Additive Data Insertion Into MP3 Bitstream Using *linbits* Characteristics

Do-Hyoung Kim, Seung-Jin Yang, Jae-Ho Chung

DSP Lab., Dept. of Electronic Engineering
Inha University, Incheon, Korea
dokim92@chol.com

ABSTRACT

As the use of MP3 audio compression increases, the demand for the insertion of additive data such as copyright information or information on music content itself continues to grow, and related research has also progressed actively. When additive data is inserted in MP3 bitstream, it should not cause any distortion in music quality or change the file size due to the modification of MP3 bitstream structure. To satisfy these conditions, some additive data were inserted in bitstream by modifying some **linbits** among the quantized integer coefficients having big values. The characteristics of **linbits** and their distribution were also considered. As a result of subjective sound quality evaluation through the Mean Opinion Score (MOS) test, the quality of MOS 4.6 was confirmed to be achievable at the data insertion rate of 60 bytes/sec. Using the proposed method, it is possible to insert additive data effectively into encoded bitstream and to retrieve it easily; thus realizing various applications such as audio database management.

1. INTRODUCTION

As usage of the Internet became widespread and digital compression technologies developed rapidly in recent years, accessing many types of information through the Internet became easier for many people. Though these developments are considered positive and desirable, copyright and intellectual ownership issues particularly in music content have surfaced as a result. Producers and owners of published music content have suffered economic damage because of illegal distribution over the Internet, but it is not easy to restrict the practice due to the inherent nature of the Internet. In addition, the development of high-quality audio compression schemes such as the MP3 has made the problem much more serious, since this allowed people to get music content from the Net at little or no cost and high sound quality. Due to the fast-spreading view that inflexible restrictions on digital content have had virtually no effect, many innovative and effective devices that allow people to obtain music content from the Net legally are readily being developed. One of the hot issues among these developments is related to additive data insertion technology, which inserts information on the owner and distributor of the content itself into bitstream; thus enabling the owners of the content to claim ownership over the product later, if such need arises.

This watermarking technology can be manipulated to protect media in various forms, including copyright protection, fingerprinting, copy protection, broadcast monitoring, data authentication, and indexing [1,2]. In particular, copyright protection or fingerprinting schemes are sought-after technologies since they allow for the insertion of copyright information or fingerprints to the content for tracing the copy path and consequently for claiming the original copyright in the future. For this purpose, it is necessary to make the additive data hidden within the music content and unrecognized by way of loss in sound quality or change in media file.

Recently developed watermarking technologies may be classified into two: techniques in time domain [3,4,5] and in spectral [6] or cepstral domain [7].

This study proposed new information or watermarking insertion technique that satisfies the abovementioned conditions and evaluated its performance. The proposed method enables the insertion of additive data into MP3 (MPEG-1 Layer III) bitstream, which is an unrivaled digital audio compression scheme among those that are currently available and popular. This method can be considered as a data insertion technique in the bitstream domain, because it inserts data through the modification of quantized coefficients in integer form. It can be used in many applications requiring additive information operation such as copyright protection and fingerprinting.

2. MP3 ENCODING AND *linbits* [8]

While detailed expressions are not treated in this paper, basic knowledge of quantization and Huffman encoding of MP3 are discussed.

2.1. Quantization

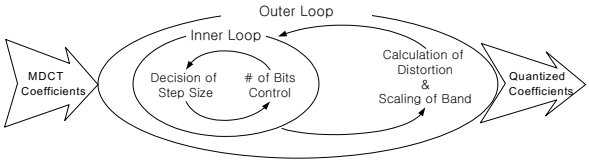

Fig. 1. Iteration Loop Performing Quantization

A simple description of quantization procedure is shown in Fig. 1. Actual quantization of MP3 is performed in the inner iteration loop doing the rate control. This inner iteration loop is first done in the outer iteration loop, which controls the distortion.

Coefficients transformed in the MDCT module have the form of floating points. These values are changed into the integer type by the non-linear quantizer, as represented in Equation (1).

$$ix(i) = \text{nint}\left(\left(\frac{|xr(i)|}{\sqrt[4]{2}^{\,qquant+quantanf}}\right)^{0.75} - 0.0946\right) \quad (1)$$

where xr(i) is the i-th value of the 576 MDCTed coefficients, qquant+quantanf the quantization step size, nint()the nearest integer, and ix(i) the quantized integer of i-th coefficient. Although the range of ix(i) is not restricted explicitly, it practically has its maximum value in the range of tens to hundreds.

