

卒業論文

-ハフマンエンコーディングを使ったMP3の高速処理化-

東海大学電子情報学部コミュニケーション工学科

1 ADT 2 3 0 1 杉本麻衣子

平成 17 年 1 月 31 日

目次

1	はじめに	3
1.1	論文構成	3
1.2	研究目的	3
2	MP 3の概要	3
2.1	構成	3
2.1.1	符号器	3
2.1.2	復号器	4
2.1.3	MP 3のエンコード	4
2.1.4	MP 3のデコード	6
3	ハフマンコード	9
4	ハフマンエンコーディング	9
4.1	符号化処理	9
4.1.1	原理	9
4.2	ハフマンコード化処理	11
4.2.1	レイヤ1,レイヤ2,レイヤ3の共通点	11
4.2.2	レイヤ1,レイヤ2,レイヤ3の異なる点	11
4.2.3	それぞれのレイヤでの符号化処理	11
4.2.4	ハフマン符号のアルゴリズム	14
4.3	知覚符号化	16
4.3.1	知覚符号化	16
4.3.2	聴覚心理符号化	17
5	レイヤ3	18
6	ハフマンエンコーディングの改良	18
6.1	ハフマンエンコードの記述	22
6.2	MP 3のデータ構造	22
6.2.1	サイド情報	22
6.2.2	エラーチェック	22
6.2.3	メインデータ処理	22
6.2.4	スケールファクタ	22
6.2.5	ハフマンコードビット列	26
6.3	データ構造からみたハフマンコード	26
6.3.1	Big Values	26
6.3.2	Big Value領域のハフマン符号化	29
6.3.3	Count 1領域のハフマン符号化	29
6.4	問題点	29
6.5	改良	29

7	今後のMP3の動向3	35
7.1	ソフトウェア	35
7.1.1	圧縮符号方式	35
7.1.2	AAC	35
7.2	MPEG4	37
7.3	ハードウェア	37
7.3.1	LSI	37
8	まとめ	37
9	感想	38
10	今後の方針	38
11	謝辞	38
12	参考文献	38

1 はじめに

社会的なユビキタス構想により、日々のIT革命は進んでいる。特に今日のインターネット社会は新たな形へと進化している。特にブロードバンドの急速な普及により以前ではテキストのみだけであったWEBページもFLASHやストリーミングといった俗に言われる「コンテンツ配信」の需要が増えつつある。

その中の一つとして注目されているのが「インターネットミュージック」で通称「音楽配信」と呼ばれている。最近では、大手企業であるアップルコンピュータ株式会社が大容量で最小ポータブルプレイヤーを発表した。

このように最近の音楽シーンではCDよりも音楽配信の需要が増えつつある。インターネットミュージックの中で一番広く使用されているのがMP3である。

MP3は利便性が非常に良く、素人でも簡単に作製することができる。また、音質も設定によってはCDと同等の音質を実現することができる。

私は、今後の音楽シーン、いや「音」の分野で大いに発展していくのではないかと思い、注目をし本研究にした。

1.1 論文構成

今回は大きく分けて4つの構成になっている。一つ目に今、述べている初めの部分。そして2つ目にMP3についての概要、3つ目に研究、4つ目に今後のMP3の動向について述べてある。

1.2 研究目的

今日までに開発されているMP3プレイヤーのほとんどが、デコードだけのものが多く、エンコードに関しては一例としてパソコン等からソフトウェア上で行われている。

このエンコード部分をMP3プレイヤーの内部に埋めこめられないか、つまりMP3のハードウェアに組み込めることはできないかというところから当初は始まった。

それから研究室の先輩方々、研究を重ね、MP3のエンコードの量子化や浮動少数点積和演算器の設計を行い、研究の焦点がエンコード部分をハードウェア上に実現するところから演算処理の速さに移ってきた。はじめにでも述べたように今日のインターネット世界ではストリーミング等が増えている。そうなる問題になるのが容量である。現状の問題としてMP3のエンコード時間は、音質を良くするほど時間がかかる傾向がある。また音質を良くすると容量がかなりいる。私は、その部分に注目をした。

また様々な研究機関が量子化やMDCTの高速化の研究が行われ、比較的に限界がみえてきているように感じている。

そこで今回、私が注目したのはハフマン符号化の部分である。他の部分に比べ未知なるものがたくさんあるように私は感じた。そこでハフマン符号を高速化し、ハードウェア上で実現が出来ないかを思い、私の研究を始めた。

2 MP3の概要

MP3とはMPEG Audio Layer の略である。MPEGの正式名称はMoving Picture Experts Groupと呼ばれISO / IEC JTC 1 / SC 29 / WG 11の通称である。そのためMP3をISO / IEC 11172-3や「MPEG-1オーディオ3」と呼ばれることがある。

2.1 構成

MP3には大きく「符号器」と「復号器」分けられることが出来る。

2.1.1 符号器

図1のように符号器は、デジタル音響信号を処理し、記録を目的として圧縮したビット流を生成する。符号器のアルゴリズムは標準化せず、聴覚のマスキングしきい値、量子化及び倍率変換のように、符号化するために様々な手段を使ってよ

い。しかし、符号器の出力は、復号器を想定した応用にふさわしい音響信号を生成するものにし、データは符号器の実現の仕方によって異なる。

*** 写像**

入力音響ビット流をフィルタに処理し、更に間引いた表現を生成する。また、写像した標本を「帯域分割標本」または「変換した帯域分割標本」と呼ぶ。

*** 聴覚心理モデル**

量子化及び符号化を制御するデータの組を生成する。

*** 量子化及び符号化**

写像した入力標本から符号化記号の組を生成し、符号化システムに依存する。

*** フレーム組立**

量子化及び符号の出力データとアンシラリデータから実際のビット流を組み立てて、誤り訂正などの情報を必要に応じて付加する。

2.1.2 復号器

図2のように復号器は規定し、圧縮したビット流データを入力する。符号器において誤り検出を適用する場合、分解し復号するとビット流は、誤り検出を行う。

*** フレーム分解**

ビット流データを分離させる。

*** 復元**

一連の写像標本の量子化形を復元する

*** 逆写像**

写真標本を PCM に戻す。

2.1.3 MP 3 のエンコード

MP 3 のエンコード方法は仕様として定めていない。最低限、デコードが可能であればよいことになっている。そのため、近年の研究では、エンコード部分の処理高速化についての研究が多くされている。主な高速処理の方法としては、主に CPU を一杯使用し、エンコードの速度を上げる。このため各アーキテクチャに特化されたアセンブラコードが用いられている。この方法は MP 3 のアルゴリズムを変更し、処理そのものの計算量を減らしている。よって、十分に高速化が行われるなら低速な処理システムにおいても MP 3 は十分に使用することが出来る。図3に一般的なエンコードプロセスの流れを示す。

先程の符号器を更に細かく書いたものである。

* Hybrid Filter Bank Analysis Subband Filter Bank, MDCT, Alias Reduction の3つの処理をあわせて Hybrid Filter Bank と呼ぶ。

- Analysis Subband Filter Bank

PCM データを 32 の周波数帯へ変換を行う。入力は 512 サンプルで 32 サンプルずつデータ入れ換えを行い、1 単位時間、32 サブバンドの 32 サンプルを出力する。

- MDCT

MDCT は Modified Discrete Transform といい離散コサイン変換 tp 窓関数処理を行う部分である。離散コサイン変換では、時間領域から周波数領域へと変換が行われ、窓関数処理では心理聴覚モデル分析で判定されたブロックタイプに対応した窓関数との乗算が行われる。処理は、Antialias Subband Filter Bank で得られた 32 サンプルを 1 単位として行われる。

- Alias Reduction

ダウンサンプリングを行うと折り返し歪みが発生してしまう。Alias Reduction ではバタフライ演算を用いて折り返し歪みの削除を行う。

* Psycho-acoustic Modeling

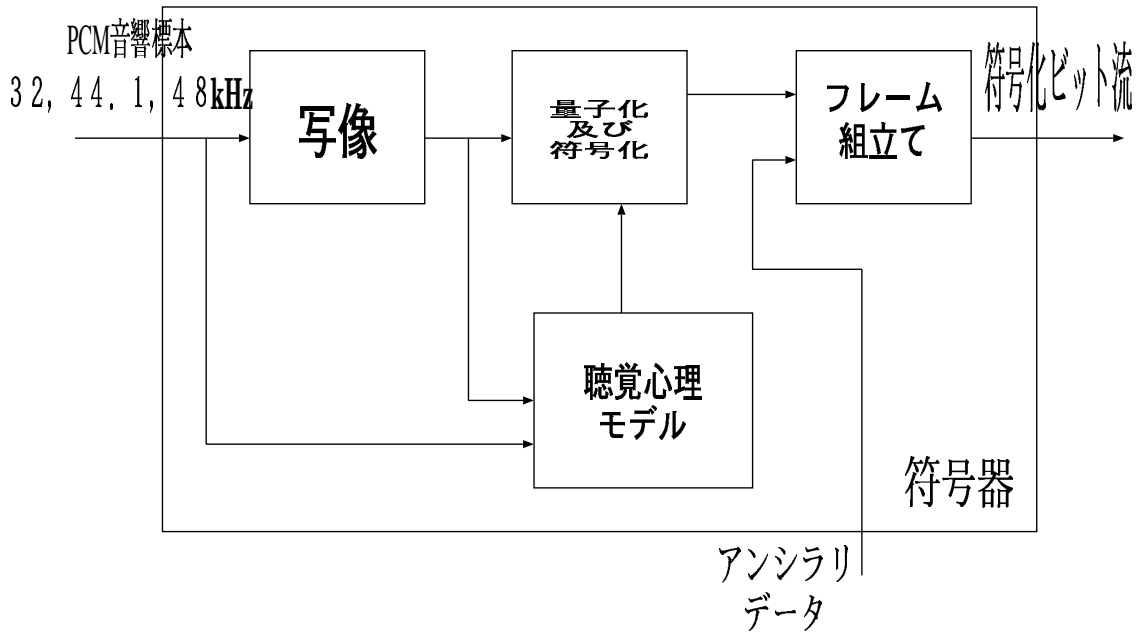


図 1: 符号器の基本構造

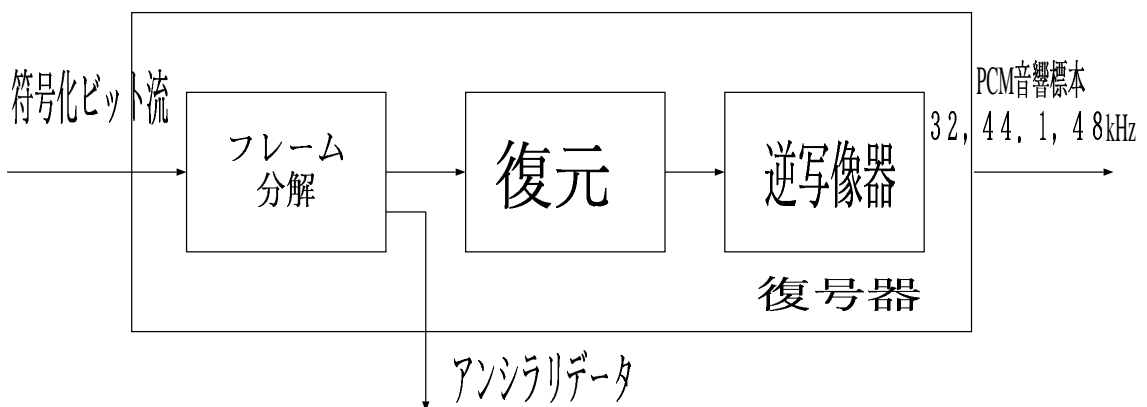


図 2: 復号器の基本構造

心理聴覚モデルと呼ばれ、人間の聴覚の特性を基に、音質の劣化を最小限に抑えつつ圧縮を行うために必要なデータとそうでないものとの判別を行う。分析の結果は MDCT やスケールファクタの算出に用いられる。

* Iteration Loop

量子化、エンコードを行った結果が規定のビット数を有効に使用するため、スケールファクタ算出と非線形量子化を繰り返し、「最適なスケールファクタを検索する。

-Scale Factor Calculation

心理聴覚モデルで分析された情報を基にスケールファクタの算出を行う。算出されたスケールファクタは非線形量子化へと送られる。

-Quantization

スケールファクタを基に非線形量子化を行う。量子化、エンコードを行った結果のビット数を解析し、最適でない場合はスケールファクタの算出に戻る。

* Huffman Encodeing

量子化されたデータに対しハフマンコードを行う。

* Side Information Encoding

サイド情報としてデコード時に必要となるスケールファクタを始めとしたデータ属性を表す各種情報の生成を行う。

* Bistream Formatting

ヘッダ、CRC、サイド情報。メインデータ、付加情報を MP 3 のフォーマットに従ったビットストリームとなるよう仕上げを行う。

2.1.4 MP 3 のデコード

デコードプロセスは MP 3 の仕様として定められている。プロセスの流れはエンコードと対照的

なものとなる。図 4 に一般的なデコードプロセスの流れを示す。先程の復号器を更に細かく書いたものである。

* Bitstream Analysis

MP 3 フォーマットのビットストリームをヘッダ、CRC、サイド情報、メインデータ、付加情報に分解する。ビットストリームより取り出されたこれらの情報を基にデコードが行われる。

