

# データハイディングを用いる MPEG オーディオ向け準可逆ローパスフィルタ

萩野谷 俊典<sup>†1</sup> 田中 清<sup>†1</sup> 高木 幸一<sup>†2</sup>

本稿では、MPEG オーディオ信号に対する、データハイディングを用いる準可逆なローパスフィルタの実現方法を提案する。提案法では、MPEG-1 オーディオレイヤー II による圧縮音声をも、まず指定したあるサブバンド番号以降の高域成分をカットして帯域制限する。次に、この音声を帯域制限前の全体の符号量で再符号化する際に生じる冗長成分に、カットした高域成分を埋め込む。一方元の圧縮音声の復元は、まず高域成分を埋め込んだ再符号化音声を、帯域制限後の低域成分の符号量で再々符号化して元の低域成分を復元する。次に、復元した低域成分の割り当てビット数を利用して再符号化音声の冗長成分から高域成分を抽出する。最後に、復元した低域成分と高域成分を合成して元のオーディオ信号を再構成する。いくつかの音楽データを用いたシミュレーション実験により、これら一連の処理によって元のオーディオ信号に近い信号を再生できることを示す。

## Quasi-Reversible Low-pass Filter for MPEG Audio Using Data Hiding

TOSHINORI HAGINOYA,<sup>†1</sup> KIYOSHI TANAKA<sup>†1</sup>  
and KOICHI TAKAGI<sup>†2</sup>

In this paper, we propose a method to realize quasi-reversible low-pass filter for MPEG audio using data hiding. In this work, we utilize compressed audio by MPEG-1 audio layer II, and we first degrade the audio signal by cutting high-frequency components (HFCs) located after a sub-band specified. Then, we embed HFCs eliminated into the redundant bits caused by re-encoding the band-passed audio using a total amount of bits allocated to all sub-bands before band-limitation. On the other hand, to reconstruct the original compressed audio, we first restore low frequency components (LFCs) by re-encoding the re-encoded audio using a total number of bits allocated to only low-frequency sub-bands located before the specified sub-band. Then, we extract HFCs from redundant components in re-encoded audio by using the number of bits allocated to LFCs. Finally, we reconstruct the original compressed audio by

combining the restored LFCs and the extracted HFCs. Through computer simulation using several music data, we show that we can regenerate audio signal that is close to the original compressed audio.

### 1. はじめに

圧縮符号化されたデジタル音声<sup>1)-4)</sup>は、パソコン、iPod、携帯電話などの電子機器やネットワークを介したストリーミング配信、通信カラオケ、デジタル放送などにおいて幅広く利用されている。しかし、デジタル音声は取り扱いが容易なため無許可コピーや違法配信の問題が後を絶たず、音声コンテンツを適切に流通・管理できる技術的な解決策が模索されている。従来、音声信号に対する暗号化<sup>5)</sup>や電子透かし<sup>6)</sup>が検討されてきたが、暗号化は鍵を用いて復号しなければ音声を利用することができず、電子透かしは埋め込まれている透かしデータを正しく検出できなければ有効に機能しない弱点がある。とくにコンテンツ提供者の立場からは、音声の品質を柔軟に制御することができれば、利用者レベルに応じたコンテンツ配信・管理が可能となり有用である。

このような立場から、本稿では、デジタル音声の品質制御を可能とする、データハイディングを用いる準可逆なローパスフィルタの実現方法を提案する。提案法では、MPEG-1 オーディオレイヤー II<sup>1)-4)</sup>により符号化した圧縮音声を利用し、これをまず指定したあるサブバンド番号以降の高域成分をカットして、故意に品質を劣化させた音声を作成する。次に、この帯域制限した音声を、帯域制限前の全帯域に割り当てられていた符号量で再符号化する。この再符号化は冗長な符号化となり、低域成分の割り当てビット数が増加するため<sup>7)</sup>、この冗長成分に帯域制限でカットした高域成分を埋め込む。一方元の圧縮音声を復元するためには、まず高域成分を埋め込んだ再符号化音声を、帯域制限後の低域成分の符号量で再々符号化して低域成分を復元する。次に、復元した低域成分の割り当てビット数を利用して、再符号化音声の冗長成分(増加割り当てされたビット)から高域成分を抽出する。最後に、復元した低域成分と抽出した高域成分を合成して、元のオーディオ信号を再構成する。いくつかの音楽データを用いたシミュレーション実験により、提案法の性能を検証する。

<sup>†1</sup> 信州大学工学部

Faculty of Engineering Shinshu University

<sup>†2</sup> KDDI 研究所

KDDI R&D Laboratories Inc.

