

MP3 エンコーダの高速化

李 芝 剛[†] 神 山 智 章^{††} 清 水 尚 彦^{††}

MP3 エンコーダの高速化にあたって非線形量子化部分の高速化を行った。高速化方法としてループを使わずに最初の量子化ステップを算出し、最適なハフマンビットを得るために二分探索法を用いた。これにより、非線形量子化の回数を削減し、MP3 エンコーダ全体の高速化を図ることができる。

An Optimization of MP3 Encoder for Faster Execution

ZHIGANG LI,[†] TOMOAKI KOUYAMA^{††} and NAOHIKO SHIMIZU^{††}

This paper proposes the non-linear quantization part of MP3 Encoder for speed up. We propose calculation of the initial quantization step not using a loop and search of the optimal Huffman bit length using bisection search. This method can reduce non-linear quantization and lead to speedup of the whole MP3 Encoder.

1. はじめに

近年、LSI 製造技術、設計技術、ならびに情報圧縮伸長技術の進展により、デジタル音声および動画が普及してきている。MPEG (Moving Picture Expert Group)¹⁾ 規格は、ISO (International Standards Organization)²⁾ で策定され、ビデオ CD、デジタルビデオに採用されている動画の圧縮伸長規格である。中の音声に関する規格 MP3 (MPEG-1 Audio Layer3)³⁾ とともに、現在最も一般に普及している規格である。

しかし、MP3 という規格は、デコーダの仕様のみに定められているという、変わった特徴を持っている。これは、エンコードにはいかなる手段を用いてもかまわないことを示している。ただし、デコーダでデコードできる必要はある。これにより、特に大変な処理であるエンコード処理が、規格に縛られることなくさまざまな方法で改良等を行っていくことができるようになっていく。

MP3 の圧縮技術は、多くの複雑な計算をともなうため、負荷の大きい処理となってしまう。MP3 エンコーダの処理自体に手を加え、計算処理を削減することが必要である。実際にプロファイルで見てみると、非線形量子化処理が特に計算量が多いことが分か

る。我々は、エンコードの基本処理部は ISO²⁾ と等しく、操作性等を改良した 8hz-mp3⁴⁾ を用いて非線形量子化の計算量の削減を行った。

本報告では MP3 エンコーダの各ルーチンの処理時間をソフトウェアで測定した。ISO の仕様書³⁾ の方法と文献 2), 4) の方法で量子化ルーチンは全体の処理時間の 45%と 30%を占めている。このルーチンを高速にすることによって、全体の処理時間を短くすることができる。

2. MP3 について

MP3 は、ISO/IEC のワーキンググループで策定された動画音声符号化に関する国際標準規格 MPEG の、音声符号化に関する規格の一部である。MP3 では、人間の聴覚特性を利用して、知覚が難しい音の情報を省くことにより、圧縮率を高めている。そのため、10 分の 1 程度に容量を減らす一方で音質の劣化が少ない特徴を持つ。一般的な MP3 エンコーダのブロック図を図 1 に示す。本章では Hybrid Filter Bank, Iteration Loop, およびハフマン符号器について順に概略を述べる。

2.1 Hybrid Filter Bank

MP3 では、サブバンドフィルタと MDCT (Modified Discrete Cosine Transform)⁵⁾ を組み合わせたハイブリッド構成により、入力の 16 bit 直線量子化された PCM データを時間領域から周波数領域に写像している。サブバンドフィルタは 512 タップの PFB (Polyphase Filter Bank) であり、時間領域から 32

[†] 東海大学大学院工学研究科電気工学専攻
Graduate School of Engineering, Tokai University

^{††} 東海大学工学部通信工学科
Communication Engineering Department, School of Engineering, Tokai University

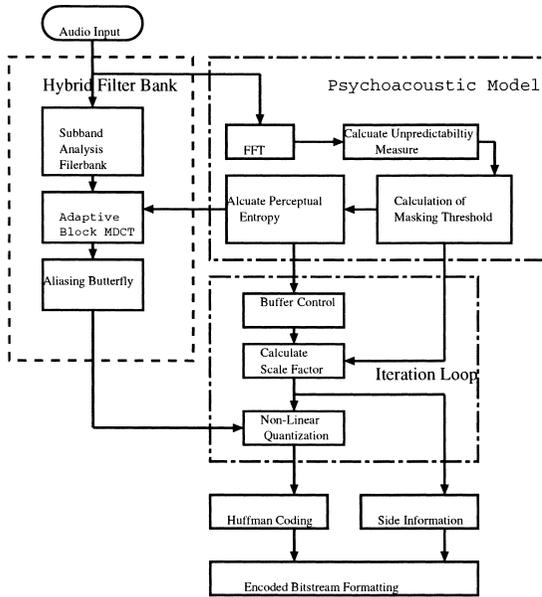


図 1 MP3 エンコーディングのブロックダイアグラム
Fig. 1 Block diagram of MP3 Encoding.