* Huffman Decodeing

メインデータのハフマンデコードを行う。ハフマン符号は非線形量子化されたデータへデコードされる。

* Side Information Decoding

スケールファクタを始めとしたフレーム内の情報を取り出す。

* Inverse Quantization

サイド情報を基にデコードされたデータの逆量子化を行う。

* Alias Reduction

折り返し歪みを削減する。エンコード時とは正負を反転したバタフライ演算を行う。

* IMDCT

Inverse Modified Discrete Cosine Transform といい、MDCT の逆の処理に相当する。1 グラニュールを処理単位とし、周波数領域から時間領域への変換と窓関数処理を行う。

* Synthesis Subband Filyer Bank

3 2 に分解されたサブバンドを合成し、PCM データへと復元が行われ、デコードプロセスが完了する。

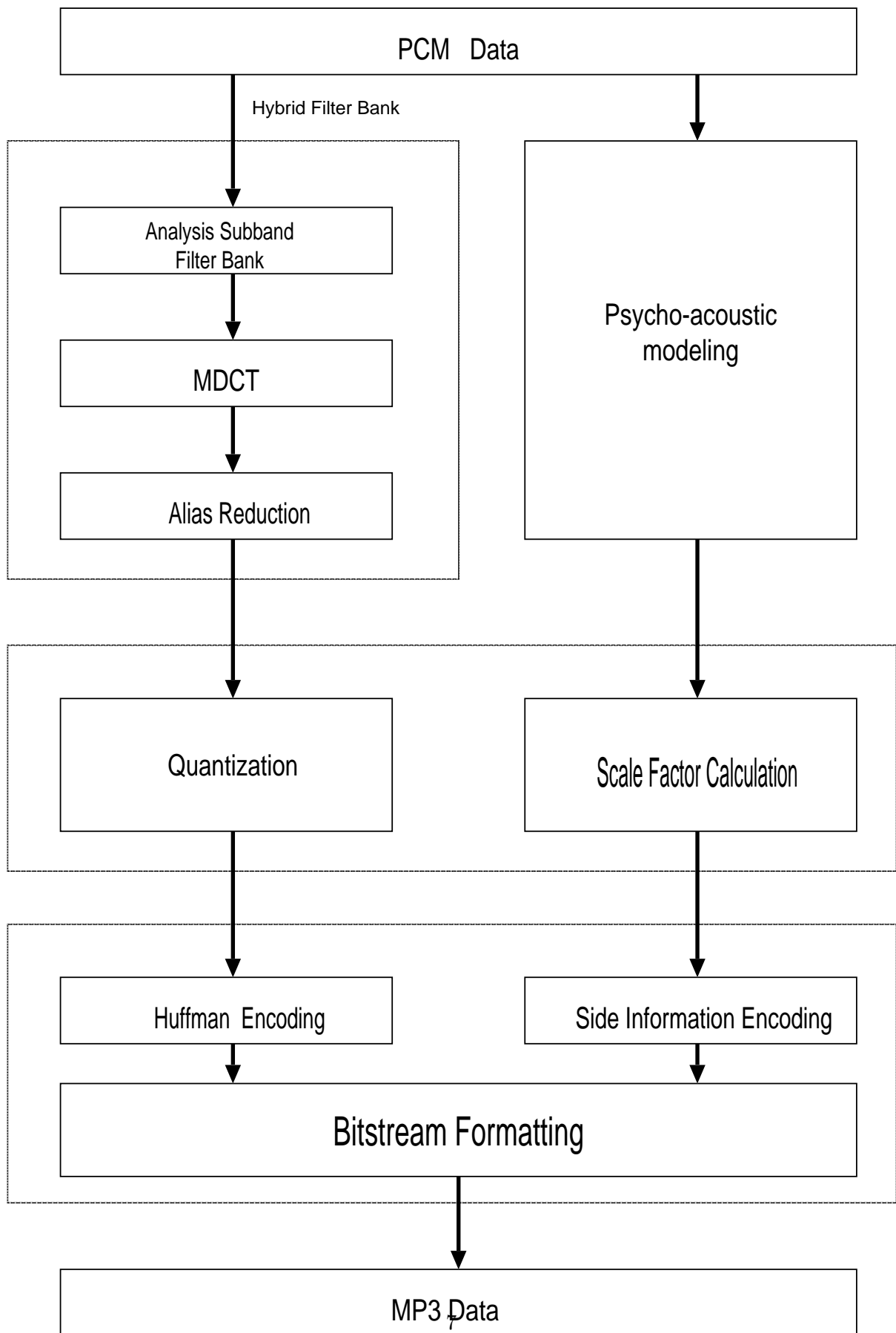


図 3: エンコードプロセス

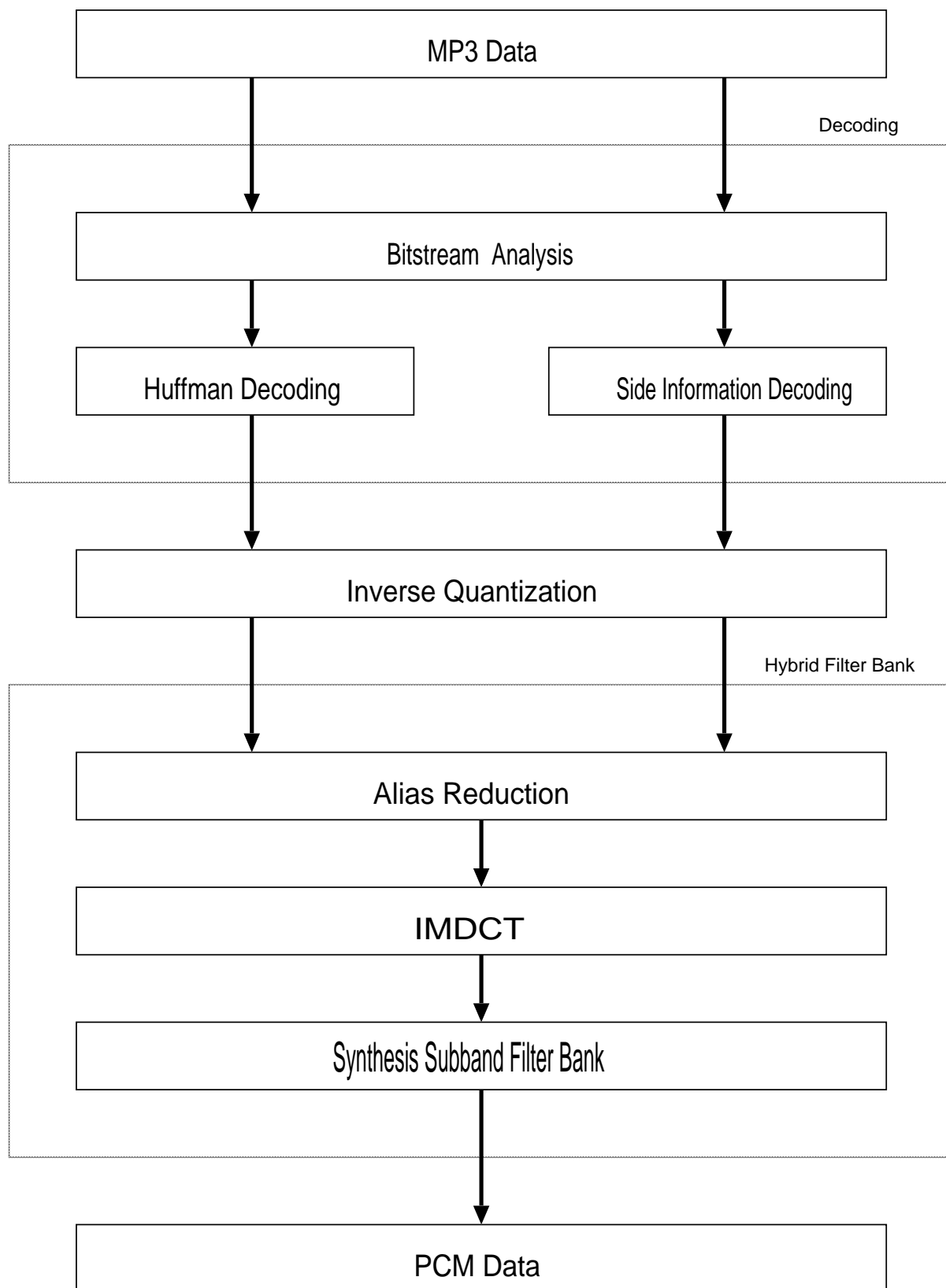


図 4: デコードプロセス

3 ハフマンコード

ハフマンコードは、出現頻度の高いデータ(コード)の表現語長をより短くすることで効率よくデータを表現することができる符号である。文章などのコードの出現頻度に落差が激しいものなどに特に有効である。ただし、あらかじめ出現頻度を知っておく必要がある。

それでは、一般的なハフマンコードについて示すハフマンコードはハフマン木という二分木によって表現することができる。例えば図5に示したようにハフマン木の下端、すなわち葉の部分に2, 3, 5, 7とあるが、これは扱うデータの個数が4つ、またそれぞれの出現頻度の比を示している。これらの親の節では、二つの子の出現頻度の和になるようになっており、木の深さが浅いほど出現確率の高いデータが位置するように作られている。このように表されるハフマン木は、根に近いほど出現頻度が高くなる。左の子を0、右の子を1と符号づければ、表3.1のハフマンコード表のようになる。この表よりハフマン木より導出されたハフマンコードは根に近いほど(語長が短いほど)出現頻度が高く、また、すべて瞬時符号であることがわかる。

4 ハフマンエンコーディング

今回の研究においてのメインとなる部分である。そのため、まずは、データを理解するところから始めることにした
まずは、一般的な概要のみを述べていく。

4.1 符号化処理

4.1.1 原理

ISO / IEC 11172-3 (MPEG - Audio) アルゴリズムは、聴覚心理アルゴリズムを用いている。聴覚心理アルゴリズムの主要部分を図6に示す。

* フィルタバンク

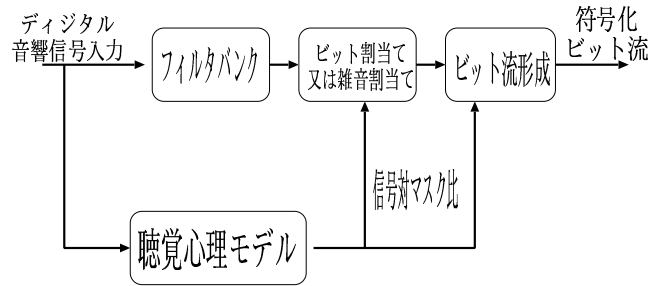


図 6: ISO/IEC 1172-3 符号器構成図

多重位相フィルタバンク及び混成多重位相フィルタバンク (MDCT フィルタバンク) が使用されており、各々時間及び周波数上での特殊な写像を行う。フィルタバンクから出力した標本を量子化する。

* 聴覚心理モデル

二つの聴覚心理モデルがあり、モデル1はレイヤ1及び2にモデル2はレイヤ3に対して実績がある。最終出力は、各々の分割帯域(レイヤ1及びレイヤ2の場合)又は分割帯域のグループ(レイヤ3の場合)に対する信号対マスク(SMR)とする。

* ビット割当て又は雑音割当て

割当て器はフィルタバンクの出力標本及びSMRを考慮しながら、ビット速度及びマスキングの両方の要求を満たすようにビット割当て(レイヤ1及びレイヤ2の場合)又は雑音割当て(レイヤ3の場合)を調整する。

* ビット流形成

量子化されたフィルタバンクの出力、ビット割当て(レイヤ1及びレイヤ2の場合)又は雑音割当て(レイヤ3の場合)及びほかの必要な付加情報の符号化及びフォーマットを行う。レイヤ3の場合はハフマン符号化も行う。

* フィルタバンク

レイヤ1及び2では32個の分割帯域からなるフィルタバンクを用いる。分割帯域中の標本は12個又は36個ずつ一括して処理する。レイヤ3

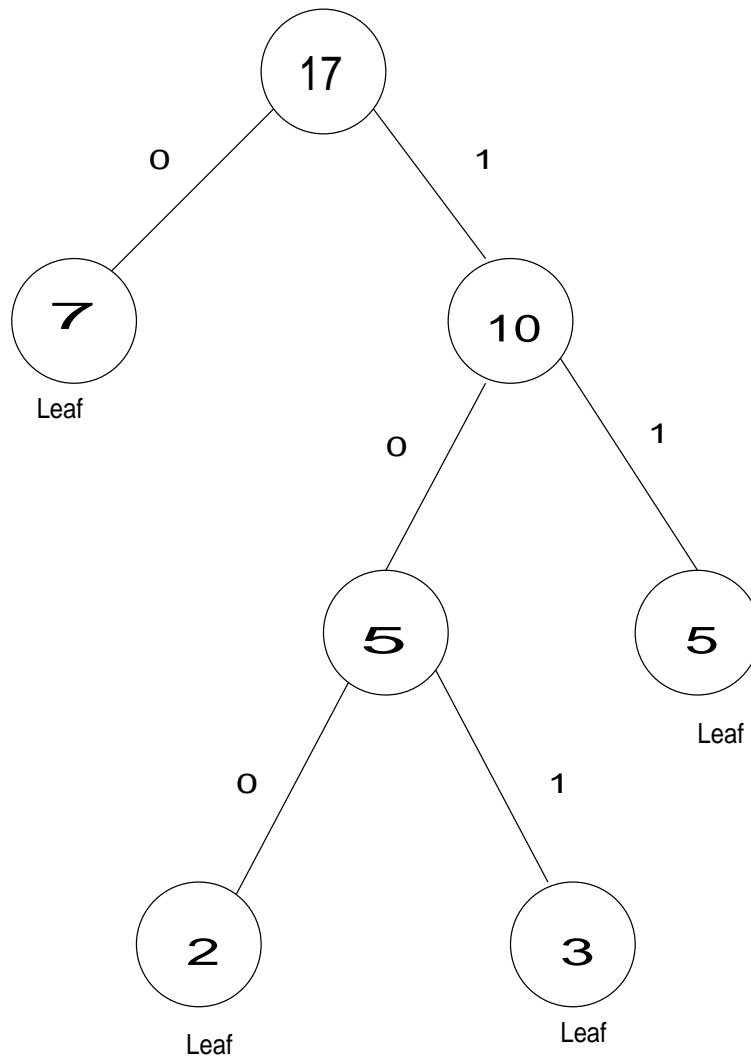


図 5: ハフマンの木一例

表 1: ハフマン木から読み取れること

出現頻度の比	ハフマン符号	語長
2	100	3
3	101	3
5	11	2
7	0	1

ではフィルタバンクは入力信号に依存した、 6×32 個又は 18×32 個の周波数帯域の解像度をもつ。

*ビット割当て又は雑音割当ての方法
ビット速度制御は、レイヤ1及びレイヤ2では、ビット割当て処理として、レイヤ3では雑音割当てループとして実現する。いずれの場合も結果を量子化パラメタ及び量子化された標本として、ビット流形成に伝達する。