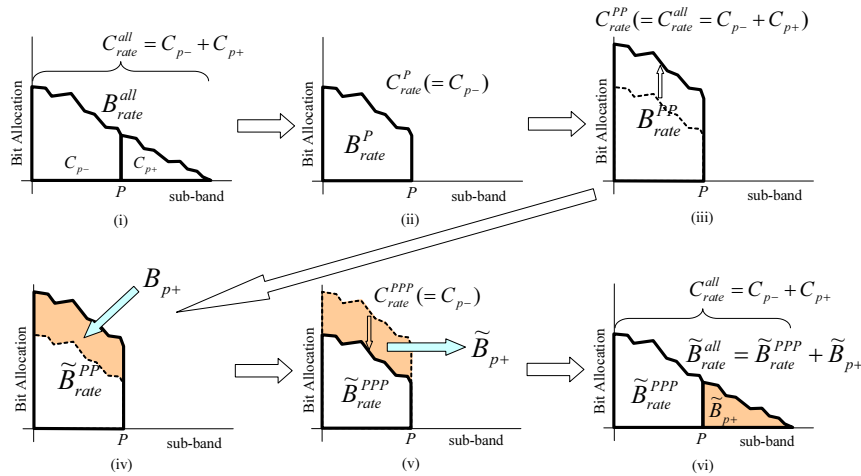


図 1 一連の処理の概念  
Fig. 1 Concept of series of processing

## 2. 一連の処理の流れ

本稿では、MPEG-1 オーディオレイヤー II 符号化の動的ビット割り当て処理に注目し、その一部に必要な修正<sup>7),8)</sup>を加えて、準可逆なローパスフィルタを実現するための一連の処理を行う。その概念を図 1 に示す。

### 2.1 帯域制限と高域成分の埋め込み

まず、MPEG-1 オーディオレイヤー II による圧縮音声で帯域制限し、これを再符号化して生ずる冗長成分に高域成分を埋め込む手順を示す。(図 1, 図 2 参照)

Step1: オリジナル音声を符号化して、圧縮符号  $B_{rate}^{all}$  を生成する(図 1(i)). ただし、 $rate$  は符号化レート、 $all$  はすべてのサブバンドを利用していることを示す。また、図中の  $C$  は符号量を表し、全帯域に割り当てられる符号量  $C_{rate}^{all}$  は、あるサブバンド  $P$  以前の低域成分の符号量  $C_{p-}$  と  $P$  以降の高域成分の符号量  $C_{p+}$  の和で表される。

Step2:  $B_{rate}^{all}$  のサイド情報とサンプル値を復号し、指定したあるサブバンド番号(遮断周波数  $P$ )以降の割り当てビット数を 0 に書き換え、再びサイド情報とサンプル値を符号化して、帯域制限した圧縮符号  $B_{rate}^P$  を生成する(図 1(ii)). このとき、遮断周波数  $P$  よりも低域に割り当てられている符号量  $C_{p-}$  を記録しておく。

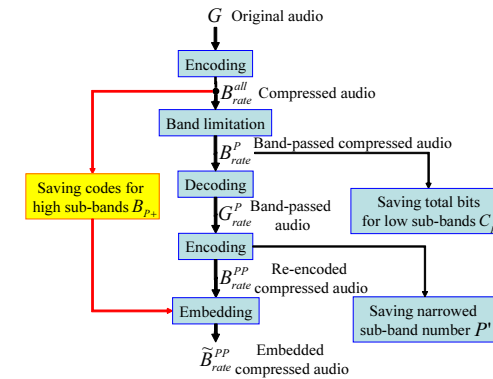


図 2 帯域制限から再符号化による高域成分の埋め込み  
Fig. 2 From band-limitation to embedding HFCs by re-encoding