の周波数帯域に写像する．MDCT はプリエコー抑止と圧縮効率を考慮して 2 種類のブロック長を切り替える適応ブロック長 MDCT であり，ロングブロック時の MDCT ウィンドウ幅は 36 で 18 の周波数領域に写像されることから，PFB から出力される 32 の帯域すべてに MDCT を施して得られる周波数領域は最終的に 576 となる．MDCT 出力はエイリアシング歪みを含んでいるため，MDCT 出力にエイリアシング歪みを除去するバタフライ演算を行う．

2.2 Iteration Loop

2.2.1 非線形量子化

MP3 オーディオ信号圧縮に用いる非線形量子化を以下の数式に示す．

$$I = \left\lfloor \left(\frac{|X|}{2^{\frac{q}{4}}} \right)^{0.75} - 0.0946 \right\rfloor \quad (1)$$

ここで，量子化ステップ q は整数である．また， $\lfloor \cdot \rfloor$ は小数点以下を四捨五入して整数化する演算である．この演算を 1 グラニュールのすべてに対して行う．

文献 7) では非線形量子化公式を直線近似で求めて，計算量の削減を図る．文献 8) では仮数部と指数部を分離して，仮数部の計算を二項定理を使用した．

2.2.2 量子化ステップについて

q の値は量子化後の I の当該フレームにおける最大値が，あらかじめ決められた値より大きくなってはならないという MP3 の規定によって，通常 (負の値と

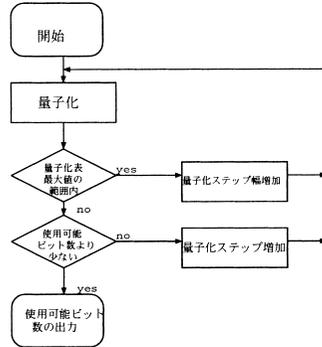


図 2 ISO 仕様書の処理図
Fig. 2 Sample flow chart in the ISO specification.

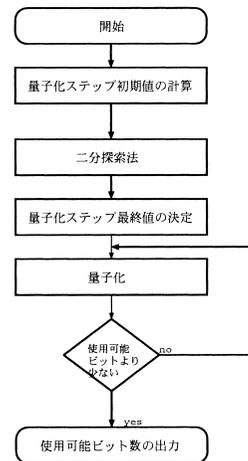


図 3 ISO ソースと 8hz-mp3 の処理図
Fig. 3 The flow chart in the ISO/8hz-mp3 source code.

なる) 小さな q から +1 ずつ，最大値を超えない最小の q が求められるまで量子化を繰り返す．

この繰返しと，ハフマン符号化したときの利用可能ビット数を超えないための q の値の補正を，あわせて MP3 では内部ループと呼ぶ．その内部ループ処理を図 2 に示す．この方法では時間のかかる量子化処理を何度も繰り返すことが，MP3 エンコーダの速度低下の一因となっていた．

2.2.3 初期量子化ステップの算出

文献 6) は初期量子化ステップを -210 に固定している．これに対して文献 4) と 2) の初期量子化ステップを算出した値は正確ではない．

2.2.4 最適な量子化ステップの探索

文献 4) と 2) は初期量子化ステップとして概算値を用い，二分探索法を使って最適な量子化ステップを探索した．その処理を図 3 に示す．

2.3 ハフマン符号化器

量子化した信号は、20種類以上のハフマン符号から適応的に選択された符号で符号化された後、ビットストリームに組み込まれ、ファイルに書き出される。

3. 高速化方法

3.1 初期量子化ステップの計算方法

量子化は式(1)に基づいて行われる。1グラニューール中の最大値の絶対値を X_{max} 、四捨五入整数化を行っていない量子化値を I_{max}^* 、 X_{max} を量子化した出力を I_{max} 、ハフマン符号化テーブルに存在する最大の値を I_{limit} とすると、以下ようになる。

$$I_{max} = \lfloor I_{max}^* \rfloor = \left\lfloor \left(\frac{X_{max}}{2^{\frac{q}{4}}} \right)^{\frac{3}{4}} - 0.0946 \right\rfloor \quad (2)$$

$$I_{max}^* = \left(\frac{X_{max}}{2^{\frac{q}{4}}} \right)^{\frac{3}{4}} - 0.0946 \quad (3)$$

$$I_{max} \leq I_{limit} \quad (4)$$

式(3)は次のように変形できる。

$$\left(\frac{X_{max}}{2^{\frac{q}{4}}} \right)^{\frac{3}{4}} = I_{max}^* + 0.0946 \quad (5)$$

$$\left(\frac{X_{max}}{2^{\frac{q}{4}}} \right) = (I_{max}^* + 0.0946)^{\frac{4}{3}} \quad (6)$$