*ビット流形成
レイヤ1及びレイヤ2では、分割帯域の各々の標本に対して固定長のPCM符号を用いる。ただし、レイヤ2では量子化された標本がグループ化されていることある。レイヤ3ではハフマン符号を用いる。

*入力高域通過フィルタ
符号器の入力に遮断周波数 $2 \sim 10$ Hzの高域通過フィルタを含んでいることが望ましい。これによって、最も低域の分割帯域に対して必要以上に多くビットを割り当てることを防ぎ、結果的に全体の音質を向上させる。

*分析帯域分割フィルタ
分割帯域分割フィルタは、標本化周波数 f_s の広帯域信号を、標本化周波数 $f_s / 32$ の等間隔の分割帯域 32 個に分割するために用いる。

4.2 ハフマンコード化処理

MP3の元であるMPEG-1オーディオには、レイヤ1、レイヤ2、レイヤ3という3つが含まれている。MP3はレイヤ3を使用するが3つの方式の相違点についての記述について述べたいと思う。

4.2.1 レイヤ1、レイヤ2、レイヤ3の共通点

* $384 \sim 1152$ サンプルを一つのブロックとして処理している。

*入力信号を 32 の周波数帯域ごとに分ける働きをするフィルタによって周波数帯域を 32 等分出来る。

*周波数帯域ごとにスケールファクタ(信号の強さを表す指数)とビット割当て情報によって可変精度でサンプルを符号化する。

4.2.2 レイヤ1、レイヤ2、レイヤ3の異なる点

*レイヤ1、レイヤ2、レイヤ3はそれぞれ 384 サンプル、 1152 サンプル、 768 サンプルを1ブロックとして符号化する。

*レイヤ1は 32 のサブバンドに分割した信号をそれぞれ 12 サンプルずつ1つにまとめ処理する。(1ブロックの長さは 32×12 サンプルで 384 サンプル)

*レイヤ2は、1バンドにつき 36 サンプルを1ブロックとして扱う。(1ブロックは $32 \times 36 = 1152$ サンプル)

*レイヤ3はMDCTを使い、各バンドを 12 サンプルずつ変換し、さらにそれを2つ1組に処理する。(1ブロックは $32 \times 12 \times 2 = 768$ サンプル)

4.2.3 それぞれのレイヤでの符号化処理

基本的にレイヤは周波数分割しブロックにまとめた後の処理方式は同じです。しかし、レイヤ1、2、3の順により高度な処理が増えます。それにより、符号化効率が $1 \sim 2$ 割程度向上します。レイヤ1、2の符号処理は図7にも示したがポリフェーズフィルタバンクで入力信号を周波数後とに 32 の周波数帯域に分割する。そして、聴覚心理モデルでそれぞれの周波数帯域に最適な精度を求める。もし、一律の精度で符号化すれば、レイヤ1、2符号化とPCMの符号化の音質はほぼ同

表 2: レイヤ 1、レイヤ 2、レイヤ 3 の比較

	レイヤ 1
符号化データ構造	ブロックごとに符号化
フィルタバンク	3 2 バンドのポリフェーズフィルタバンク
サンプル化	スケールファクタとビット割当てをした後、可変精度で符号化
1 ブロックの長さ	3 8 4 サンプル
1 ブロックの中身	1 セグメント
その中身	3 2 サブバンド × 1 2 サンプ
	レイヤ 2
符号化データ構造	ブロックごとに符号化
フィルタバンク	3 2 バンドのポリフェーズフィルタバンク
サンプル化	スケールファクタとビット割当てをした後、可変精度で符号化
1 ブロックの長さ	1 1 5 2 サンプル
1 ブロックの中身	3 セグメント
その中身	3 2 サブバンド × 1 2 サンプル
	レイヤ 3
符号化データ構造	ブロックごとに符号化
フィルタバンク	3 2 バンドのポリフェーズフィルタバンク
サンプル化	スケールファクタとビット割当てをした後、可変精度で符号化
1 ブロックの長さ	7 6 8 サンプル
1 ブロックの中身	2 グラニュール
その中身	3 2 サブバンド × 1 2 個の MDCT

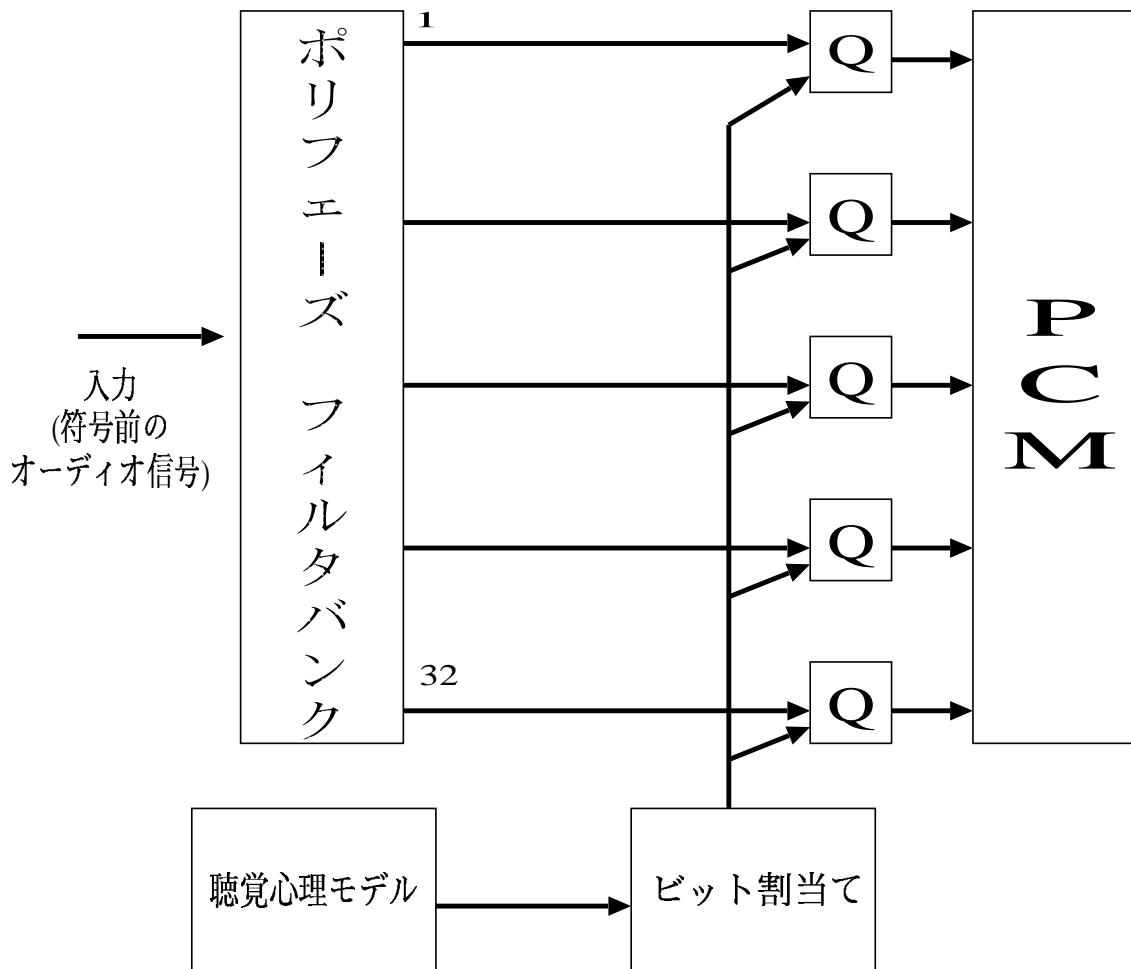


図 7: レイヤ1、レイヤ2の符号化処理

じである。更に各周波数ごとに最適の精度で符号化する。大部分の周波数帯域では CD よりも低い精度で符号化する。このようにレイヤ 1、2 ではビット割当てによって周波数後とに符号化精度を最適なものにし、仮にビット割当てを行わなければ PCM 符号化と精度も符号化効率も変わらない。これが、レイヤ 1、レイヤ 2 の処理技術である。次にレイヤ 3 の符号化処理である。レイヤ 3 の符号化処理は、図 8 にも示したが、ポリフェーズフィルタバンクで周波数帯域ごとに分割した信号を MDCT 信号処理でさらに 1/2 分割にする。MDCT によって細かく分割された信号を最適な精度で量子化する。そして、量子化した値をハフマン符号で圧縮符号化する。ハフマン符号化を使うと 0 と 1 の列に置き換えるよりも 1.5 倍程度効率よく圧縮符号化することが出来る。今回の研究では、このレイヤ 3 の部分が最も重要となっている。

4.2.4 ハフマン符号のアルゴリズム

MDCT 方式のハフマン符号化は、64 個の DCT 係数に対して DC 成分と AC 成分で行う。DC 成分の符号化では隣接する直前のブロックの DC 成分との差がハフマン符号化される。AC 成分に対しては、量子化によって大部分の係数値が 0 となることから、非零の AC 成分とその AC 成分の前に位置する零値の AC 成分の個数を組合せ、零ラングレスを考慮したハフマン符号化が行われる。まずはジグザグスキャンによる並び替えからみていこうと思う。2次元の DCT 変換係数をハフマン符号化するために、最初、2次元配列から 1次元配列への変換を行う(このことを「ジグザグスキャン」という)。8×8の DCT 変換係数からなる 2次元信号を $Sq(v,u=0,1,\dots,7)$ とし、これらをさらにジグザグスキャンをして 1次元信号に変換した信号を $ZZ(k)(k = 0,1,\dots,63)$ とする。 $ZZ(0)$ が DC 成分、 $ZZ(1) \sim ZZ(63)$ が AC 成分である。DC 成分は、直前のブロックにおける DC 成分の量子化値との差分信号をハフマン符号化する。

AC 成分は $ZZ(1), ZZ(2), \dots, ZZ(63)$ の連続する零の個数と非零の AC 成分を組み合わせでハフマン符号化する。

DC 差分値の符号化を行うには、DC 差分値の値に対応するカテゴリ SSSS とハフマン符号を求める。ここで SSSS は、DC 差分値 w_p グループ分けするために変数であり、DC 差分値の正の場合は 2進を表現した場合に上位ビットからみて最初に「1」が立つ LSB の番号であり、DC 差分値が負の場合には -1 を 2進表現した場合に上位ビットからみて最初に「0」が立つ LSB の番号である。カテゴリ SSSS に対応するハフマン符号のみでは、グループは特手できても具体的なグループ内の DC 差分値が表現できないため、ハフマン符号の後に付加ビットを加えて DC 差分値の値を特定する。SSSS はこの付加ビットのビット長に対応しており、DC 差分値が正の場合には DC 差分値の LSB から SSSS ビットを DC 差分値が負の場合には DC 差分値 -1 の LSB から SSSS ビットが付加ビットとなる。これにより、復号器では付加ビットの先頭ビットが 1 の場合には DC 差分値が正、0 の場合には DC 差分値が負であると判断できる。このように DC 差分値のハフマン符号化は DC 差分値の大きさのカテゴリを最初にハフマン符号化は、DC 差分値の大きさのカテゴリを最初にハフマン符号を指定し、その後続く付加ビットで、カテゴリ内の DC 差分値の具体的な値を指定する形、(カテゴリのハフマン符号) + (付加ビット) の順番に結合された可変長符号を用いて行われる。

最後に AC 成分の符号化である。AC 成分の符号化は非零の AC 成分とその非零係数の前にある連続する零の AC 成分の個数を組み合わせで可変長符号を構成し、さらに DC 差分値の符号化と同様に付加ビットを加える、具体的な符号化手順は、最初に AC 成分を $ZZ(1)$ から順番にみていき、非零の AC 成分があれば、それ以前の零値の AC 成分の個数を 4ビットの RRRR とし、さらに現在の非零の AC 成分に対応するカテゴリを 4ビットの SSSS(SSSS の求め方は DC 差分値符号化の場合と同様)として表し、ハフマン符号を求める。零値