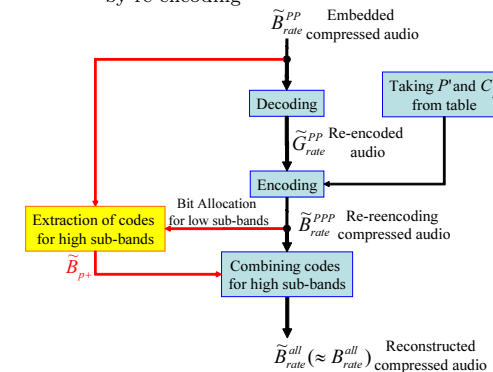


図 3 再々符号化による高域成分の抽出と圧縮音声の再構成  
Fig. 3 Extraction of HFCs by re-rencoding and reconstruction of compressed audio

Step3:  $B_{rate}^P$  を復号し、PCM 信号  $G_{rate}^P$  を再生する。

Step4:  $G_{rate}^P$  を帯域制限前の全体の符号量  $C_{rate}^{all}$  で符号化し、冗長な圧縮符号  $B_{rate}^{PP}$  を生成する(図 1(iii)). このとき、動的ビット割り当てではサブバンド番号  $0 \sim P-1$  の範囲で行うが、文献 7) の方法で  $P' (< P)$  の範囲に収める。この  $P'$  の値も記録する。

Step5:  $B_{rate}^{PP}$  の冗長成分に、帯域制限によってカットされた高域成分  $B_{p+}$  を以下のように

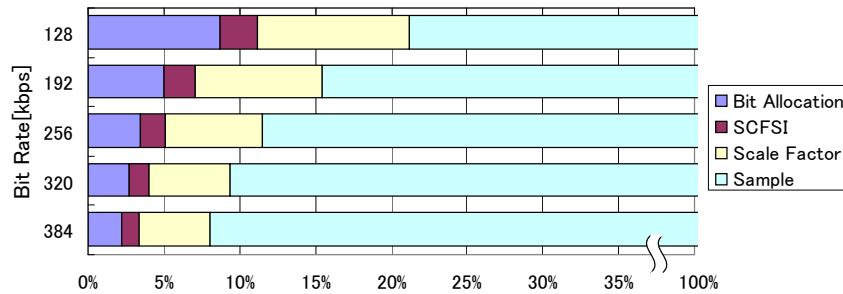


図 4 埋め込む高域成分の各符号の割合  
 Fig. 4 Ratio of each code in HFCs to be embedded

埋め込む (図 1(iv)).

Step5-1: 遮断周波数  $P$  以降の高域成分をビットに分解し, Bit Allocation 値, SCFSI 値, Scale Factor 値, Sample 値の順に一次元配列に格納する (3, 4 参照).

Step5-2:  $B_{rate}^{PP}$  の冗長成分を LSB 側から MSB 側まで順番に走査し, 一次元配列内のビットを順次挿入していく (5 参照).

Step6: 高域成分を埋め込んだ圧縮符号  $\tilde{B}_{rate}^{PP}$  を保存する.

## 2.2 高域成分の抽出と元の圧縮音声の再構成

次に, 再符号化した帯域制限音声から元の低域成分を復元するとともに高域成分を抽出し, これらを合成して元の圧縮音声を再構成する手順を示す. (図 1, 図 3 参照)

Step1: 高域成分が埋め込まれた圧縮符号  $\tilde{B}_{rate}^{PP}$  を復号し, PCM 音声  $\tilde{G}_{rate}^{PP}$  を再生する.

Step2: 記録しておいた  $P'$  と  $C_{p-}$  を取り出す.

Step3:  $\tilde{G}_{rate}^{PP}$  を低域成分の符号量  $C_{p-}$  で符号化し, 再々符号化した圧縮符号  $\tilde{B}_{rate}^{PPP}$  を生成する (図 1(v)).

Step4:  $\tilde{B}_{rate}^{PPP}$  の割り当てビット数を用いて, 再符号化音声  $\tilde{B}_{rate}^{PPP}$  の高域成分が格納されている冗長成分の位置を特定する.

Step5:  $\tilde{B}_{rate}^{PPP}$  の冗長成分から, 埋め込んだ高域成分  $B_{p+}$  を以下のように抽出する.

Step5-1: 冗長成分を LSB 側から MSB 側まで順番に走査し, 走査したビットを一次元配列に取り込む (5 参照).

Step5-2: これを Bit Allocation 値, SCFSI 値, Scale Factor 値, サンプルビットの順に展開し,  $B_{p+}$  を再構成する (3, 4 参照).