Quantization step size is decided in the inner iteration loop considering the number of available bits. In other words, for the given quantization step size, the inner iteration loop calculates the number of bits required to encode the current granule.
From the calculation above, if the given step size is permitted to encode the current granule without exceeding the maximum available bit, the outer iteration loop begins to work. This loop calculates the distortion that could occur depending on the step size decided in the inner loop and controls any distortion by the amplification of each critical band.

2.2. Huffman Encoding And *linbits*

Most of coefficients in the ending part of 576 quantized integer coefficients are zero. This is because general audio signal has low energy in the high-frequency band. This part is encoded by the number of continuing zeros as couple (***rzero***). From the end of the zeros to the direction towards the beginning, coefficients also have relatively small values; thus, continuing -1, 0, and 1 are encoded by a quadruple Huffman table as quadruples (***count1***). The rest of the coefficients except ***rzero*** couples and ***count1*** quadruples are encoded by the 32 Huffman code tables as couples. This part, called ***bigvalues***, is again subdivided into three ***subregions*** that use an independent Huffman table. The matter of which table should be used for each ***subregion*** is decided by considering the maximum magnitude of corresponding ***subregions.*** The Huffman table can handle a maximum value of 15; if the maximum of a certain region is greater than 15 (Excess-15 case), the ESC mode begins to work. In ESC mode, if the current quantized integer coefficient is bigger than 15, the value of 15 is encoded by the corresponding Huffman table; the difference between the coefficient and 15 is encoded directly as a binary form. The maximum number of bits to be used in encoding these Excess-15 values is referred to as ***linbits***.

For the operation of the abovementioned strategy, MP3 has many Huffman tables of the same codes and of different ***linbits***. From tables 16 to 23, each has the same codebook but different ***linbits***. Tables 24 to 31 operate in the same pattern. Table 1 summarizes these characteristics of the Huffman code table.

For example, suppose that the maximum value of a current ***subregion*** is 25. For the expression of 25, additive bits must cover the maximum difference between 25 and 15, i.e., 10. Thus, the value of the ***linbits*** in this case has to be 4 bits. If the current coefficient is 21, 15 is encoded by a corresponding Huffman table (table 19 or 24, in this case), and 6 (21-15=6) is encoded into bitstream using 4bit ***linbits*** as the form of $0110_{(2)}$.

Table 1. Properties of the Huffman Code Table

Table No.	Max Code	ESC on/off	*linbits*	Table No.	Max Code	ESC on/off	*linbits*
0	0	off	0	10-12	7	off	0
1	1	off	0	13	15	off	0
2-3	2	off	0	14	Not Used		
4	Not Used			15	15	off	4
5-6	3	off	0	16-31	15	on	4-13
7-9	5	off	0				

3. INSERTION OF ADDITIVE DATA

3.1 Data Insertion in ESC Mode

As mentioned earlier, the Huffman code tables for MP3 encoding are designed to express the maximum 15 of quantized integer coefficients. If a bigger value than 15 occurs, ESC mode starts to work using tables 16-31.

In ESC mode, additive data were inserted by modifying some LSBs among the corresponding ***linbits***. Since any modification of ***linbits*** may cause serious distortion of sound quality, the modification of a portion of the ***linbits*** was considered and an MOS test performed to evaluate any defect in sound quality in various situations.

Under Huffman tables 16-31, the minimum and maximum ***linbits*** are 1 bit and 13 bits, respectively. For purposes of simplicity, the number of modified bits was fixed as shown in Table 2. As seen in this table, LSB 1 bit is modified for 1 bit ***linbits*** case, and LSB 2 bits are modified for 2 or more bits ***linbits*** case.

Table 2. Number of Modified Bits in ***linbits***

Table No.	Linbits	Number of Modified Bits
16	1	1
17-31	2-13	2

3.2 Manipulation of Data Insertion Rate Using *bigvalues*

Although different from the characteristics of a music signal, the ESC mode is a relatively frequently occurring event. Unless the number of modified coefficients was restricted, severe distortion of sound would not be avoided. Thus, the frequency of modified coefficients should be controlled reasonably to prevent any defect in sound quality despite the decrease in the data insertion rate.

