Power-aware Bandwidth and Stereo-image Scalable Audio Decoding

Wendong Huang, Ye Wang and Samarjit Chakraborty
School of Computing
National University of Singapore
{huangwd, wangye, samarjit}@comp.nus.edu.sg

ABSTRACT
We propose a new workload-scalable audio decoding scheme that would enable users to control the tradeoff between playback quality and power consumption in battery-powered portable audio players. Our objective is to give users a control at the decoder side, similar to the Long Play (LP) recording mode at the encoder side in many media recording devices. The main contribution of this paper is a proposal for a Bandwidth and Stereo-image Scalable (BSS) decoding scheme for single-layer audio formats such as MP3. The proposed scheme is based on an analysis of the perceptual relevance of different audio components in the compressed bitstream. The bandwidth and stereo-image scalability directly translates into scalability in terms of the computational workload generated by the decoder. This can be exploited by a voltage/frequency scalable processor to save energy and prolong the battery life.

Categories and Subject Descriptors
D.4.1. [**Process Management**]: Scheduling; D.4.7. [**Organization and Design**]: Real-time system and embedded systems

General Terms
Algorithms, Design, Experimentation, Human Factors.

Keywords
Bandwidth Stereo-image Scalable (BSS), low power processing

1. INTRODUCTION
Power consumption has become a critical constraint in the design of both hardware and software for portable embedded systems. A large fraction of such systems are targeted towards streaming multimedia applications such as audio and video decoding. Typically these applications exhibit a high variability in their computational demand. Many dynamic voltage/frequency scaling and dynamic power management techniques exploit this variability and slow down or switch-off the processor during periods of low demand, thereby saving energy [1,2,3,5]. Clearly, these techniques rely on accurately *predicting* the variation of the computational demand, often using control-theoretic feedback schemes. However, these schemes are themselves computationally expensive and also error prone.

MM'05, November 6–11, 2005, Hilton, Singapore.

To address these problems, in this paper we adopt a different approach towards saving energy. We achieve power efficiency by re-designing the decoder architecture enabling QoS versus power tradeoff. The proposed workload scalable audio decoding scheme allows the user to switch between multiple output quality levels for a single-layer audio format such as MP3. Each such level is associated with a different level of power consumption, and hence battery life.

Our work is mainly motivated by the following observations.

1) *Perceptual characteristics of individual users*: Although most perceptual audio codecs, such as MP3, cover a frequency range of 20 kHz, most adults can hardly hear frequency components above 16 kHz. Further, within the wide swath of frequencies that most people can hear, high frequency bands are perceptually less important than low frequency bands [8]. We can therefore leave those irrelevant high frequency components un-decoded with little perceptual degradation.

2) *Listening environment*: Portable audio players are typically used on the move and in a variety of environments such as in a bus, train, or plane, using simple earpieces. In such noisy environments, it is difficult for most users to distinguish between various playback qualities – they appear to be more tolerant to small quality degradation in such situations.

3) *Service types and signal characteristics*: Different applications and signals require different bandwidths. For example, a storytelling audio clip requires significantly less bandwidth compared to a music clip.

Based on these observations, we propose a Bandwidth and Stereo-image Scalable (BSS) decoding scheme which allows users to control the tradeoff between battery life and decoded audio quality, with the understanding that a slight degradation in audio quality – which may not even be perceptible to the user – can significantly increase the battery life of the player. This novel feature allows the user to tailor the quality level of the decoded audio according to his/her hearing ability, listening environment, and required service type. Although the proposed scheme can be implemented using any perceptual audio decoder, as a proof of concept, we redesign a standard MP3 decoder into a BSS-MP3 decoder.

As shown in Figure 1, the quality level, chosen manually by the user to decode any audio clip, determines: (i) the decoding scheme or algorithm for the specified level, and (ii) the voltage and the frequency with which the processor runs the decoding scheme. When a clip is decoded at a lower quality level, by sacrificing the playback quality slightly, the processor can be run at a lower frequency (and voltage), thereby saving power.