Step6: 再々符号化した圧縮符号  $\tilde{B}_{rate}^{PPP}$  ( $\approx B_{p-}$ ) に抽出した高域成分  $\tilde{B}_{p+}$  を合成し, 元の圧縮符号  $\tilde{B}_{rate}^{all}$  を再構成する (図 1(vi)).

## 3. 埋め込む高域成分の取り扱い

ここでは, 再符号化した圧縮符号  $B_{rate}^{PP}$  の冗長成分に, 遮断周波数  $P$  以降の高域成分  $B_{p+}$  を, 限られた容量の中で効率よく埋め込むための取り扱い方法を示す.

### 3.1 高域成分に含まれる符号の種類

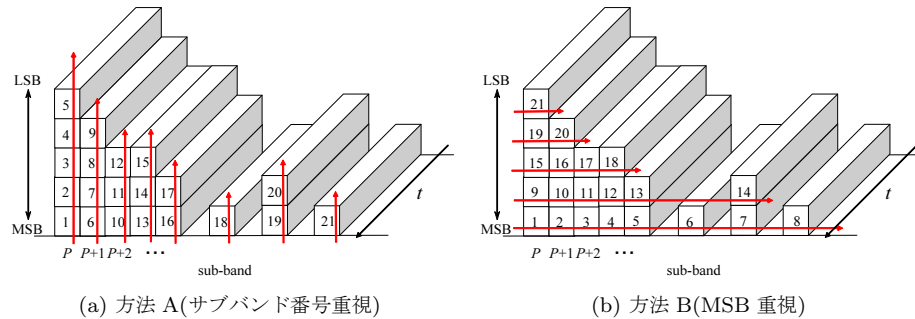
- Bit Allocation: サンプル値の符号長を決定する値である. この値が 1 以上ならばサブバンドに信号があることを示し, SCFSI, Scale Factor, Sample の各値も存在する. 符号化の基準となる最も重要な符号である.
- SCFSI: Scale Factor の伝送方法を指定する符号で, 各サブバンドに 3 つ存在する Scale Factor 値のうち, 何個用いて符号化するかを示す.
- Scale Factor: 各サブバンド毎の 12 サンプルの最大振幅を表す.
- Sample: 各サブバンドに 36 個存在し, サブバンドフィルタバンクを通して得た周波数係数を正規化し量子化した値である. サンプル値は, Bit Allocation 値から決定される符号長 (ビット数) で符号化されている.

### 3.2 各符号の情報量と埋め込み時の優先度

まず, 1 フレームサイズにおける各符号の情報量を考える. すべてのサブバンドにおいて必ず符号化される Bit Allocation の情報量は常に一定である. SCFSI と Scale Factor は, 割り当てビット数が 1 以上のサブバンドに存在する. SCFSI は 2 ビット, Scale Factor は SCFSI の値によって 6~18 ビット使用される. Sample は, Bit Allocation 値によって定められたビット数で符号化されるため, 各サブバンド毎の情報量 (符号長) は異なる. 通常, 1 フレームの大半をこの Sample のビット数が占めるが, 図 4 に示すように, 符号化レートが高くなるほどサンプルビットの占める割合は大きくなる.

ところで, 再符号化した圧縮符号の冗長成分の容量は,  $B_{rate}^{all}$  の高域成分  $B_{p+}$  のサイズとほぼ同じになる. 動的ビット割り当て処理では, 1AAU のサイズから Header (符号化レート, サンプリング周波数, チャンネル数を定義), CRC (エラーチェック), Bit Allocation, Ancillary (利用者が任意に定義できるデータ) を差し引いた残りを, SCFSI, Scale Factor, Sample の 3 つの成分に割り当てる. このため, 提案法で埋め込む高域成分  $B_{p+}$  のサイズは, この 3 つの成分の合計ビット数となる.