This study evaluated what kind of music samples causes more frequent ESC mode to occur. A list of samples used in this evaluation and their frequency of Excess-15 coefficients are represented in Table 3.

As shown in this table, the frequency of Excess-15 value occurring is as varied as the category of samples; it is more frequent in the samples of low-frequency concentrated energy and less frequent in the samples of widely spread energy. The spectra of the extreme examples are shown in Figs. 2 and 3. Fig. 2 presents the spectrum of "andrea2" in Table 3 and Fig. 3 that of "radio."

The phenomenon of the highly occurring Excess-15 value is caused by the bits allocated in the quantization procedure that are concentrated in the low-frequency band. In contrast, less occurrence of Excess-15 is caused by a fixed amount of bits that have to be used for a wide range of the frequency band. Thus,

the main factor that decides the amount of linbits occurring can be said to depend on how widely the energy of a signal is spread over the frequency band.

Table 3. Number of Occurrence of Excess-15 Quantized Values as Samples

Name	Len. (sec.)	# of Occ.	Occ. /Sec.	Name	Len. (sec.)	# of Occ.	Occ. /Sec.
piano	23.4	14871	635.5	beetho2	13.6	3833	282.4
oboe	14.5	2666	183.3	big	11.4	652	57.0
brass	6.1	3590	592.6	charl1	11.6	4610	398.0
pipeorgan	6.1	3373	548.5	john2	17.2	7319	425.2
saxophone	7.5	188	25.2	kpop	10.5	1439	136.8
wash1	12.0	4396	366.2	newday	10.5	541	51.7
andre1	14.4	8204	570.1	radio	11.8	196	16.6
andrea2	27.2	26564	975.9	secret2	16.4	6937	422.0
andrea3	20.9	3562	170.1	Stan1	14.6	1951	132.2
beetho1	10.8	2401	222.5	chorus	15.4	1538	100.0

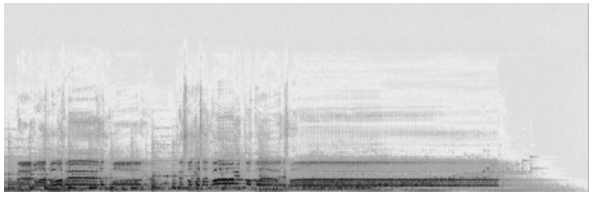

Fig. 2. Sample Spectrum With More Excess-15 Quantized Values (Low-Frequency Concentrated Energy)

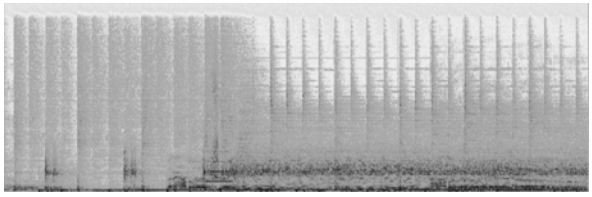

Fig. 3. Sample Spectrum With Less Excess-15 Quantized Values (Widely Spread Energy)

The variance of energy -- i.e., the distribution of spectrum -- can be judged by various criteria such as the relative proportion of each narrow-frequency band energy. Nonetheless, MP3 algorithm has a good index of this judgment without any additional calculation, i.e., by using ***bigvalues.*** As mentioned in the previous section, ***bigvalues*** indicates the range of relatively big coefficients and consequently serves as the starting point of the coefficients that have to apply the general Huffman table. For easier understanding, a simple array model of 576 quantized coefficients is shown in Fig. 4. In this figure, ***bigvalues*** is M, ***count1*** is N, and ***rzero*** is 576-(M+N).

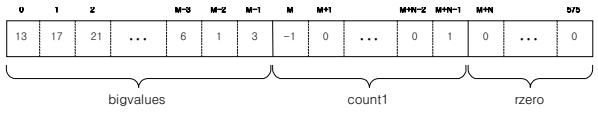

Fig. 4. Diagram of Quantized Coefficients Array

To further explain the concept of bigvalues, a high value means that valid quantized coefficients are spread over a higher frequency band, while a small value means that valid coefficients are concentrated in a low-frequency band. Thus, by observing this value, the deviation of the occurrence of Excess-15 quantized values can be traced, which is generated by the difference of energy distribution as audio samples.