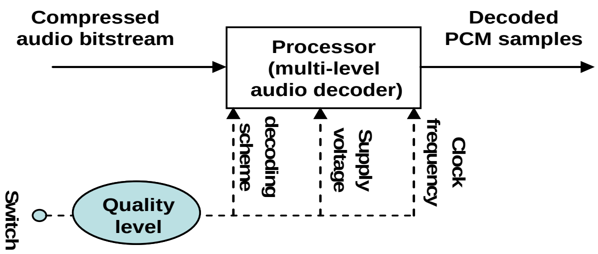

Figure 1. High-level block diagram of the BSS-MP3 decoder.

2. BANDWIDTH AND STEREO-IMAGE SCALABLE DECODING

Our approach is a multi-level decoding strategy which is based on distinguishing the perceptual significance associated with different frequency bands and channels. These two aspects allow the user to eliminate perceptually *irrelevant* computation. The proposed approach can be more effective in workload reduction in comparison with other optimization techniques such as the "do not zero-pute" algorithm in [11], which we classify as eliminating *redundant* computation. In the rest of this section, we introduce the concept of frequency bandwidth scalability and stereo-image scalability for a MP3 bitstream before describing BSS-MP3 decoder.

2.1 Frequency Bandwidth Scalability

For achieving frequency bandwidth scalability, we partition the 32 subbands defined in the MPEG 1 audio standard [6] into four groups, where each group corresponds to a perceptual quality level (see Table 1). We use a sampling frequency of 44.1 kHz as an example in the following discussion.

Group index	Decoded subband index	Frequency range (Hz)	Perceived quality level
Group 1	0-7	0 – 5512.5	AM quality
Group 2	0-15	0 – 11025	Near FM quality
Group 3	0-23	0 – 16537.5	Near CD quality
Group 4	0-31	0 – 22050	CD quality

Table 1. Four frequency band groups

As shown in Table 1, frequency band group 1 covers the lowest frequency bandwidth (5.5 kHz), which we define as the base layer. Although this group occupies only a quarter of the total bandwidth and contributes to roughly a quarter of the total computational workload, it is perceptually the most relevant frequency band. The output audio quality corresponding to this group is certainly sufficient for services such as news and sports commentaries. Group 2 covers a bandwidth of 11 kHz and almost reaches FM radio quality, which is sufficient for listening to music clips, especially in noisy environments. Group 3 covers a bandwidth of 16.5 kHz and produces an output that is very close to CD quality. Group 4 corresponds to the standard MP3 decoder, which decodes the full bandwidth of 22 kHz.

Although the incurred computational workload linearly scales with the frequency bandwidth (Groups 1-4), their perceptual contributions to the playback audio quality are vastly different. In general, the low frequency band (Group 1) is significantly more important than any of the higher frequency bands. This is further illustrated in Section 3.2.

2.2 Stereo-image Scalability

Most MP3 files are encoded using the joint/MS stereo mode. This mode exploits the similarity between the left and the right channel signals. The left and right channels are encoded as the middle (M) and side (S) channel in the MP3 file [6]. The M channel contains the most essential information from both left and right channels, and the S channel is only responsible for providing the stereo image. Hence, the M channel is *perceptually* more significant than the S channel despite the similar computational workload involved in decoding it. As a result, we can choose to partially decode the S channel or even discard it completely, when a smaller workload and hence a longer battery life is desired.

2.3 BSS-MP3 Decoder

The proposed BSS-MP3 decoder combines bandwidth and stereo-image scalability,resulting in a number of decoding levels, each associated with a differenct playback quality.

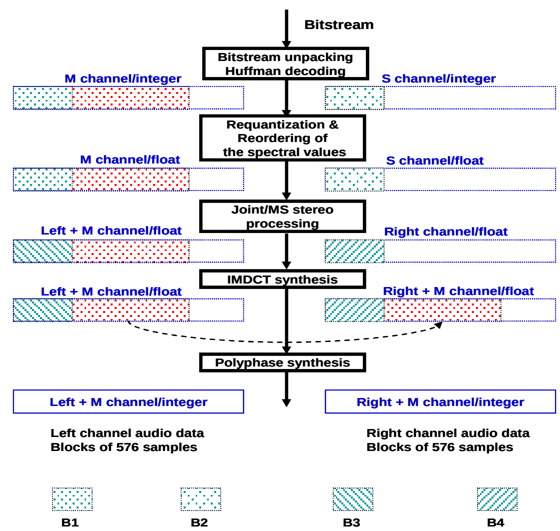

Figure 2. Block diagram of a standard MP3 decoder and the proposed BSS-MP3 decoding scheme. B1-B4 indicate frequency components associated with the different decoding stages.