ここで、以下のように定数 C を定義する。

$$C = (I_{max}^* + 0.0946)^{\frac{4}{3}} \quad (7)$$

ここで、最適な初期量子化ステップの条件を思い出し、量子化後の最大値が、ハフマン符号化テーブルの最大値と一致するような、量子化ステップが理想であった。そこで、以下 $I_{max} = I_{limit}$ として考える。よって、 C は次のようにも書ける。

$$C = (I_{limit} + 0.0946)^{\frac{4}{3}} \quad (8)$$

I_{limit} は定数であるので、 C も定数として扱えることになる。

さらに、 X_{max} と C を2進浮動小数点形式で表す。2進浮動小数点形式で表したものを以下に示す。

$$X_{max} = X_f \times 2^{X_e} \quad (9)$$

$$C = C_f \times 2^{C_e} \quad (10)$$

ここで、 X_f 、 C_f はそれぞれの仮数部、 X_e 、 C_e はそれぞれの指数部を表す。

式(6)に、式(8)、(9)、(10)を代入すると、次のようになる。

$$\left(\frac{X_{max}}{2^{\frac{q}{4}}} \right) = X_f \times 2^{X_e - \frac{q}{4}} = C_f \times 2^{C_e}$$

$$2^{X_e - \frac{q}{4}} = \frac{C_f}{X_f} \times 2^{C_e}$$

$$2^{-\frac{q}{4}} = \frac{C_f}{X_f} \times 2^{C_e - X_e} \quad (11)$$

さらにここで、次のように定義される関数 g を導入する。

$$\frac{C_f}{X_f} = 2^{g(C_f, X_f)} \quad (12)$$

$$g(C_f, X_f) = \log_2 \left(\frac{C_f}{X_f} \right) \\ = \log_2 C_f - \log_2 X_f \quad (13)$$

式(12)により、式(11)は次のように表せる。

$$2^{-\frac{q}{4}} = 2^{g(C_f, X_f) + C_e - X_e} \quad (14)$$

$$-\frac{q}{4} = g(C_f, X_f) + C_e - X_e \quad (15)$$

$$q = -4(g(C_f, X_f) + C_e - X_e) \quad (16)$$

これが、量子化の逆関数となる。これで、量子化を繰り返す必要はなくなる。しかし、この演算は非常に多くの計算を要する。よって、近似値 q^* を計算することによりさらに計算量の低減を図る。

近似値 q^* の算出方法を示す。いま、2進浮動小数点形式のフォーマットを、指数部が整数、仮数部が0.5以上1.0未満、とすると、以下のような条件が成り立つ。

$$-1 \leq \log_2 C_f < 0 \quad (17)$$

$$-1 \leq \log_2 X_f < 0 \quad (18)$$

$$-1 \leq \log_2 C_f - \log_2 X_f \leq 1 \quad (19)$$

$$-1 \leq g(C_f, X_f) \leq 1 \quad (20)$$

よって、関数 g のとる値の範囲は限られていることが分かる。

また、近似値 q^* は、 q よりも大きくなってはならない。もし、近似値 q^* が、 q よりも小さいとするならば、 q^* によって量子化される値は、ハフマンテーブルに存在しなくなってしまうからである。

加えて、量子化ステップは整数である必要がある。 C_e 、 X_e は整数であることを考えると、式(16)より、近似したときの関数 g の値は0.25の整数倍である。

これらを考え合わせると、

$$g(C_f, X_f) \cong -1 \quad (21)$$

と置くことで、関数 g の計算が不要となる。これを用いて、量子化ステップの近似値 q^* を求めると、式(16)より以下ようになる。

$$q^* = -4(-1 + C_e - X_e) \\ = 4(1 - C_e + X_e) \\ = 4(D + X_e) \quad (22)$$

ここで $1 - C_e$ は定数なので、 D と置いた。この式で必要とされる演算は、1回の固定小数点数の加算とシ

フトのみである。よって、この演算のコストは非常に少ない。指数部の取り出しも、 X が浮動小数点形式で表されていれば、ソフトで行う場合でも、ハードで行う場合でも、まったく問題とはならない。また、近似値の精度であるが、この式を用いると量子化ステップは最大で 8 大きくなる。しかし、図 5 に示した初期量子化ステップと最終的に用いられる最適量子化ステップの差の分布を見て分かるように、正確な初期量子化ステップより 8 大きいとしても特に問題ないことが分かる。ゆえに、この近似は利用できるといえる。