一方, 埋め込んだ高域成分を再構成するためには, SCFSI, Scale Factor, Sample とと



(a) 方法 A(サブバンド番号重視) (b) 方法 B(MSB 重視)

図 5 高域成分のサンプルビットの走査方法  
Fig. 5 Scanning of sample bits in HFCs

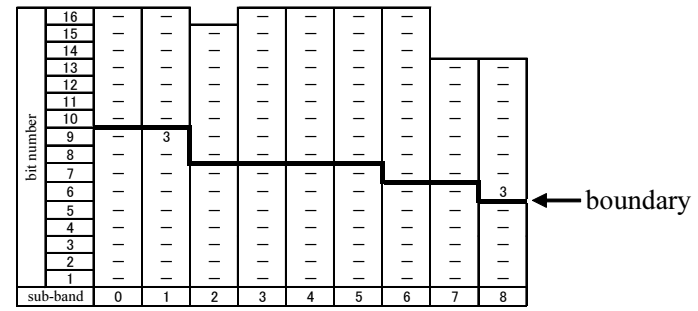


図 6 ある 1 フレームにおける割り当てビット数の誤り頻度 (100 回試行)  
Fig. 6 Frequency of errors in number of bits allocated(among 100 trials)

表 1 帯域制限音声 ( $B_{rate}^P$ ) と再々符号化音声 ( $\tilde{B}_{rate}^{PPP}$ ) の割り当てビット数の変化の統計量  
Table 1 Statistics for differences between number of bits in  $B_{rate}^P$  and  $\tilde{B}_{rate}^{PPP}$

		サブバンド番号									
BitRate[kbps]		0	1	2	3	4	5	6	7	8	
平均	128	0.12	0.06	-0.03	-0.03	-0.02	-0.02	-0.02	-0.02	-0.03	-0.03
	192	0.07	0.03	0.01	-0.01	-0.02	-0.02	-0.02	-0.01	-0.02	
	256	0.05	0.03	0.01	0.02	0.00	-0.05	-0.01	-0.01	-0.01	
	320	0.03	0.04	0.02	0.02	0.01	0.00	0.00	-0.06	-0.04	
分散	384	0.01	0.02	0.01	0.01	-0.01	0.00	0.00	-0.03	0.00	
	128	0.151	0.084	0.025	0.030	0.018	0.020	0.022	0.035	0.039	
	192	0.116	0.112	0.093	0.014	0.021	0.021	0.018	0.010	0.017	
	256	0.085	0.073	0.086	0.102	0.093	0.051	0.012	0.007	0.011	
	320	0.050	0.062	0.066	0.069	0.058	0.062	0.055	0.058	0.041	
	384	0.042	0.032	0.038	0.038	0.037	0.039	0.041	0.064	0.075	

もに、これらがどのサブバンドに存在するかを示す Bit Allocation が必要である。しかし、 $\tilde{B}_{rate}^{PPP}$  の高域成分の割り当てビット数は 0 に書き換えられている (2.1 の Step2) ため、これを  $B_{p+}$  と一緒に保存しておかなければならない。すなわち、SCFSI, Scale Factor, Sample 分の容量に、高域成分の Bit Allocation 値も格納しなければならず、容量を超過してしまう。Bit Allocation, SCFSI, Scale Factor は、どれも 1 ビットでも誤ると高域成分の再現に大きな影響が出る。このため、本稿では埋め込みの優先順位を (1)Bit Allocation, (2)SCFSI, (3)Scale Factor, (4)Sample とし、サンプル値に割り当てられる符号を最後に挿入する。

#### 4. サンプルビットの走査方法

上述の通り、高域成分  $B_{p+}$  の埋め込みにおいて、格納容量の問題からサンプルビットの

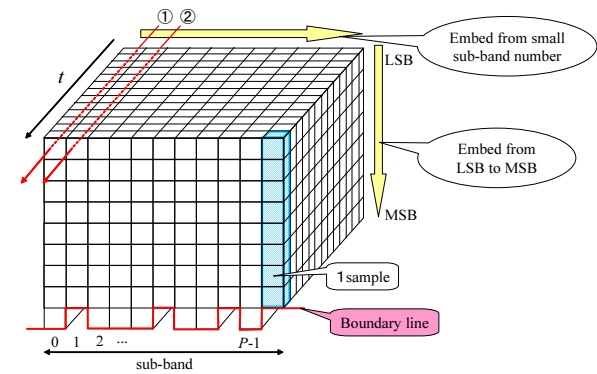


図 7 低域冗長成分の走査方法  
Fig. 7 Scanning of redundant bits in low sub-bands

すべてを埋め込むことはできない。そこで本稿では、サンプルビットの走査方法 (埋め込み順序) として、以下の 2 種類を提案する。どちらも高域成分内の低域側のサブバンドからサンプルビットを順次走査していく。図 5(a) に示す方法 A では、サブバンド番号を重視し、遮断周波数  $P$  に近い低域側サンプルの MSB から LSB までをまとめて 1 サンプルずつ走査する。一方、図 5(b) に示す方法 B では、高域に存在するサンプルの MSB を重視する。図に示すように、サブバンド  $P, P+1, P+2, \dots$  の順に最終サブバンドまでまず MSB を走査し、重要度の高いビットから同様に走査する。