A block diagram of the standard MP3 decoder [6] is illustrated in Figure 2, along with our new BSS-MP3 decoding scheme. A standard MP3 decoder parses the bitstream, decodes the side information first, and then runs several signal processing modules to convert the MP3 bitstream to PCM audio samples. Three modules which incur the most computational workload are de-quantization, Inverse Modified Discrete Cosine Transform (IMDCT) and polyphase synthesis filterbank. The standard MP3 decoder decodes the entire frequency band, which corresponds to the highest computational workload. In the case of BSS-MP3, depending on the decoding level selected by the user, the above three modules (i.e., de-quantization, IMDCT and polyphase synthesis filterbank) process only a partial frequency range, thereby incurring less computational cost as shown in the Figure 2, where B1 denotes the selected bandwidth for the M channel, B2 for the S channel, B3 for the left channel and B4 for right channel, respectively.

Let the lower *m* subbands of the M channel and the lower *s* subbands of the S channel be decoded by BSS-MP3 decoder, where $s \le m$. This implies a typical asymmetric decoding where we decode more subbands of M channel than the S channel. However, symmetric partial MS decoding is also possible when $s = m$.

As shown in Figure 2, our BSS-MP3 decoder decodes and stores only data from the lower *m* and *s* subbands for M and S channels respectively after Huffman decoding. Due to the absence of the [*s*+1, *m*] subbands in the S channel, [*s*+1, *m*] subbands of the M channel remain unchanged after the joint/MS stereo processing module. They are simply copied to both the left and the right channel, as indicated by the dashed line after the IMDCT module in Figure 2. The only true stereo components are the lower *s* subbands in the left and the right channels.

During the above processes, the computational workload of BSS-MP3 decreases almost linearly with the number of decoded subbands in both the channels.

3. EXPERIMENTAL EVALUATION

3.1 Energy Saving Evaluation

We evaluated the energy savings resulting from our decoder using audio clips with a bitrate of 128 kbits/sec. All the audio clips were of a sampling rate of 44.1 kHz and were coded in the joint stereo mode. Our processor model was based on the Sim-Profile configuration of the SimpleScalar instruction set simulator [12]. We simulated the execution of our decoder with three different playback delay values: 0.5, 1 and 2 secs.

The audio clips we used were all of duration 20 sec and the minimum required processor frequencies to support the different playback quality levels are listed in Table 2. These frequency values were computed by taking into account both, the variability in the processor cycle requirements of the different granules in the MP3 bitstream, as well as the number of bits constituting each granule.

Playback delay	**m:32 s:32**	**m:32 s:24**	**m:32 s:16**	**m:32 s:8**	**m:32 s:0**	**m:24 s:24**	**m:24 s:16**
0.5 sec	322	292	257	221	169	244	211
1.0 sec	314	285	251	216	165	238	206
2.0 sec	299	270	238	205	156	226	195
Playback delay	**m:24 s:8**	**m:24 s:0**	**m:16 s:16**	**m:16 s:8**	**m:16 s:0**	**m: 8 s:8**	**m:8 s:0**
0.5 sec	177	130	162	129	88	85	50
1.0 sec	172	127	158	126	86	83	48
2.0 sec	163	120	150	119	82	78	46

Table 2. Minimum clock frequencies (MHz) at which the processor needs to be run for different quality levels of the BSS-MP3 decoder and for three different playback delays. *m:x* and *s:y* denotes the case where the M channel is decoded using the *x* lower subbands and the S channel is decoded the *y* lower subbands.