Based on the abovementioned facts, the number of modified Excess-15 coefficients was limited according to the value of ***bigvalues*** (Table 4).

Table 4. Constraint of Modification as ***bigvalues***

Range of ***bigvalues***	Max # of Modified Coefficient in ***subregion***
bigvalues < 200	3
200 < bigvalues < 300	5
300 < bigvalues < 400	7
400 < bigvalues	No Limit

The energy concentrated in the low-frequency range can also cause concentrative modification of some lower-scale factor bands, consequently resulting in a serious audible distortion; hence the limit. In contrast, if ***bigvalues*** is big, the probability of Excess-15 occurrence is relatively small and the modified coefficients are distributed over a wide frequency band because of a widely-spread signal spectrum; thus leading to less sound distortion. In this case, there was no need to limit the number of modified coefficients.

3.3 More Dispersion to Scale Factor Bands for Preventing Concentration

After experimenting on the abovementioned idea, only an MOS reading of 4.2 was obtained, which is not considered good sound quality. For practical applications, an MOS of 4.5 or more is needed, preferably at the full score of 5. After analyzing this audible distortion, the limits in 3, 5, and 7 proposed in Table 4 were again found to be concentrated at the start of the ***bigvalues,*** and their concentration contributed to the distortion. To solve this problem, the modification was dispersed by preventing the sequential occurrence in the same scale factor band. For example, when the ***bigvalues*** reading is at 250 and Excess-15 coefficients occur continuously from the first to the tenth position, the 5 candidate coefficients are selected in different scale factor bands to prevent concentrative bit change. If the ***bigvalues*** reading is more than 400 (Table 4), the modification is not limited by this dispersion. Using the dispersion of modified coefficients to many scale factor bands, the total sound quality could be improved.

4. EVALUATION OF SOUND QUALITY AND ITS RESULTS

4.1 Subjective Test of Sound Quality Using MOS

This section explains the method and the results of sound quality evaluation performed for the abovementioned additive data insertion idea. A total of 16 male and 4 female university students participated in the listening test. Although not professionally trained for such a listening test, the participants said they enjoyed listening to various kinds of music. The name and the length of samples used in the experiments are shown in Table 3. These samples were selected to represent the different music genres. Actual test samples of the same music name were extracted in different positions representing the different moods of that music.

An MP3 music sample encoded at 128 kbps without additive data (A) was first presented. MP3 music at 128 kbps with additive data (B) was then presented to listeners. Each listener was asked to rate the sound quality of music B, using the assumption that there was no distortion in music A as shown in the following evaluation index scoring 1(Very annoying) to 5(No audible distortion). If a listener wanted to listen to the presented samples again, that music was presented repeatedly.

Samples were played using a general MP3 player software in Pentium PC with devised high-performance sound adapter and a high-performance monitoring headphone. For a worst-case sound quality scenario, additive data was selected to have the maximum distance from the original data as shown in Table 5.

Table 5. Bit Modification Condition for Making the Worst Sound Quality

Linbits	Original LSB Code	Modified LSB Code
1 bit case	0	1
	1	0
2 bits case	00	11
	01	11
	10	00
	11	00

4.2 MOS Results and Data Insertion Rates

Table 6 shows the averages and variances of MOS and data amount per second inserted in bitstream. The average MOS reading was 4.6, and the data amount inserted in a second was about 63 bytes (about 500 bits). This is sufficient for adding about 60 ASCII data into bitstream, which can be used to insert information about music content such as the singer's name, title of the music or lyrics for a pop song, as well as copyright information.