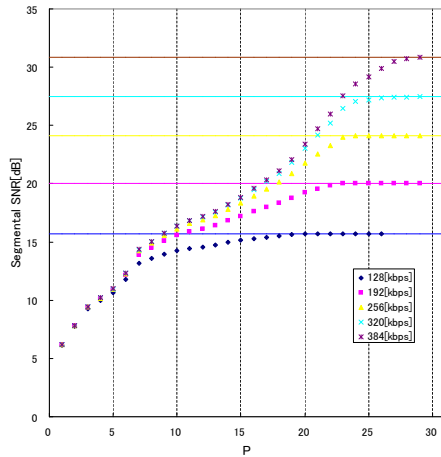
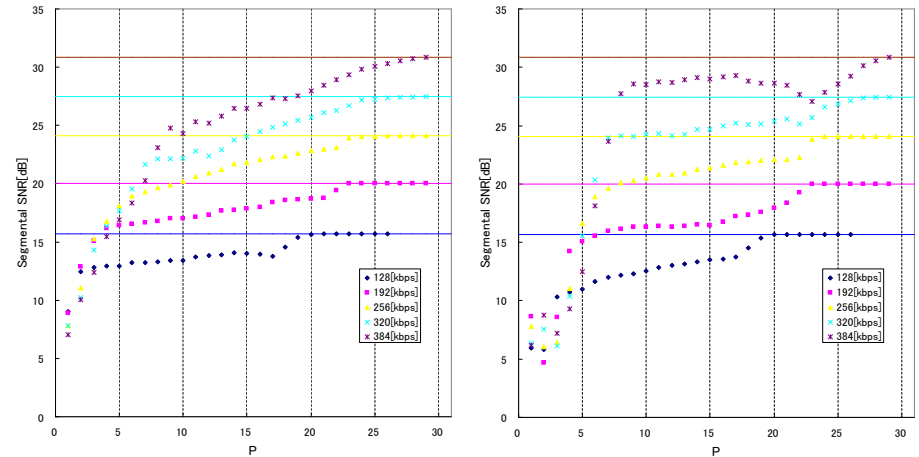


図 8 オリジナル音声  $G$  に対する帯域制限音声  $G_{rate}^P$  の Segmental SNR [dB]  
Fig. 8 Segmental SNR [dB] of band-passed audio  $G_{rate}^P$  for original audio  $G$

### 5. 再符号化した圧縮符号の冗長成分の取り扱い

予備実験として、再符号化した圧縮符号  $B_{rate}^{PP}$  の冗長成分に乱数を埋め込み ( $\tilde{B}_{rate}^{PP}$  を生成)、これを再々符号化した圧縮符号  $\tilde{\tilde{B}}_{rate}^{PPP}$  を生成する試行を 100 回行った。このとき、まず  $B_{rate}^P$  と  $\tilde{\tilde{B}}_{rate}^{PPP}$  の割り当てビット数の変化を調査した結果を表 1 に示す。ただし、楽曲として“Pops”を利用し、 $P = 9$  である。これより、誤差は 0 近傍の狭い範囲に収まっており、 $B_{rate}^P$  の割り当てビット数は、再符号化後の冗長成分への情報の埋め込みと再々符号化によって大きく変化していないことがわかる。これをさらに詳しく調べるために、384[kbps] の第 1000 フレームについて、 $B_{rate}^P$  と  $\tilde{\tilde{B}}_{rate}^{PPP}$  のそれぞれの割り当てビット数を比較し、100 回の試行中どのビット番号で何個誤りが生じたか、その頻度を計数した結果を図 6 に示す。ただし、表中の“-”は頻度無しを示し、太線より上の LSB 側が埋め込みに利用する冗長成分であり、太線から下の MSB 側が元の  $B_{rate}^P$  の符号領域となっている。従って、この  $B_{rate}^P$  の符号領域 ( $B_{rate}^P$  の割り当てビット数) が、 $\tilde{\tilde{B}}_{rate}^{PPP}$  から埋め込んだ情報を抽出するための境界線となる。この結果から、 $B_{rate}^P$  の符号領域内の LSB (第 1 サブバンド、ビット番号 9) で 3 回、冗長成分内の MSB (第 8 サブバンド、ビット番号 7) で 3 回それぞれ誤りが生じている。このような誤りは境界線の前後で発生する恐れがあり、埋め込んだ高域成分を抽出す