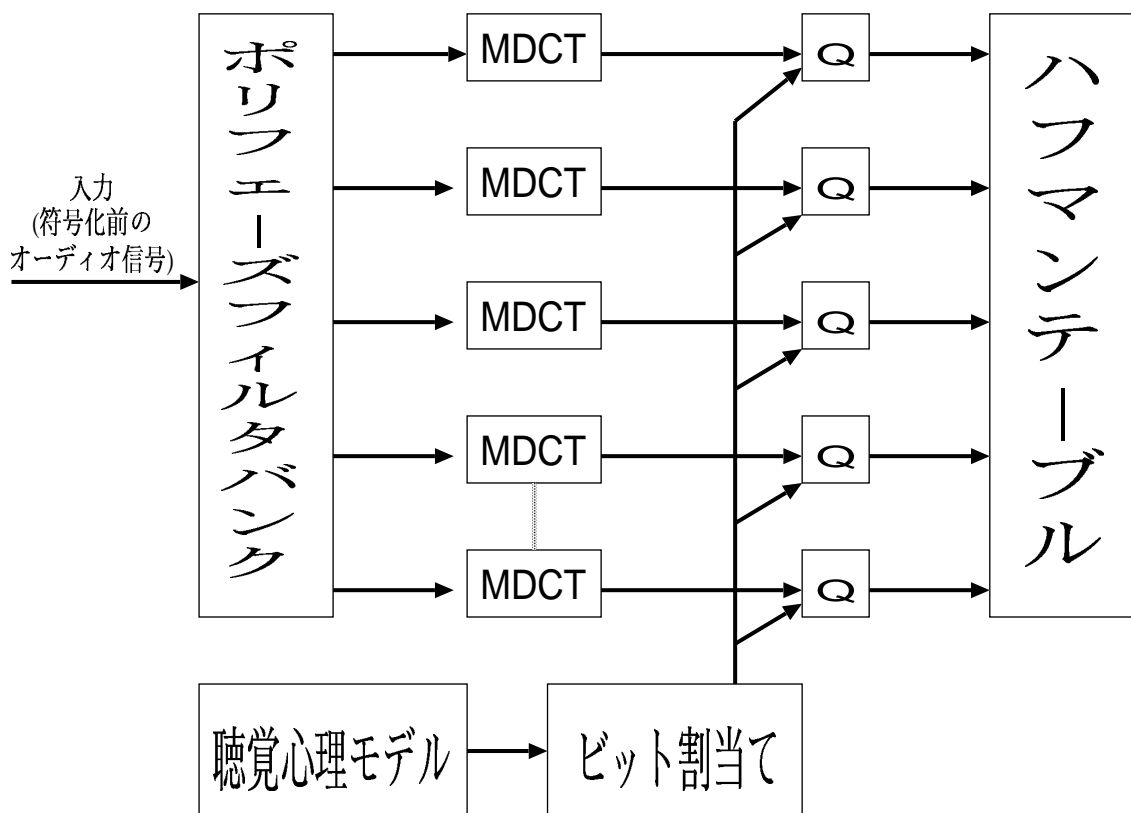


図 8: レイヤ 3 の符号化処理

のランレングスを加味した可変長符号化を行うことにより、連続する零値の個数を一括して一つの符号で表現できるため、高い符号化効率を得られる。AC成分の処理は以上のように二つの特徴がある。そのうちの一つは零値のAC成分が16個以上連続する場合である。すなわち、零ランレングスを表現するためのRRRRが4ビットであることにより16以上の零ランレングスは表現できない。そのため、連続する零値が15以下となるまで16個の零のグループに対して一つのRRRR / SSSS = 15 / 0の組み合わせに対応する符号(ZRL)を出力する。もう一方は、最後のAC成分ZZ(63)が0の場合であり、このときは、ブロックの最後に残った零値のAC成分に対して一括してRRRR / SSSS = 0 / 0の組み合わせに対応する符号(EOB)を出力した後、直ちに当該ブロックの符号化処理を終了する。なお、ブロック内の最後の係数ZZ(63)が悲零の場合にはEOBは用いられない。ZRLまたはEOB以外のハフマン符号の後には、具体的なAC成分の大きさを特定するための付加ビットを加える。すなわち、非零AC成分が負の場合にはAC - 1の値のLSBからSSSSビットを抽出し、これを付加ビットとする。DC差分値の場合と同様の復号器では付加ビットの先頭ビットが1の場合にAC成分は正、0の場合にはAC成分は負である判断する。

4.3 知覚符号化

今まで述べてきたことには、MPEGにとって最大の特徴が前提としてあるからである。そこで、この項目では、その前提にある部分を述べたいと思う。

最初に、オーディオに使われる信号として二つの方法がある。「エントロピー符号化」と「知覚符号化」である。「エントロピー符号化」は一つの信号の統計的な偏りを利用して、情報を失うことなく、より能率よく符号する方法である。画像圧縮でもしばしば使われていることがある。つまり、原信号と再生信号は全く同じであり情報の損失がない。もう一方の「知覚符号化」は信号を受け取る人間

の感覚の性質を利用し、感度の低い細部の情報を省略して符号量を削減していく方法である。そのため、原信号と再生信号が異なる。しかし、知覚には同じように聞こえている。通常のオーディオではこの二つの方法を組み合わせて使用されているが、MPEGでは、「知覚符号化」が採用されている。

4.3.1 知覚符号化

MPEGオーディオで知覚符号化をしている原理は「聴覚心理」と呼ばれている。聴覚心理の応用については、オーディオ装置の性能改善のために比較的早くから詳しく調べられていたMPEGオーディオに利用されている聴覚心理も基本性質は1970年代にすでに明らかにされていた性質ではある。今日の信号処理技術、エレクトロニクス技術の急速な進歩によって、現実のものになってきたといえる。MPEGオーディオでは主に静寂時の「最小可聴限界」と「マスキング特性」が利用されている。静寂時の最小可聴限界とは、聴覚が検知できる音の最小レベルのことで聴覚が静寂時に聞き取ることのできるノイズの関係する。図9のように静寂時の最小可聴限界は音の周波数に依存する。図中のAのように静寂時の最小可聴限界より上のレベルの音は聞き取ることが出来るがBのように静寂時の最小可聴限界より下の音は聞き取ることができない。

次にマスキング特性とは主に歪みや背景ノイズの検知限界と関係している。特定の音の検知限界は、同時に聞いている他の音によって大きく変化する。例えば、静寂な環境ではせせらぎの音を聞き取れませんが、嵐の中では聞き取れない。これをマスキング効果と言う。マスクする音をマスキャー(Masker)といい、マスクされても聞こえなくなる音をマスキュー(Maskee)と呼ぶ。マスキング効果の様子は図10に示す。図は純音Aが存在している状態での他の純音の可聴限界を示したものだ。Aに対して一定の周波数範囲内では、他の純音はかなり聞き取りにくくなります。例えば、Bはかなりレベルの高い純音ですが、聞き取れず、

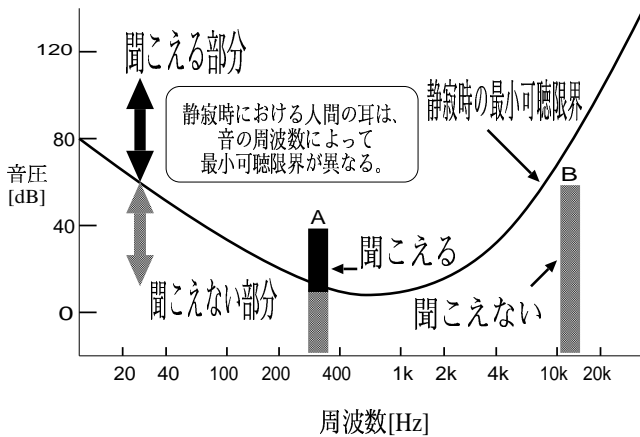


図 9: 知覚が検知できる音の最小レベルと周波数

Cは聞き取ることになります。マスキング効果はマスキャーとマスキューの周波数が近いほど強くなります。マスキング効果の及ぶ周波数幅をクリティカルバンドと呼ぶ。クリティカルバンドは、マスキャーやマスキューの周波数によって大きく異なる。クリティカルバンド幅は、1 kHz以上の周波数では周波数にほぼ比例し、1 kHz以下の周波数では100 Hz幅でほぼ一定である。マスキャーとマスキューが時間的に差がある場合でも時間が短ければマスキング効果が働く。この効果をテンポラルマスキング(継時マスキング)と呼ぶ。テンポラルマスキングはマイカが先行している場合の方が効果が大きくなる。

このような仕組みからなっている。では、この構造で、どうして耳には繋がった音に聞こえるのだろうか。実は、この謎には耳の構造そのものに隠されているのだ。耳には、基底膜という弾力のある膜を持っている。基底膜は、鼓膜近くの入口では高い周波数を共鳴し、奥に入るにしたがって低い周波数に共鳴する、スペクトラムアナライザの役割として働いている。また、基底膜には、内毛細胞、外毛細胞という2種類の有毛細胞が触れていて、これにより振動が中枢神経に伝達されている。そして、中枢神経に伝えられる時のS/N(信号対雑音比)が制限され、かなり低いと考えられているためだと言われている。

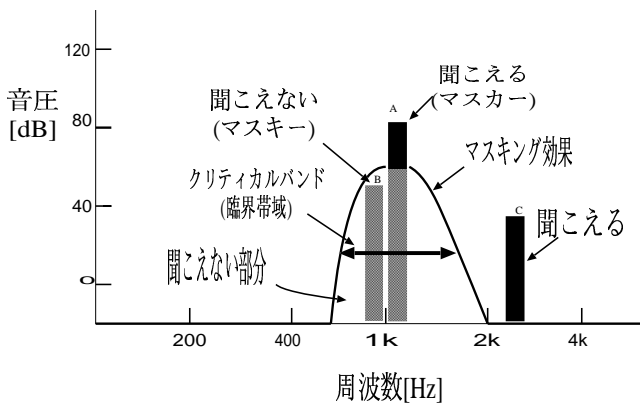


図 10: マスキング効果

4.3.2 聴覚心理符号化

先程から説明をしている知覚符号化を行っているところが「聴覚心理符号化」である。前章などで説明はしているが、もう少し詳しくみていく。MPEGはサブバンドごと、あるいは周波数ごとに細かくスケールファクタと量子化ステップを調整することが可能である。この自由度を利用して量子化ノイズを最小可聴限界以下になるように符号化する処理が聴覚心理符号化である。まず、原音と最小可聴限界から原音を感知できる限界である「マスキングスレッシュホールド(しきい値)」を求める。そして、各サブバンドごとに、量子化によって発生する量子化ノイズがマスキングスレッシュ

ルドより小さいレベルになるように、各サブバンドの量子化ステップを割り当てる。原音がマスキングスレッシュOLDより小さい部分では、原音は聞こえないため、サンプルから省くことが可能となる。聴覚の感度が高く、マスキングが十分働いていない部分では、量子化ステップが最も細かく設定される。

5 レイヤ3

今回の研究である。ハフマンエンコーディングはレイヤ3で行われる。レイヤ3はレイヤ1やレイヤ2に比べさらに符号化品質を向上させるためにs存在し、これにより周波数分解向上、データ冗長性の削減に貢献する。まず、非線形量子化におけるビット割当てを行うために、16ビット直線量子化されたPCM信号を用いて、心理聴覚に基づいた量子化誤差のマスキングレベルが計算される。このとき、同時にMDCTのブロック長が予測不可能を用いた心理聴覚エントロピーに基づいて決定される。一方、16ビット直線量子化されたPCM入力信号を用いて、心理聴覚に基づいた量子化誤差のマスキングレベルが計算され、同時にMDCTのブロック長が予測不可能を用いた心理聴覚エントロピーに基づいて決定する。一方、16ビット直線量子化されたPCM入力信号は、時間領域から32の周波数帯域へ続いてそれぞれの帯域が適応ブロック長MDCTでさらに細かいスペクトルラインに写像される。適応ブロック長MDCTは、プリエコー抑圧を目的としている。ブロック長は、心理聴覚モデルに基づいて決定された18または 6×3 のいずれかが使用さ、ハイブリットフィルタバンクにより周波数分解機能が32から $32 \times 18 = 576$ に向上する。得られた写像信号は、折り返し歪削除バタフライを経た後、心理聴覚モデルに基づいたビット割り当てに従って非線形量子化される。量子化は繰り返しループを伴っており、時間領域でフレーム境界を越えたビット配分が行われ、ハフマン符号化された後、フレームに組み込まれる。また、レイヤ3はループ