As the playback delay was increased from 0.5 sec to 1 and 2 secs, the minimum required frequency showed a moderate decrease for all the decoding levels. The reason behind this is the increased buffering time which smoothens some of the variability in the workload. Figure 3 shows the normalized energy consumption corresponding to 14 different decoding levels with the playback delay always set to 0.5 sec. Clearly, the decoding configuration (m:32, s:32) consumes the highest amount of energy. This corresponds to a standard MP3 output. Compared to this baseline, when an audio clip is decoded at the level (m:8,s:8), which results in a playback quality corresponding to that of an AM radio, an energy savings of 98% is achieved.

It is worth mentioning that our scheme is easy to implement, has no runtime overhead, and does not involve any runtime voltage or frequency scaling. It only adds a useful utility feature, i.e., a *manual switch*, for the user to have more control over the battery life/playback quality tradeoff.

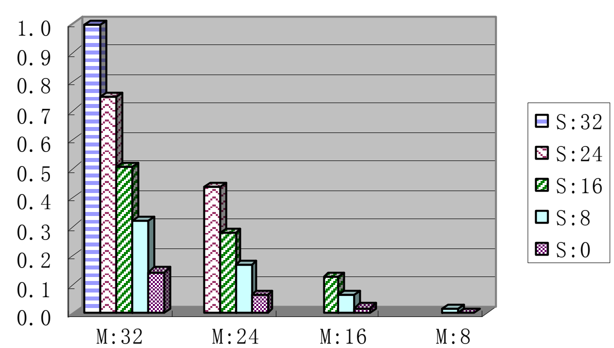

Figure 3. Normalized energy consumption for 0.5s playback delay and the fourteen decoding levels.

3.2 Perceptual Evaluation

To evaluate the effectiveness of our proposed BSS-MP3 decoder, we carried out experiments on a group of 13 subjects (male and female undergraduate and graduate students). All subjects were asked to evaluate the audio quality using the mean opinion score (MOS), which is a five-point scale (5-ecxcellent, 4-good, 3-fair, 2-poor, and 1-bad). We used five music clips for evaluation of which four were pop music and one was pure instrumental classic music. These MP3 clips were all of joint stereo mode, sampling rate 44.1 kHz, with bitrate equal to 128kbits/sec. For each clip, we prepared five different copies for testing. These copies were generated by the BSS-MP3 decoder with profiles of (m:32, s:32), (m:32, s:24), (m:32, s:16), (m:32, s:8), (m:32, s:0), respectively. Each clip had two additional copies with (m:32, s:32) and (m:8, s:8) given as the anchor quality references, the former at a MOS scale of 5 and the latter at MOS 3. For fairness, all test samples were arranged in a random order except for the anchor references.

We performed two sets of subjective evaluations – one for evaluating stereo-image scalability and the other for evaluating bandwidth scalability.

The result of our first evaluation is shown in Figure 4. We can see that discarding S channel completely, i.e., converting stereo to mono (i.e., m:32; s:32 to m:32; s:0) results in noticeable quality degradation. However, if we decode the entire M channel, but decode only the lower quarter of S channel (i.e., m:32; s:32 to m:32; s:8), the quality degradation is negligible. This observation is significant for our scheme, because we need to decode only a quarter of the S channel, yet provide a satisfactory playback audio quality. This profile results in a reduction of processor frequency of 30%, which corresponds to a power savings of approximately 70% (see Table 2).

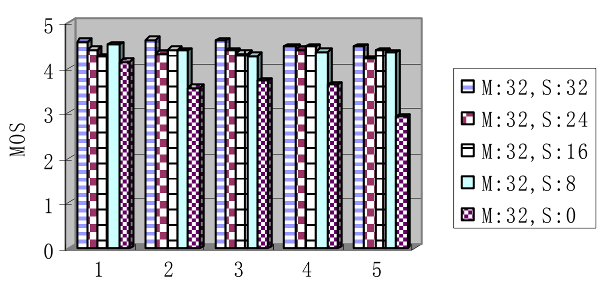

Figure 4. Evaluation results for different BSS-MP3 decoding profiles. The horizontal axis indicates the clip index where indice 1-4 are pop music clips and 5 is an instrumental clip.