(a)  $G$  に対する  $\tilde{G}_{rate}^{all}$  (方法 A) (b)  $G$  に対する  $\tilde{G}_{rate}^{all}$  (方法 B)

図 9 オリジナル音声  $G$  に対する再構成音声  $\tilde{G}_{rate}^{all}$  の Segmental SNR [dB]  
Fig. 9 Segmental SNR of reconstructed audio  $\tilde{G}_{rate}^{all}$  for original audio  $G$

る際に誤りを招く。そこで本稿では、再符号化圧縮符号  $B_{rate}^{PP}$  に高域成分を挿入するとき、図 7 のように冗長成分を走査する。すなわち、境界線領域から遠い  $B_{rate}^P$  のサンプルビットの LSB のビット層からサブバンド番号の小さい順 (0, 1, 2, ...) に走査してゆき、現在のビット層の走査が終了したら、(LSB の) 次のビット層を走査する。このような順序で走査を続けてゆき、最終的に境界線を超えない位置まで走査する。提案法では、高域成分の埋め込みには  $B_{rate}^P$  の割り当てビット数を境界線として利用し、高域成分の抽出には  $\tilde{\tilde{B}}_{rate}^{PPP}$  の割り当てビット数を境界線として利用する。

### 6. 実験結果と考察

実験には、RWC 研究用音楽データベース (音楽ジャンル)<sup>9)</sup> から抜粋した“Pops”(No.1), “Jazz”(No.30), “Classic”(No.53) の 3 曲を利用し、必要な修正<sup>7),8)</sup> を加えた MPEG-1 オーディオレイヤー II 符号化器を用いた。帯域制限では、サブバンド番号 (遮断周波数)  $P$  を  $1 \leq P \leq 26$  (112~160[kbps] の場合) に変化させた。以降、紙面スペースの都合上、“Pops”の結果のみ掲載する。

まず、オリジナル音声  $G$  に対する帯域制限した音声  $G_{rate}^P$  の Segmental SNR<sup>10),11)</sup> の計