モジュールより作業が行われ、3つの階層で構成されている。では始めに最上の構成であるが、これを「ループフレームプログラム」と呼び。これは「外部ループ」というサブルーチン呼び出し、このサブルーチンは更に「内部ループ」というサブルーチン呼び出す。次に「外部ループ」の説明をする。これは別名、「ひずみ制御ループ」と呼ばれ内部ループ内の周波数成分を量子化するとき生成した量子化雑音を制御する。雑音の有色化は、量子化の前に実際の倍率を同倍率郡の範囲内で周波数成分に乗じることによって行う。実際のシステムでは各ひずみ制御ループで使用する倍率の増加とともに、このような乗算を昇順に行う。ループは通常、許容ひずみ以上の同倍率郡がなければ終了する。しかし、これが得られない場合、次のいずれかの条件によって外部ループさせる。すべての同倍率郡において、増幅が終了した。少なくとも一つの帯域が、倍率の伝送形式によって決まる上限を越えた。以上のときにループが抜ける。次に「内部ループ」である。内部ループは別名、速度制御ループと呼ばれる。内部ループは、周波数領域データの量子化を行い、同時にフレーム形式に備える。細分化及び量子化ステップ幅の選択は、このループで行う。一つのハフマン符号語をcount1という四つ組の一つを符号化するために使用する。2つの異なったハフマン符号表があり、それぞれ符号長に対応させる。count1及びrzeroに計数されない量子化値の組の個数を、その値に応じて同倍率郡3グループに分割する。

6 ハフマンエンコーディングの改良

前章まではMP3の概要について述べてきたが、ここからはハフマンエンコーディングに注目をし、更に細かく見ていこうと思う。そこから改良の道を導き出していく。

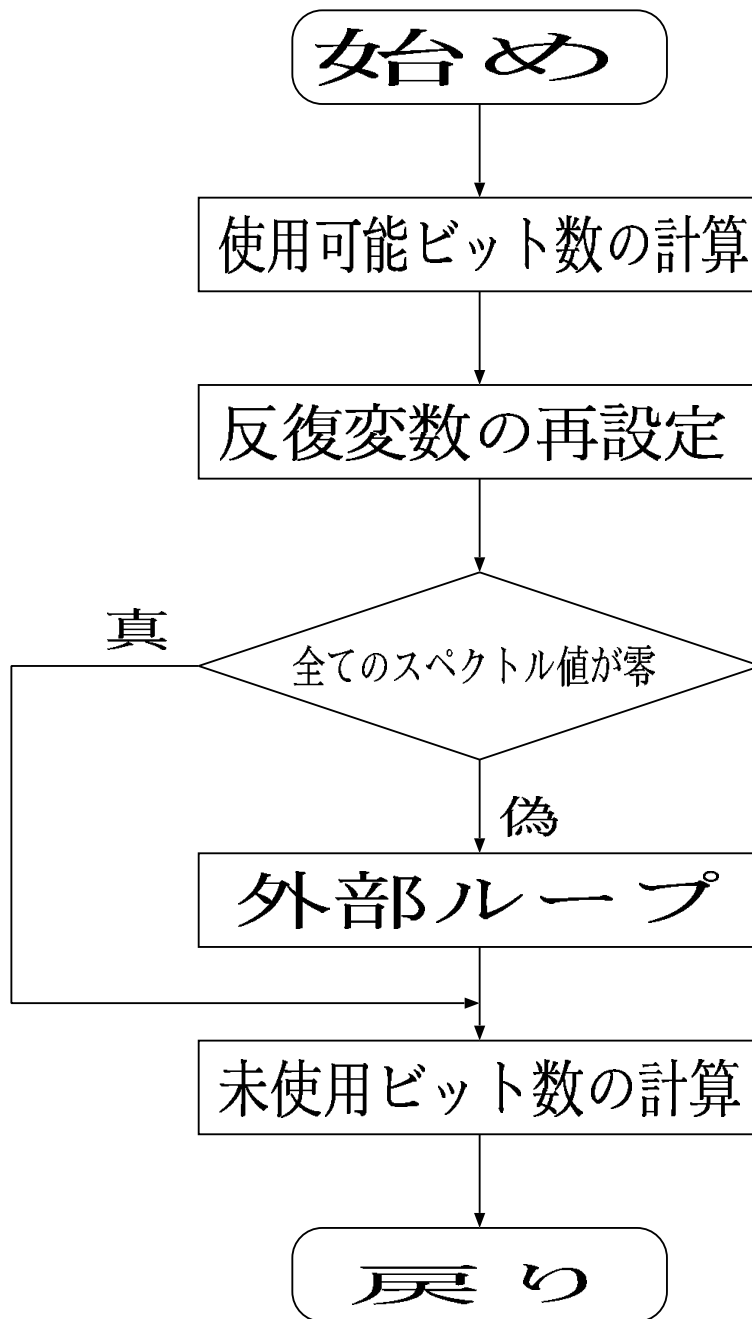


図 11: レイヤ3ループ

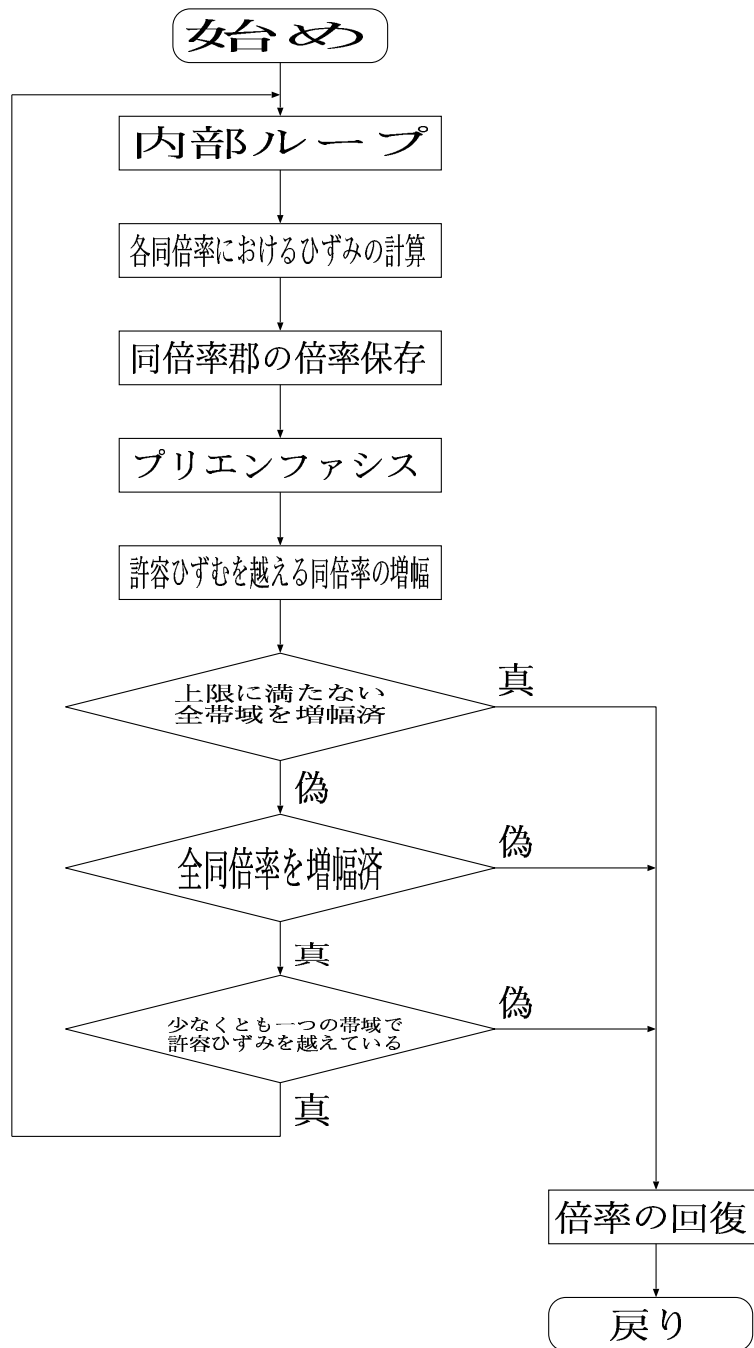


図 12: レイヤ 3 の外部ループ

ノ

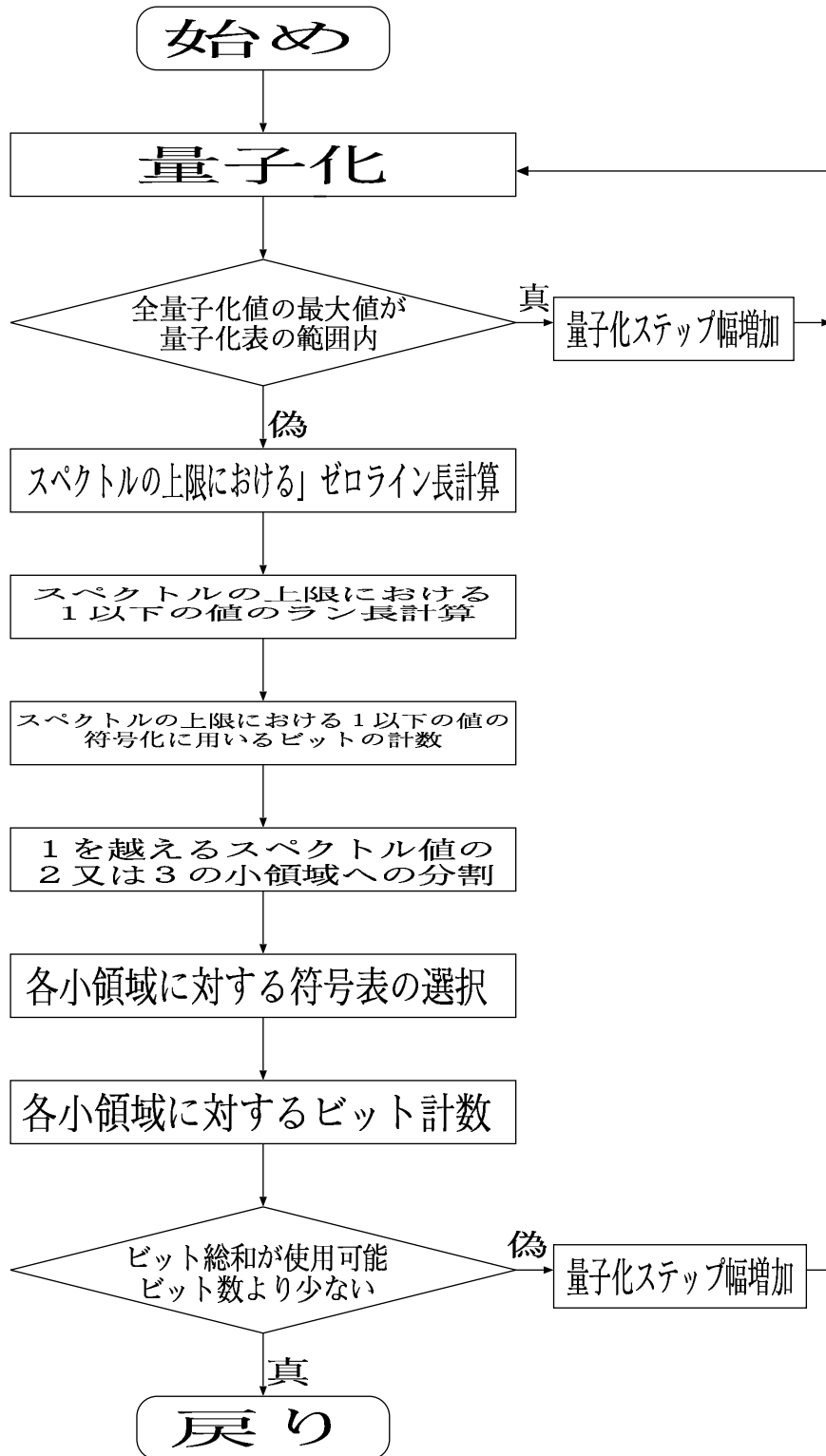


図 13: 内部ループ

6.1 ハフマンエンコードの記述

ハフマンエンコードに関しては、図14のプログラムが仕様書(ISO/IEC11172-3)に書かれている。他のエンコーディングの記述よりも少なめである。仕様書内でもこの部分しかハフマンエンコーディングについては述べられていないのが現状である。これをどう改善していくかが論点となる。

6.2 MP3のデータ構造

MP3のデータ構造はフレームの内部に存在している。まず、フレームヘッダーの後にエラーチェック、サイド情報、メインデータと続いている。またサイド情報とメインデータをあわせてオーディオデータとも呼ばれている。エラーチェックの有無はフレームヘッダーに含まれている保護ビットの値で決まる。そしてサイド情報のバイト数も同じくフレームヘッダー内のチャンネル数で決まる。メインデータにはオーディオデータ本体が格納されている。

6.2.1 サイド情報

サイド情報では主にメインデータブロック内のデータ形式や配置に関する情報、そしてグラニューの情報も保存されている。このグラニュー情報というのがMP3ではとても重要なものとなる。グラニューはMP3独特のものでサンプルデータの集まりの最小単位である。必ず576サンプルを含むことになっている。また、1フレームに必ず2グラニューが格納されている。つまり $576 \times 2 = 1152$ サンプルが1フレームであることになる。

6.2.2 エラーチェック

MP3はデータに誤りを検出し場合によっては誤りを修正することができる。この機能はオプションであるためエンコーディング・デコーディングであっても処理を実装しなくても構わないよ

うになっている。特にMP3で使用しているレイヤー においてエラーチェックをMP3の全てのデータに対して行うのではなく、フレームヘッダー後半2バイトとサイド情報の17バイトまたは32バイトに対して行っているためメインデータはエラーチェックされないのである。この理由については仕様書に「エラーが起こった時の重大性はメインデータ情報よりもヘッダーやサイド情報の方が高い」と掲載されており、この根拠を基にメインデータのエラーチェックは行われていない。

