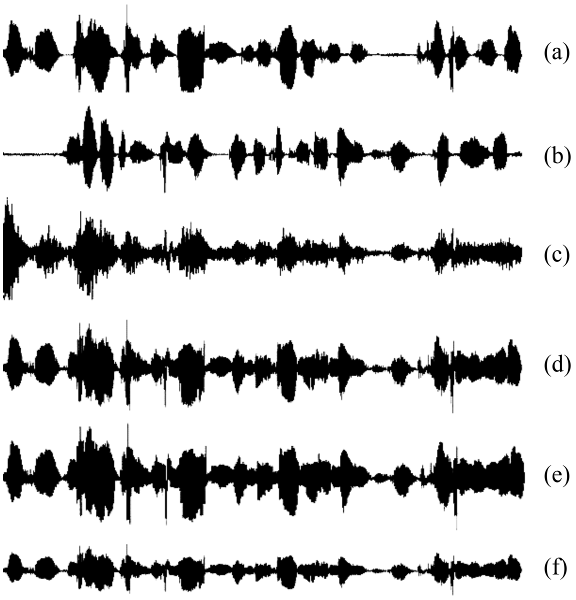

Fig.4 Waveform of experimental input and output. (a) 1st input; (b) 2nd input; (c) Output of AGW; (d) Output of ASW; (e) Output of AEW; (f) Output of AAW

In the experiment on subjective hearing perceptibility, the output of ASW scheme is a little bit better than that of AEW because the average energy from the output of ASW is a little bit lower than that of AEW. But obviously, the computational time consumption of AEW scheme is greater than that of ASW. The output of AGW scheme is the worst, but is still acceptable. The output of AAW is medium, but it has the lowest energy level in the whole experiment. Therefore, its resolution for hearing perceptibility is degraded.

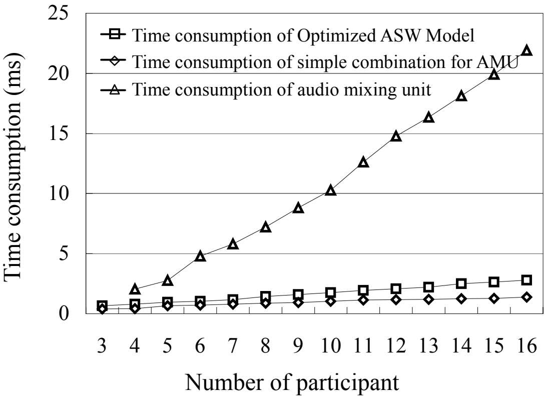

Fig.5 Time consumption comparison

According to the results and analysis above, we concentrate on ASW scheme. We do the performance test for ASW scheme and use Intel® VTune™ Performance Analyzer 7.0 to do the profiling. We ran our test program on an Intel Pentium IV 1.8 G PC with 512 MB memory. We could see the time consumption of AMU Simple Combined Model which simply uses an AMU of single output, and that of Optimized ASW Model. The *y*-axis unit is microsecond, and the *x*-axis indicates the participant's number. Fig.5 shows the time consumption increases quickly with the increase of the number of participant, and that the time consumption performance of Optimized ASW Model is much better. The time consumption of optimized ASW Model is very near to that of a single AMU. The time consumption has linear relationship with the number of participants. And the Optimized ASW Model has good performance when the number of participants ranges from 3 to 16, so it can be widely used in multipoint speech audio mixing application.

CONCLUSION

The Optimized ASW Model can meet the practical demands not only in general application, but also has good performance on occasions when more participants take part in an audio mixed conference. Because Optimized ASW Model keeps the characteristics of the input audio streams, it provides output with good quality. And it has low computational time consumption and good hearing perceptibility, so it can be used in many real-time multimedia communication applications.

References

Daigle, J.N., Langford, I.D., 1986. Model for analysis of packet voice communications systems. *IEEE Journal on Selected Areas in Communications*, **4**(6):847-855.

González, A.J., Abdel-Wahab, H., 1998. Audio Mixing for Interactive Multimedia Communications. JCIS'98, Research Triangle, NC, p.217-220.

ITU-T, 2000. Packet-Based Multimedia Communication System. ITU-T Recommendation H.323 v4.

Rangan, P.V., Vin, H.M., Ramanathan, S., 1993. Communication architectures and algorithms for media mixing in multimedia conferences. *IEEE/ACM Transactions on Networking*, **1**(1):20-30.

Schulzrinne, H., Caner, S., Frederick, R., Jacobson, V., 1996. RTP: A Transport Protocol for Real-time Applications. IETF RFC 1889.

Tu, W., Hu, R.M., Ai, H.J., Xie, X., 2002. Audio MP in video conference. *Geomantics and Information Science of Wuhan University*, **27**(1):98-101 (in Chinese).

Yang, S.T., Yu, S.S., Zhou, J.L., 2001. A multipoint real-time speech mixing and scheduling algorithm based on packet networks. *Journal of Software*, **12**(9):1413-1419 (in Chinese).