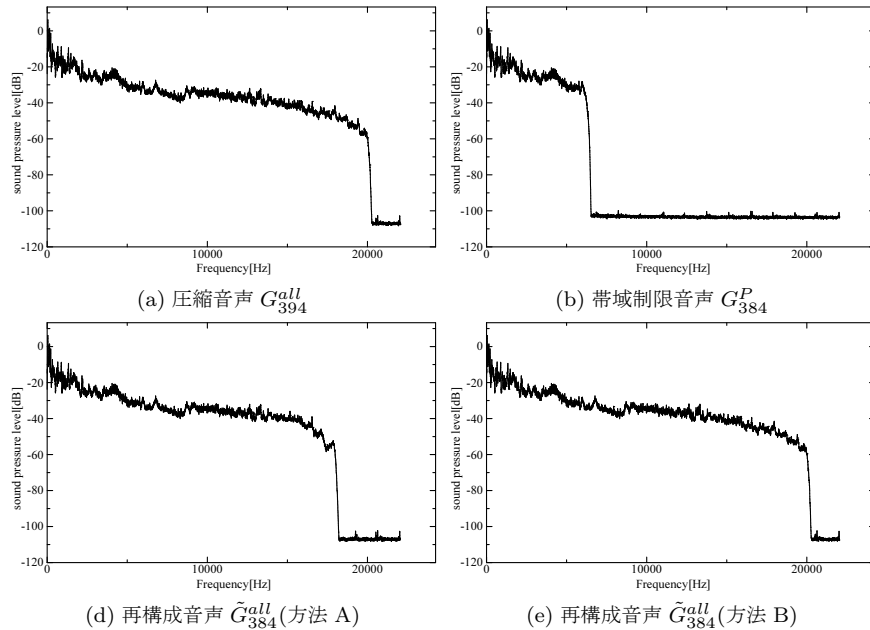


図 10 スペクトル波形の比較

Fig. 10 Comparison of frequency spectrum

測結果を図 8 に示す。遮断周波数  $P$  を減少することによって SNR 値が下がり、効果的に音質劣化が実現されていることがわかる。次に、この  $G_{rate}^P$  を再符号化して高域成分を埋め込んだ  $\tilde{G}_{rate}^{PP}$  を、さらに再々符号化を介して再構成した  $\tilde{G}_{rate}^{all}$  のオリジナル音声  $G$  に対する Segmental SNR を図 9 に示す。これより、埋め込んだ高域成分を利用した音声の再構成によって、帯域制限によって劣化した音声の SNR 値が大幅に向上することがわかる。高域サンプルビットの挿入方法の違いを比較すると、全般に MSB を重視する方法 B の方が高い SNR を示した。これは、高域に存在する全サンプルの重要度の高いビットが音声の再構成時に用いられたことによると思われる。

次に、元の圧縮符号化音声  $G_{rate}^{all}$ 、帯域制限音声  $G_{rate}^P$ 、高域成分の埋め込まれた再符号化音声  $\tilde{G}_{rate}^{PP}$ 、再構成音声  $\tilde{G}_{rate}^{all}$  のスペクトル波形を図 10 で比較する。音声の再構成により、帯域制限した音声の高域成分がかなり復元されていることがわかる。しかし、元の圧縮音声のスペクトルと比較すると、方法 B は  $G_{rate}^{all}$  に近い周波数まで復元されているのに対

し、方法 A では約 18[kHz] までしか復元されていない。このことが、方法 B と比べて全般に音質が低かったことの原因と思われる。

また、方法 B で高い符号化レートのとき遮断周波数  $P = 23$  前後で生じているギャップは、“Pops” では第 20 サブバンド以降の高域に多くのサンプルビットが存在するにもかかわらず、 $P$  以降のサンプル値の MSB に近い上位ビットしか格納されず、高域の再現性が低下したために生じたと思われる。

## 7. まとめ

本稿では、MPEG オーディオ信号に対する、データハイディングを用いる準可逆なローパスフィルタの実現方法を提案した。MPEG-1 オーディオレイヤー II 符号化による圧縮音声を用いた検討を用い、帯域制限した音声の再符号化によって生じる冗長成分を高域成分を埋め込み、これを再々符号化によって抽出し、復元された低域成分と抽出した高域成分を合成して元のオーディオ信号を再構成した。シミュレーション実験により、帯域制限により劣化した音声再構成によって元の圧縮音声に近い音質を復元することができた。

今後は、再構成した音声の品質をさらに改善するための検討が必要である。

## 参考文献

- 1) 藤原 洋, 最新 MPEG 教科書, 株式会社アスキー, 2001.
- 2) 藤原 洋, 画像&音声圧縮技術のすべて, CQ 出版, 2000.
- 3) 安田 浩, 藤原 洋, デジタル放送インターネットのための情報圧縮技術, 共立出版, 1999.
- 4) 米満潤, MPEG 技術, トリケップス, 1997.
- 5) 岡本龍明, 山本博資, 現代暗号, 産業図書, 1997.
- 6) 松井甲子雄, 電子透かしの基礎, 森北出版, 1998.
- 7) 萩野谷 俊典, 田中 清, 高木 幸一: “帯域制限した MPEG オーディオに対する再符号化・再々符号化が符号化データと再生音声に与える影響の考察”, 情報処理学会 オーディオビジュアル複合情報処理研究会, to appear, March 2010.
- 8) 萩野谷 俊典, 田中 清, 高木 幸一: “MPEG-1 オーディオレイヤー II による高ビットレート符号化音声の品質向上”, ITE Technical Report, vol.33, no.54, pp.5-8, ME2009-215, Dec. 2009.
- 9) 後藤 真孝, 橋口 博樹, 西村 拓一, 岡 隆一: “RWC 研究用音楽データベース: 音楽ジャンルデータベースと楽器音データベース”, 情報処理学会 音楽情報科学研究会, 2002-MUS-45-4, Vol.2002, No.40, pp.19-26, May 2002.
- 10) 守谷 健弘, Speech Coding, 電子情報通信学会, 1998.
- 11) K. Takagi, S. Sakazawa, Y. Takishima: “Light Weight MP3 Watermarking Method for Mobile Terminals”, IEICE Trans. Fundamentals, vol.E91-A, no.9, pp.2546-2554, 2008.