6.2.3 メインデータ処理

メインデータではオーディオのサンプル値が格納されている。つまり、ハフマン符号はメインデータの中に格納されているのである。こう考えるとフレーム処理はメインデータは全てこの中に含まれていると思いがちであるが実は、そうではない。レイヤー にはビット蓄積と呼ばれる仕様がある。このフレーム処理のメインデータの先頭は基本的に一つ前のメインデータにある。ちなみにビット蓄積とは、隣接するフレームの間でオーディオデータの保存領域の貸し・借りをする仕組みである。これにより、全体としては決められたビットレートを保ちながら、個々のフレームではビットレートに捕らわれずに一定の音質を維持することができる。MP3のフレームは単体で処理できないのである。これはビット蓄積に限られたことではなくデコード処理においても隣接するグラニューとデータを連携させながら処理を行っていく。フレームは独立したデータとして扱うのは不可のである。

6.2.4 スケールファクタ

スケールファクターとは、10進数で言うならば桁取りのことである。例えば「35120」と表現するよりも「 3512×10 」と表現した方がすっきりする。このデータをファイルに保存するならば「3.512」と「1」の二つのデータに分けて保存するとファイルサイズが小さくでき

```

HuffmanCodebits(){
    for(i=0;i<big_values*2;i+=2){
        hcod[|x||y|]
        if(|x|=15&&linbits>0)
            linbitsx
        if(x!=0)
            signx
        if(|y|=15&&linbits>0)
            linbitsy
        if(y!=0)
            singy
        is[l]=x
        is[l+1]=y
    }
    for(i<big_values*2+count1*4;i+=4){
        hcod[|v||w||x||y|]
        if(v!=0)
            signv
        if(w!=0)
            signw
        if(x!=0)
            signx
        if(y!=0)
            signy
        is[l]=v
        is[l+1]=w
        is[l+2]=x
        is[l+3]=y
        for(l<576;l++)
            is[l]=0
    }
}

```

図 14: 仕様書に掲載してあるプログラム

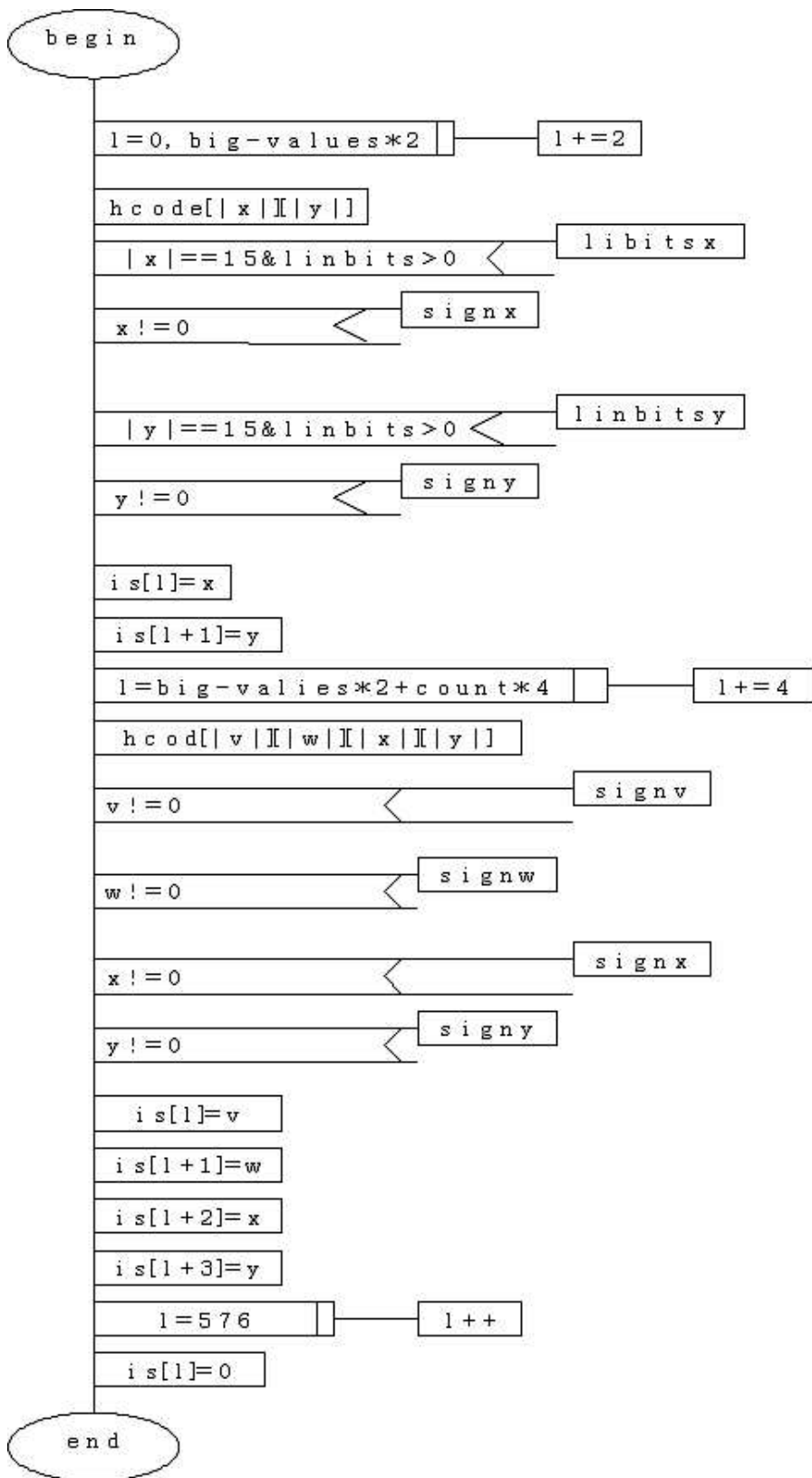


図 15: 仕様書からのフローチャート

オーディオデータ

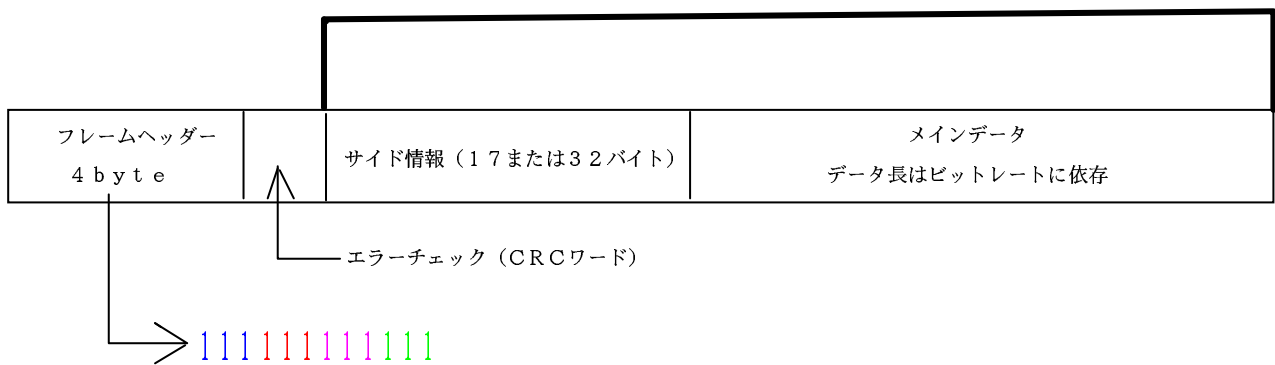


図 16: フレーム内部のデータ構造

る。MP3で使われているものはもっと複雑であるが指数の1がMP3のスケールファクタになる。また「3.512」というデータがハフマン符号ビット列に保存されている。

1グラニュールには576のデータがあるとは先程述べた。ということは、ハフマン符号化をすると576のデータが得られるのは解かるだろう。しかし、スケールファクタは576個の保存をしなくては行けないわけではない。576のデータを13または22のバンドに分けバンド毎にスケールファクタを保存するのである。この分け方から生まれるバンドをスケールファクタバンドという。

6.2.5 ハフマンコードビット列

ハフマンコードビット列は、ハフマンエンコーディングにとって重要なものとなるので次の項目で詳しく述べていこうと思う。よって、今はハフマンコードビット列とスケールファクタビット列を位置づけした図のみに留めておく。

6.3 データ構造からみたハフマンコード

実は、ハフマンコードビット列には1グラニュール分の576データよりも少ないデータ数を保存する。576データに満たない分は0の値とするからである。このことをZERO領域と呼ばれる。ハフマンコードビット列には大きく分けて2つの領域となる。前半は-8206~8206の値が2つずつ得られるBig Value領域、後半は-1、0、1の何れかの値を4つずつ得られるCount領域である。全体は周波数の低い方から高い方への配置となっているが、この場合周波数が高くなるにつれて信号の値が小さくなりやがて0になることを選定にしている。Big Value領域のデータ個数はグラニュール情報のbig valuesを2倍とすることで得られ、Big Value領域終了後からCount領域となる。ハフマン符号化の原理というのは前半でも述べたように簡単に言うと分岐をたどって辿り着くとこ

ろを着地点とする分け方である。MP3のハフマン符号化は多少変わってくる。ビット値の0、1を使い、ゴールには4つの整数値がある。それに木構造には31種類も用意割れている。それが仕様書にも掲載されているハフマン符号表のことである。ハフマン符号表は、0から31までの連番が付けられているもの、A、bとアルファベットがつけられた2つの表が定義されている。前者はxとyと呼ばれる2つの値を得ることができ、Big Value領域で適用される。反対に後者はv、w、x、yの4つの値を得ることができCount 1領域で適用される。MP3のハフマン符号化の基礎は各データのビット数が同じではない。ビット数は出現確率によって大きいものほど小さく、出現確率が小さいほど大きくなるように定義されている。また、MP3ではハフマンテーブルが固定になっているのでハフマン符号化と呼ばれていてもハフマン符号化のアルゴリズムとは直接関係ない。

6.3.1 Big Values

各グラニュールのスペクトル値は異なるハフマン符号表を用いて符号化する。0からナイキスト周波数までの全周波数帯域を幾つかの領域に分割して、それぞれ異なる符号表を使ってエンコーディングを行う。分割は最大量子化値に応じて行う。これは高い周波数の値が小さいか、または全く符号化する必要がないことが期待できると仮定に基づいている高い周波数から開始し、値が0に等しい量子化値の組を数える。数えた結果をzeroとする。次に、絶対値1以下の量子化値の4つ組の個数を求める。これをcountとする。再び偶数個の量子化値が残る。最後に、周波数0までの領域のスペクトル量子化値の組の個数をBig Valuesとする。

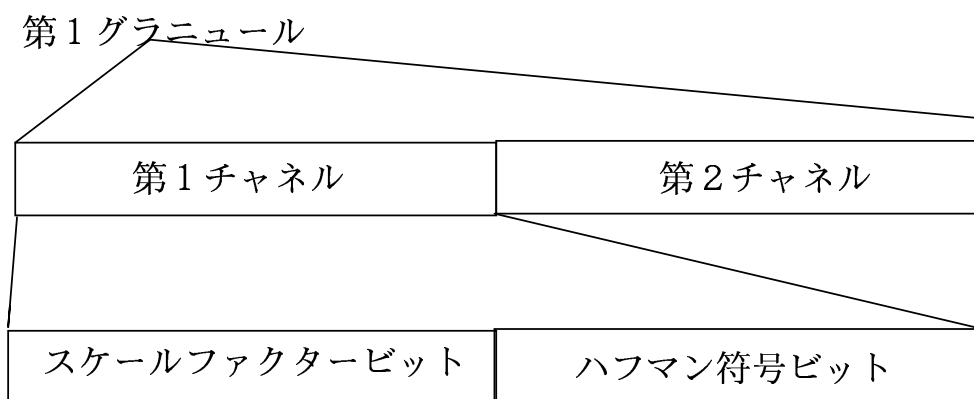


図 17: メインデータのデータ構造

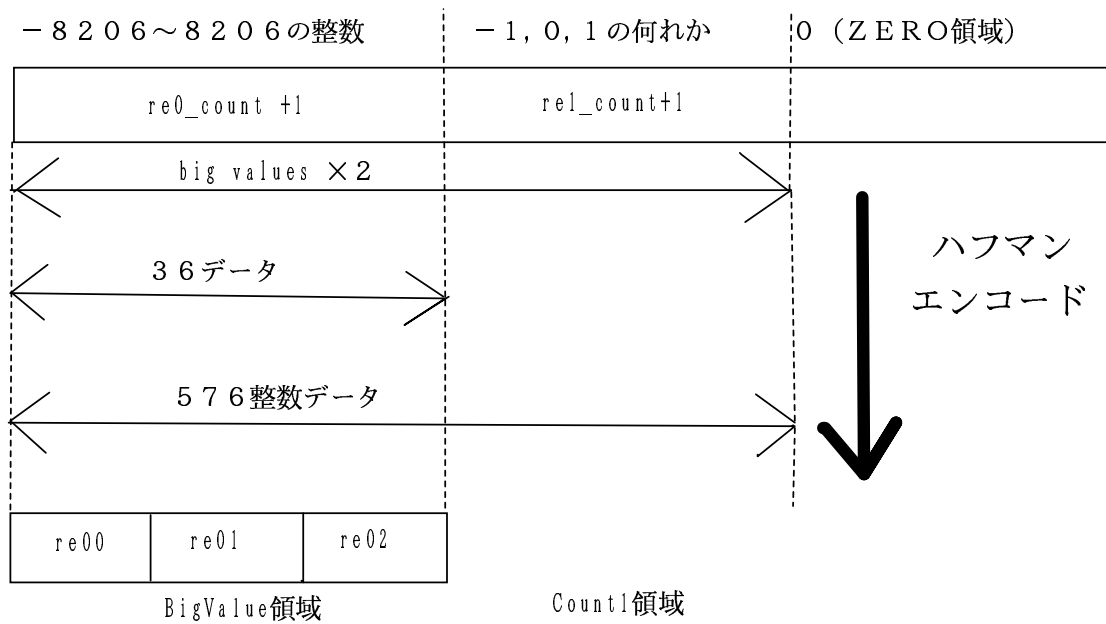


図 18: ハフマンコードビット列の構造

6.3.2 Big Value領域のハフマン符号化

Big Value領域では先程も述べたとおりハフマン符号表の0~31を使用する。結果は-8206~8206の値が2個1組となる。MP3のハフマン符号には拡張仕様というものがある。これはこの章の始めの辺りで示した「仕様書に掲載してあるプログラム」または「仕様書からのフローチャート」を見てもらうと解かりやすいと思うが15を超える大きな数値の加算分を示すlibitsと正負の符号をあらわすsignと呼ばれるものがある。libitsはテーブル事に定義する。libitsは最大13ビットとなり、Big Value領域のデータの最大値は