Table 6. Experiment Result: MOS Value and Data Insertion Rate

Sample Name	MOS	Var.	Insertion Rate (bytes/sec)	Sample Name	MOS	Var.	Insertion Rate (bytes/sec)
pipeorgan	4.85	0.13	131	beetho2	4.6	0.25	58
piano	4.45	0.26	114	john2	4.7	0.33	96
brass	4.45	0.37	106	andrea2	4.75	0.20	126
oboe	4.5	0.37	44	newday	4.5	0.58	12
saxophone	4.15	0.66	5	charl1	4.6	0.25	97
andre1	4.7	0.22	126	andrea3	4.8	0.17	36
beetho1	4.7	0.22	45	big	4.8	0.17	13
stan1	4.5	0.37	28	wash1	4.75	0.20	72
radio	4.15	0.66	3	kpop	4.55	0.26	32
secret2	4.6	0.25	86	chorus	4.65	0.34	20
Average	**MOS : 4.59**		**Var. : 0.31**		**Insertion Rate : 63 bytes/sec**		

As can be seen from the table, however, there is a very low MOS average for some samples such as for "saxophone" or "radio." This is because two listeners gave very low scores of MOS 2 for these samples. That fact can be explained from the relatively wide variance of corresponding samples. For the sample "newday," one listener gave an MOS 3 rating for that sample; hence the low score and big variance. These results were not initially understood because those samples have much less data insertion rate, i.e., considerably less modification of bitstream. These listeners were eventually found to be prejudicial to the samples presented. For example, one listener was very much familiar with the "radio" and "newday" samples, but the unfamiliar listening environment during the experimentation process disoriented him. In the case of the "saxophone" sample, listeners felt a certain sense of nervousness because it was dynamically played by a tenor saxophone. Consequently, some listeners made their evaluations based on their favorite kind of music rather than on sound quality.

Another problem found in Table 6 is the large deviation in the data insertion rate among the samples. Nonetheless, this should not be taken seriously when the music content as a whole is considered, not just some parts of the music representing its property. For example, "radio," which had a very low data insertion rate, and "andre1," one of the samples representing the opposite case, had data insertion rates for their whole music content of 40 bytes/sec and 88 bytes/sec, respectively. Thus, when the whole music content is considered, deviations in the data insertion rate should not be a serious problem.

5. CONCLUSION

In this study, additive data was inserted in MP3 bitstream modifying the LSBs of ***linbits*** generated for quantized coefficients of Excess-15, and their performance evaluated. In addition, using ***bigvalues*** characteristics minimized the audible distortion of music by considering the distribution of energy and by preventing the concentration of coefficient modification. Through some experiments of subjective sound quality participated in by an untrained test group involving 20 men and women, the sound quality of MOS 4.6 was found to be achievable at the insertion rate of about 63 bytes/sec. The proposed method can be applied to various applications that need additive data insertion including copyright protection.

This study will contribute to develop the algorithm to guarantee the fixed rate of additive data and to ensure higher security against unauthorized intruders for better copyright protection.

6. REFERENCES

[1] Gerhard C. Lanelaar, Iwan Setyawan, and Reginald L. Lagendijk, "Watermarking Digital Image and Video Data," IEEE Signal Processing Magazine, Vol. 17, No. 5, pp. 20-46, Sept. 2000

[2] Michael Arnold, "Audio Watermarking: Features, Applications and Algorithms," Proc. of IEEE International Conference on Multimedia and Expo (ICME 2000), Vol. 2, pp. 1013 -1016, 2000

[3] P. Bassia and I. Pitas, "Robust Audio Watermarking in the Time Domain," Proc. of Ninth European Signal Processing Conference (Eusipco-98), pp. 25-28, 1998

[4] Mohamed F. Mansour and Ahmed H. Tewfik, "Audio Watermarking by Time-Scale Modification," Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01), Vol. 3, pp. 1353 -1356, 2001

[5] L. Boney, A. H. Tewfik, and K. N. Hamdy, "Digital Watermarks for Audio Signals," Proc. of the Third IEEE International Conference on Multimedia Computing and Systems, Vol. 3, pp. 473-480, 1996

[6] J. Herre, C. Neubauer, F. Siebenhaar, and R. Kulessa, "New Results on Combined Audio Compression/Watermarking," Proc. of AES 111th Convention, 2000

[7] S. K. Lee and Y. S. Ho, "Digital Audio Watermarking in the Cepstrum Domain," IEEE Trans. on Consumer Electronics, Vol. 46, No. 3, pp. 744-750, Aug. 2000

[8] ISO/IEC 11172-3, Information Technology - Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbits/s, Part 3. Audio, 1993