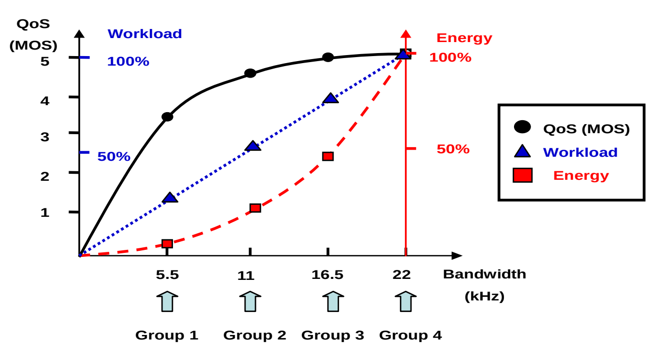

Figure 5. The relationships between the decoded audio bandwidth, perceptual QoS in terms of MOS, workload and energy consumption level.

For the second evaluation, we employ only bandwidth scalability by performing a symmetric M and S channel decoding (i.e., *both* M and S channels are partially decoded). The results are depicted in Figure 5, together with their corresponding workload and energy consumption.

Clearly, we obtain significant power savings, by allowing a slightly degraded output audio quality. Also, it is clear that switching from the frequency band group 4 to 3 results in almost no degradation in the perceptual audio quality, while the reduction in energy consumption is more than 50% which directly translates to an extended battery life.

The workload of the BSS-MP3 decoder is roughly proportional to the frequency bandwidth to be decoded. Reducing the decoded audio bandwidth results in a gradual reduction in audio quality, but a sharp reduction in the energy consumption.

4. CONCLUSION

Very recently we proposed a workload-scalable MP3 decoder in [13], where the bandwidth scalability is considered to reduce workload. In this paper we extended the work in [13] by incorporating both bandwidth and stereo-image scalability. We believe that the proposed scheme provides a new alternative for low-power audio decoding and has the potential to be deployed in future media players. The work reported here is a part of ongoing research; we are currently in the process of implementing our decoder and evaluating it by taking into account factors like OS overhead. We also plan to investigate the application of our scheme in multi-channel surround sound audio formats.

5. REFERENCES

[1] L. Cai and Y.-H. Lu. Dynamic power management using data buffers. In DATE, 2004.

[2] K. Choi, K. Dantu, W.-C. Cheng, and M. Pedram. Frame-based dynamic voltage and frequency scaling for a MPEG decoder. In ICCAD, 2002.

[3] K. Choi, R. Soma, and M. Pedram. Off-chip latency-driven dynamic voltage and frequency scaling for an mpeg decoding. In DAC, 2004.

[4] M. Potkonjak, and G. Qu. Energy minimization with guaranteed quality of service. In ISLPED, 2000.

[5] C. Im, S. Ha, and H. Kim. Dynamic voltage scheduling with buffers for low-power multimedia applications. ACM Transactions on Embedded Computing Systems, 3(4):686–705, 2004.

[6] ISO/IEC 11172-3, Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, 1993.

[7] A. Maxiaguine, Y. Zhu, S. Chakraborty, and W.-F. Wong. Tuning soc platforms for multimedia processing: Identifying limits and tradeoffs. In CODES+ISSS, 2004.

[8] Y. Wang, W. Huang, and J. Korhonen. A framework for robust and scalable audio streaming. In ACM Multimedia, 2004.

[9] W. Yuan and K. Nahrstedt. Energy-efficient soft real-time cpu scheduling for mobile multimedia systems. In 19th ACM Symposium on Operating Systems Principles (SOSP), 2003.

[10] B. Grill. A bit rate scalable perceptual coder for MPEG-4 audio. 103rd AES Convention, 1997, Preprint 4620

[11] P. De Smet, F. Rooms, H. Liong, and W. Philips. Do not zero-pute: An efficient homespun MPEG-Audio layer II decoding and optimization strategy, In Proc. of ACM Multimedia 2004, Oct. 2004.

[12] T. Austin, E. Larson, and D. Ernst. SimpleScalar: An infrastructure for computer system modeling. IEEE Computer, 35(2):59–67, 2002.

[13] S. Chakraborty, Y. Wang, and W. Huang. A perception-aware low-power software audio decoder for portable devices, In Proc. of ESTIMedia 2005