$8206 (= 15 + 2 \times \text{最大 libits 乗} - 1)$ となる。上記に述べたハフマン符号表(0~31)には違いがある。一つは符号化可能な値の最大値の違いである。狭い範囲に分布する値の符号化の方が符号のビット長を小さくできるので値の分布がテーブルの範囲内に可能な限り収めるテーブルを選択するのだ。

ハフマン符号表には値の最大値が同じものが幾つかある。これは符号ビット長の傾向の違いである。例えば表2と表3は値の最大値が共に2になるが、表2は小さな値を少ないビット数で符号化できるが、大きな値は多くのビット数を必要とする。一方、表3は全体的に中位のビット数を符号化できる。つまりエンコーディングは全ての表で実際に1領域分の符号化を行って総ビット数を最小にできる表を選択しなくてはならないのだ。

6.3.3 Count 1領域のハフマン符号化

Count領域の場合は、4値ハフマン符号表Aまたは4値ハフマン符号表Bを使用する。libitsは無く結果は-1, 0, 1の何れかの値が4個1組となる。上記に述べたBig Value領域との違いはCount領域の値の数は構文中の領域によって明示的に符号化しない。count領域の最後はそのグラニューールのすべて

のビットを取り出したときに明らかになる。

ハフマン符号化データの並び順は、そのグラニューールのblock_type(ブロック数値の並び方、変換の長さ及び変換の回数についての情報を与える。)に依存する。block_typeが0、1又は3のとき、ハフマン符号化データは周波数の昇順に並べる。block_type==2のとき、ハフマン符号化データはそのグラニューールの倍率値の並びと同じ順番で並べる。ハフマン符号データは、同倍率群0から始める同倍率群の順に与える。各同倍率群内では、データは窓0から始まり窓2で終わる連続する時間窓の順に並べる。各窓内では、量子化値を周波数の昇順に配置する。

6.4 問題点

今までみてきたようにハフマンエンコーディングは、他のエンコーディングのもの比べ、他に依存している部分が多い。そのため、複雑化かつ難題化させている。特にハフマン符号表の選択に関しては一筋縄ではいかないことばかりである。どれをどう選定するか、更に高速化するには事前に膨大なデータサンプリングを必要とする。それをどう圧縮しておくか等の課題が非常に多いことが解かった。

6.5 改良

私は、この問題点をどう回避していかうかを考えていった。特に事前にデータサンプリングを用意するのは困難極まりないことである。また逆に一定の部分しかできない、つまりフィルタ等を使って制御をすると音質が悪くなってしまう。そこで私は一つの考えをあみ出してみた。それは、MP3を用いて行うことといえば、ポータブルプレイヤーに使うことが一番多いと思う。ポータブルプレイヤーは基本的に個人でしか聴かないものである。そうすると私もそうだが好きな音楽ジャンルしか聞かないということだ。人間というものは好き嫌いというものがあり、音にもそれは言える。

表 3: 各ハフマン符号表の符号化可能な値の最大値 (参考文献 14 から引用)

表番号	linbits	x、yの最大値	値の最大値
1	0	1	1
2	0	2	2
3	0	2	2
5	0	3	3
6	0	3	3
7	0	5	5
8	0	5	5
9	0	5	5
10	0	7	7
11	0	7	7
12	0	7	7
13	0	15	15
15	0	15	15
16	1	15	16
17	2	15	18
18	3	15	22
19	4	15	30
20	6	15	78
21	8	15	270
22	10	15	1038
23	13	15	8206
24	4	15	30
25	5	15	46
26	6	15	78
27	7	15	142
28	8	15	270
29	9	15	526
30	11	15	2062
31	13	15	8206

BigValue領域

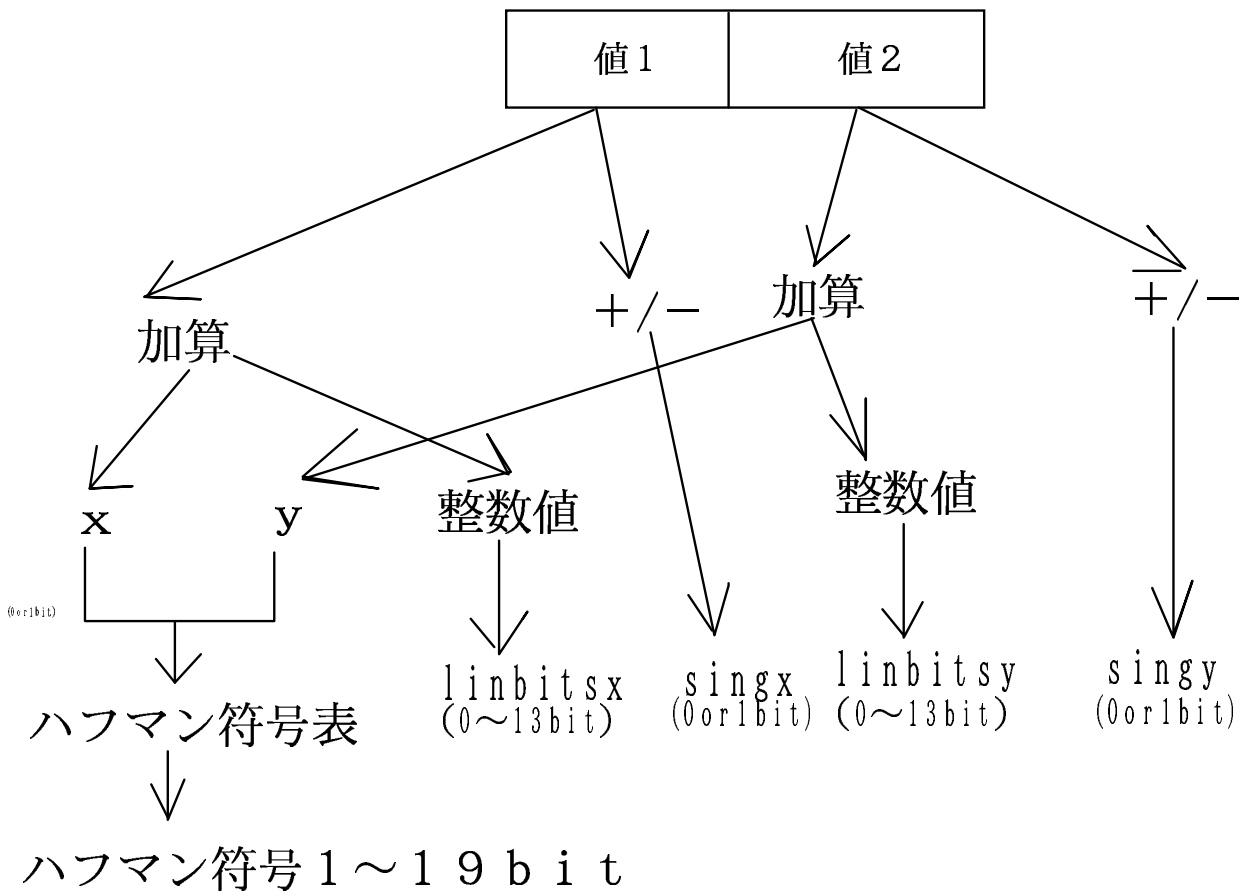


図 19: BigValue領域のハフマンエンコード

表 4: 4 値のハフマン符号表 A

v	w	x	y	hlen	hcod
0	0	0	0	1	1
0	0	0	1	4	0101
0	0	1	0	4	0100
0	0	1	1	5	00101
0	1	0	0	4	0110
0	1	0	1	6	000101
0	1	1	0	5	00100
0	1	1	1	6	000100
1	0	0	0	4	0111
1	0	0	1	5	00011
1	0	1	0	5	00110
1	0	1	1	6	000000
1	1	0	0	5	00111
1	1	0	1	6	000010
1	1	1	0	6	000011
1	1	1	1	6	000001

表 5: 4 値ハフマン符号表 B

v	w	x	y	hlen	hcod
0	0	0	0	4	1111
0	0	0	1	4	1110
0	0	1	0	4	1101
0	0	1	1	4	1100
0	1	0	0	4	1011
0	1	0	1	4	1010
0	1	1	0	4	1001
0	1	1	1	4	1000
1	0	0	0	4	0111
1	0	0	1	4	0110
1	0	1	0	4	0101
1	0	1	1	4	0100
1	1	0	0	4	0011
1	1	0	1	4	0010
1	1	1	0	4	0001
1	1	1	1	4	0000

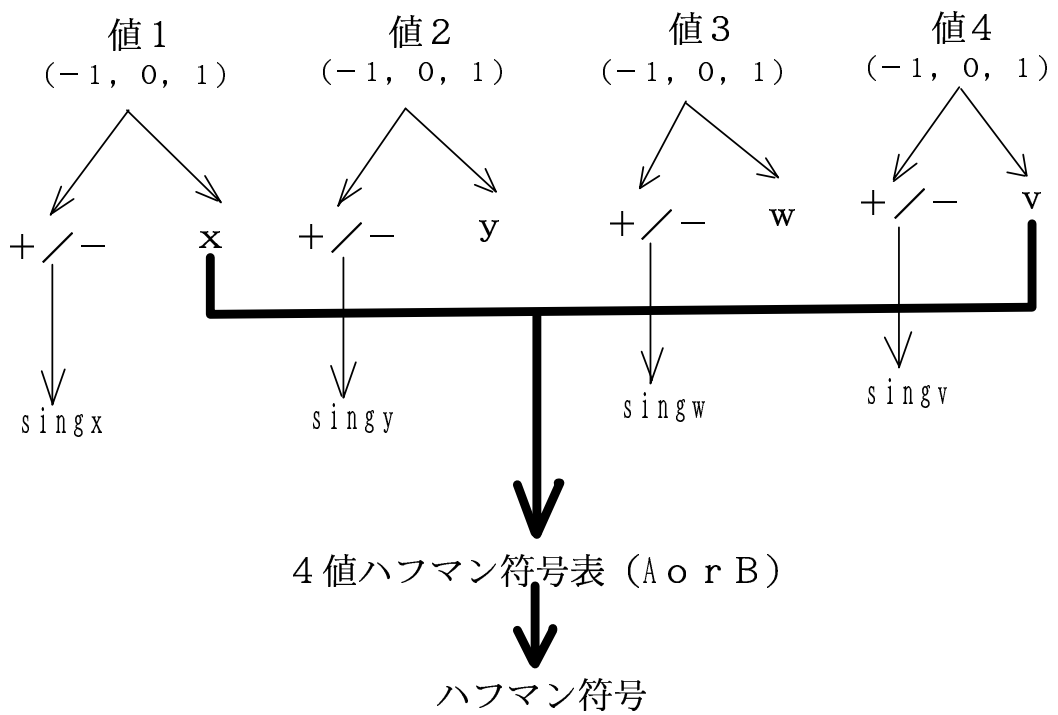


図 20: Count 領域のハフマンエンコード

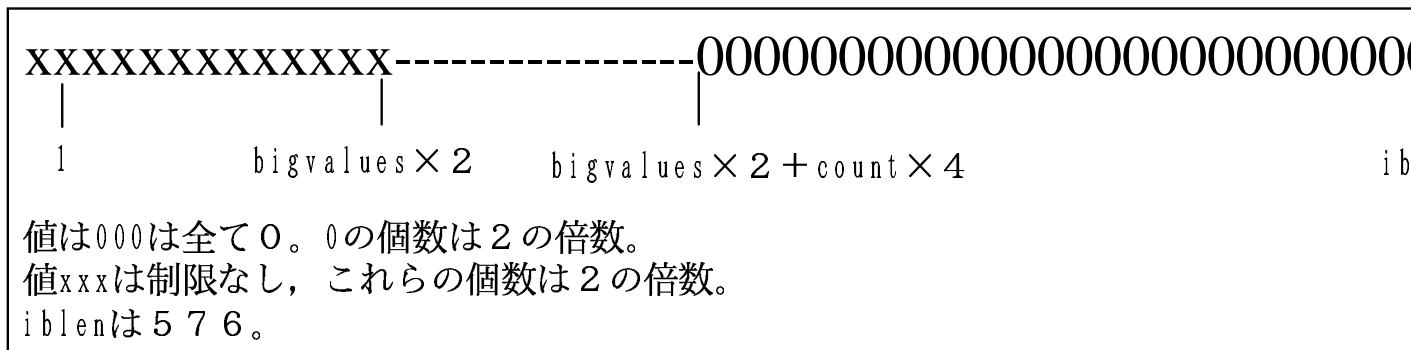


図 21: bigvalues をまとめたもの

好きな音楽のリズムやテンポといったものはほとんど変わらない。よって周波数も大きくは変わらないのである。そこで私は人工知能のような記憶性を持たせてみてはどうかと思った。記憶製を持たせることにより、その人の聴く傾向に合わせられる。そうなれば、ある程度の記憶があれば、エンコーディングもできるのではないかと思う。つまり記憶をし、なおかつデータ検索をさせるということだ。

量子化での時点ではあるが実際、音声の統計データを利用してエンコーディングの改良を研究している例もある。この時は、検索範囲を初期量子化ステップから + 200 と広範囲にしたために問題となったと指摘されているが、ハフマンコードのエンコーディング上なら可能ではないかと思う。ハフマンエンコーディングの場合、頻度の高いものから小さな番号で振り分けていくのである程度の番号までを記憶しておくなどすれば、問題はないと思う。私はこれを応用していきたいと考えている。

7 今後のMP3の動向3

ここでは最後に今後のMP3に関する世の中の動きについて少し触れておこうと思う。

7.1 ソフトウェア

7.1.1 圧縮符号方式

MP3の基である。「MPEG」に関係のある方式で今後、MP3にも影響がある。「オーディオ圧縮符号方式」を説明する。MPEG-1オーディオのサンプリングレートを半分に落したLSF(低サンプリング周波数)仕様と、MPEG-1オーディオを拡張した5.1チャンネルを可能としたMC(マルチチャンネル)仕様が含まれている。LSFとMCは、まったく別の仕様である。LSFはインターネットなどの低ビットレート用としてMCはホームシアターの5.1チャンネル用として作成されてい

る。これによりMPEG-2 MCは5.1チャンネル方式でありながら、MPEG-1オーディオデコーダでメイン2チャンネルを再生できる。

7.1.2 AAC

MPEG-2にはAACという圧縮方式がある。AACとはオーディオ高圧縮率符号化技術と呼ばれ、この方式は、MCやLSFとは異なる、従来のオーディオ方式とはまったく関係なく標準化されている。MPEG方式よりも性能が良いオーディオ圧縮符号化方式がいくつか時代とともに現れた。そこでMPEGでもそれらの技術を取り入れた新しい標準も作ることにしました。その新しい符号化方式として誕生したのが「AAC」なのです。この技術は日本のデジタル放送にも採用されています。AACには、オプションとして利用できる処理が決められている。

1. 入力データ

入力データは、オーディオ信号でAAAの場合2チャンネルのステレオ信号や5.1チャンネルの信号が入力となる。

2. ゲインコントロール

ゲインコントロール(利得制御)は、カスケットなどの立上りの鋭い音に対する性能を若干向上させる処理である。(オプションとなっているので日本のデジタル放送では使っていない。)

3. フィルタバンク

フィルタバンクはMDCTと呼ばれているアルゴリズムを用い、信号を128または1024の周波数成分に分解する。

4. TNS、インテンシティ/カップリング、予測器、M/S

TNS、インテンシティ/カップリング、予測器、M/S(ステレオ信号を左右の和と差に分けて行う符号化)という処理は、圧縮符号化効率を向上するのに役立っている処理である。

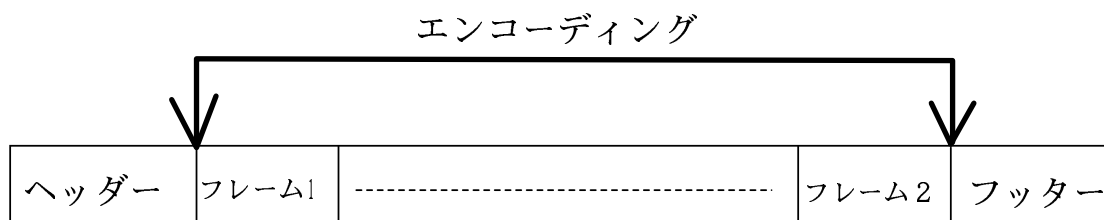


図 22: エンコーディング部分の簡易図

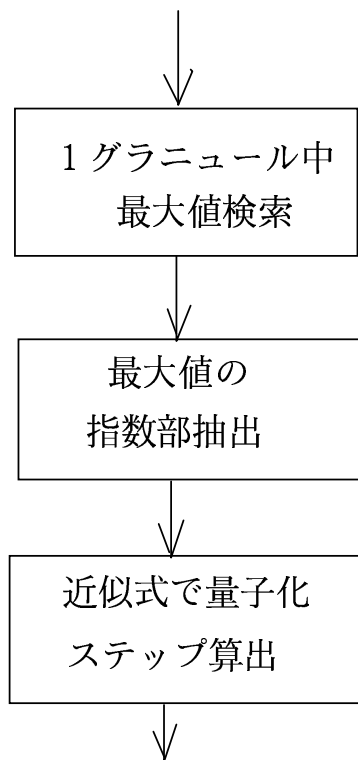


図 23: 初期量子化ステップ算出の流れ

5. スケールファクタ

スケールファクタは MDCT 係数などの大きさを表す指数を算出する処理で、これによって非常にダイナミックレンジ (音声信号の最小値と最大値の比率のこと) の大きな音を歪みなく符号化できる。

6. 量子化器

量子化器は、これまで精度に制限なく処理してきた信号ここで始めて整数値に丸めて (小数点を切捨て、整数にすること)、伝送に適した形する。

7. ノイズレスコーディング

ノイズレスコーディングは、量子化で得られた MDCT 係数を整数化した値やスケールファクタをそれら p 数学的な性質を使ってコンパクトに圧縮する。

以上のような仕組みを基に新しいプロファイル (圧縮機能のセット) 技術も生まれており、MP 3 にも多用されている。

1. 適応ブロック長 MDCT

1024 サンプルのブロックと 128 サンプルのブロックの組み合わせ

2. TNS (瞬時ノイズ成型)

量子化ノイズの最適化

7.2 MPEG 4

実は最近では、MPEG-4 というものが水面下で動き始めている。これはデジタルオーディオビデオデータ (AV データ) を用いた通信、アクセス、データ加工を行う新しい手法である。この MPEG-4 を開発された背景には、通信、コンピュータ、テレビ・映画の境界線が現在無くなりつつあるということである。つまり、すべてのものを一つに終結させようと言うことだ。更に今まで開発されてきた MPEG との大きな違

いは、柔軟性と拡張性に優れている点である。そのため MPEG 4 の目指す主要として掲げられている機能はコンテンツ単位のノインタラクティブィ、自然・合成混合のデータ符号化、エラーの多い環境での耐性、遠隔監視と制御、そして高圧縮である。

最後に出た高圧縮は符号化効率の向上や複数の多重データ列の符号化が目標となっている。よって MPEG 4 のターゲットとして特定されたアプリケーションに対して、従来の標準と比較して同程度のビットレートならば主観的に良好な品質ができ、複数のビット列同期もとれるのである。

このような機能の働きにより従来では実現できなかった 64 kb/s で CCIR Grade 4 の品質を目標としているのである。

以上のように MPEG 4 はまだ私達の生活の中では、耳慣れない言葉ではあるが、これが完成された広範囲に使える MPEG 4の方が確実に需要が増えるといえる。まさに MP 3 にとっては脅威的存在といっても過言ではない。

そのためにも MP 3 の更なる改良が必要であると私は考える。今後の動向に非常に注目である。

7.3 ハードウェア

7.3.1 LSI

ハードウェアの面では音質追求のものが増えてきており、その中でも今後発展がありそうなのはハードディスクデコーダ (HDD) である。最近では、TV の長時間録画などで私達の生活にも浸透をし始めているが音の世界でも使われる時代が来ているようだ。その先駆けとして沖電気工業株式会社が先日行われた Embedded Technology 2004 にて HDD Audio Player 用の LSI を発表した。

この LSI を使うことによりパソコンと機器との間で非常に高速のデータ転送を可能とすることが可能となったのである。更に HDD であるため容量も非常に多く保存、そして音質にもこだわられる可能が生まれるわけである。そうすると益々、音への追求というのが増えるだろうし、確実に MP

3を求められるようになるのではないかと私は予想する。

8 まとめ

この研究を行うにあたって、実際にハフマンエンコーディングの流れを掴むまでに非常に長い時間を費やすことになってしまった。その背景としていえるのは、量子化やM D C Tのように計算によって導き出されるような数値とは違って「頻度」というものに注目しなければならない。そのためには、大卒だけの知識では解からなくなってしまふというのが身に滲みて解かった。

ハフマンエンコーディングはデコーディングのような復元とは違い、自分で構築をしていかないといけない。一見、勝手に決められるようで非常に便利であるかのように思えるが、ハードウェア上で実現するにはそれが仇となるのが解かった。設計をしていくために基準が必要であるからだ。その辺りが今後の実装していく上での一番苦労するところではないかと思う。

高速化と音質の同時をあげるというのは、なかなか難しいものがあるようだ。まずは、「音質」か「高速化」かを一本に絞ることが今後の改良していくための近道ではないかと考えられる。

9 感想

最初は音楽が大好きというだけでこの研究を取り組み始め、機械の部分について解かるかと言われると自信を持って答えられないようなところから私の研究が始まりました。

そのため、解からないことばかりの毎日でした。更にはハフマンコードの部分は資料が少なくほとんど仕様書しかツールがなく、いろいろな方にご迷惑をおかけしながら研究をしていくことに結果的になってしまったのは自分にとって悔いの残るところです。

しかし、この研究を行い機械についての凄さと同時に技術者の大変さがよく解かりました。よく開発にはドラマがあるとよく聞きますが、まさにそ

れを実感できました。初歩的なことではありますが、何でも機械を使い便利になった世の中だからこそ忘れていないかと思えます。

他にもこの一年の研究を通して自分にとって初めての経験だったり、自分にとっての甘かった部分等について考えさせられたり、良い経験が出来ました。この経験を活かして今後の自分に役立てていきたいと思えます。

10 今後の方針

今回は、結果的に理論上でのことしか述べれず、実際にハードウェア上へ実装というところまでは出来ませんでした。今後、この研究をどこかで続けていくことが出来るとしたら、この理論を実装し、製品として出来るくらいのレベルまでに仕上げたいです。そして更にもっと音質について追求をし、容量が軽くても音質がCD並みくらいになるように考えていきたいです。

11 謝辞

本研究を進める上で経験が浅い私を御指導して頂きました、東海大学電子情報学部コミュニケーション工学科清水尚彦助教授に感謝致します。また、技術的指導を頂きました田中大輔氏を始めとする清水研究室諸学生、及び、友人達に感謝致します。

12 参考文献

参考文献

- [1] 田中 大輔,MP 3 エンコード,デコードに適した浮動小数点積和演算器の設計
- [2] 神山智章,MP 3 技術とその改良評価
- [3] 神山智章,ハードウェアMP 3 エンコードの設計

- [4] 大矢雅則, 渡邊昇, 数理情報科学シリーズ17
量子通信理論の基礎—量子情報から光通信
へ—, 1998年4月10日 初版第1刷, 星
雲社 発行, ISBN4-7952-0117-X C3342
- [5] テレビジョン学会, 総合マルチメディア選書
MPEG, 平成8年4月20日 第1版第
1刷, オーム社 発行, ISBN4-274-07840-X
- [6] 亀山 渉, 花村 剛, IDG 情報シリーズ MPEG-
1 / MPEG-2 / MPEG-4 デジタル放送
教科書 (上), 2003年2月1日 初版, 株式
会社 IDG ジャパン 発行, ISBN 4-8728
0-4775
- [7] 小野 定康, 鈴木 純司, わかりやすい JPEG
/ MPEG 2 の実現法, 平成7年7月15日
第1版第1刷, 株式会社オーム社 発行 ISBN
4-274-07819-1
- [8] 藤原 洋 監修, マルチメディア通信研究会 編
集, ポイント図解式最新 MPEG 教科書, 19
95年12月21日 第1版第5刷, 株式会社
アスキー 発行, ISBN 4-7561-024
7-6
- [9] 安田 浩, MPEG / マルチメディア符号化の
国際標準, 平成6年9月30日 第一版, 丸善
株式会社 発行, ISBN 4-621-0398
7-3 C3055
- [10] 日本工業標準調査会 審議, デジタル記録媒
体のための動画信号及び付随する音響信号
の1.5 Mbit / s 符号化—第3部 音響 JIS
X 4323 = 1996 (ISO / IEC 111
72-3 : 1993), 平成8年11月20日
制定
- [11] HIEHATA Yasuhiko, なまえまだ
ない, ハフマン符号化プログラム詳
細, <http://www.enteh.org/hiehata>
- [12] インターネットラジオ局4畳半ヴ
ギ, <http://www.0405.net>
- [13] NETWARP, <http://homepage2nifty.com/netwarp>
- [14] 最初から説明する Inside
MP3, <http://www.geocities.jp/bywnn498/mp3>
- [15] ウェブ におんがく
を, <http://www.interq.or.jp/blue/inside/index.html>
- [16]