もう少し精度の良い近似を試みる。式 (16) により関数 g を展開すると以下ようになる。

$$q = -4(\log_2 C_f - \log_2 X_f + C_e - X_e) \quad (23)$$

この式の $\log_2 C_f$ について考える。 C は式 (8) で定義されている定数である。よって、 $\log_2 C_f$ は定数であり、式 (23) は、 $\log_2 X_f$ の関数といえる。ここで、

$$E = \log_2 C_e \quad (24)$$

と置くならば、式 (23) は次のように書き改められる。

$$q = -4(-\log_2 X_f + E + C_e - X_e) \quad (25)$$

このときの、2 進浮動小数点形式のフォーマットが先ほどと同様であれば、以下のような条件が成り立つ。

$$-1 \leq \log_2 X_f < 0 \quad (26)$$

ここで、先ほどと同じ条件、近似値 q^* は q よりも大きくなってはならない、および、 $\log_2 X_f$ と E の値が 0.25 の整数倍である必要がある、の 2 つの条件を加えて考えると、

$$\log_2 X_f \geq 0 \quad (27)$$

と置く必要がある。これより、量子化ステップの近似値 q^* を求めると、以下ようになる。

$$\begin{aligned} q^* &= -4(E + C_e - X_e) \\ &= -4(F - X_e) \end{aligned} \quad (28)$$

ここで $E - C_e$ は定数なので、 F と置いた。この式で必要とされる演算は、1 回の固定小数点数の加算とシフトのみである。また、近似値の精度は、この式を用いることにより、量子化ステップが最大で 4 大きくなる、ところまでに抑えることができる。

この近似式を用いて初期量子化ステップを求めることにより、文献 3) の方法に比べ、量子化がまったく必要なくなり、大幅な処理コストの低減を行うことができる。

3.2 非線形量子化に最適化した近似演算方法

非線形量子化の処理は、式 (1) で定義される。この式で、特にコストのかかる部分は次に示す部分である。

$$\left(\frac{|X|}{2^{\frac{q}{4}}}\right)^{\frac{3}{4}} \quad (29)$$

この式を以下の 2 つの式に分けて考える。

$$Y = \frac{X}{2^{\frac{q}{4}}} \quad (30)$$

$$Z = Y^{\frac{3}{4}} \quad (31)$$

3.2.1 Y の算出

まず、式 (30) の計算方法について考える。 X は浮動小数点形式で表され、仮数部 X_f 、指数部 X_e に分けられ、以下のように表せる。

$$X = X_f \times 2^{X_e} \quad (32)$$

これを用いて、式 (30) を書き直すと、

$$\begin{aligned} Y &= \frac{X_f \times 2^{X_e}}{2^{\frac{q}{4}}} \\ &= X_f \times 2^{X_e} \times 2^{-\frac{q}{4}} \\ &= X_f \times 2^{X_e - \frac{q}{4}} \end{aligned} \quad (33)$$

このとき、 q が 4 の倍数であれば、シフトと整数の加算を用いて処理を行える。しかし、 q は任意の整数なのでこのままではいまだコストのかかる処理である。そこで、 $\frac{q}{4}$ を、整数部 $\lceil \frac{q}{4} \rceil$ と小数部 $\lceil \frac{q}{4} \rceil - \frac{q}{4}$ に分けて考える。これを用いると式 (33) は次のようになる。

$$\begin{aligned} Y &= X_f \times 2^{X_e} \times 2^{-\lceil \frac{q}{4} \rceil} \times 2^{-\lceil \frac{q}{4} \rceil - \frac{q}{4}} \\ &= X_f \times 2^{-\lceil \frac{q}{4} \rceil} \times 2^{X_e - \lceil \frac{q}{4} \rceil} \end{aligned} \quad (34)$$

この、式 (34) を見ると、基本的に仮数部の部分と、指数部の部分とに分かれている。指数部の計算である、 $2^{X_e - \lceil \frac{q}{4} \rceil}$ の部分は、出力が浮動小数点形式である限り、単なる加算で計算できる。また、仮数部の計算である $X_f \times 2^{-\lceil \frac{q}{4} \rceil}$ の部分は、 X_f と $2^{-\lceil \frac{q}{4} \rceil}$ をパラメータにして、テーブルを参照するようになれば簡単に近似値を得ることができる。 X_f の任意の精度分のビットをパラメータとして用いることにより任意の精度の近似値を得ることができる。

このままでも、利用可能であるが、より小さいテーブルで効率的に実現できるように改良する。まず、 $2^{-\lceil \frac{q}{4} \rceil}$ のとりうる値を考えると、± 合わせて、8 つの値をとる。その様子を表 1 に示す。このとき、2 の補数形式で表記されているものとする。

表 1 を見ると、 $2^{-\lceil \frac{q}{4} \rceil}$ が、負の値を持つところがある。この負の部分、正の部分と共有できればテーブルを縮小できる。

そこで、 q が正のときに、 $2^{-\lceil \frac{q}{4} \rceil}$ に 1 を加えるような処理を考える。ただし、 q の下位 2 ビットが 00 のときには何も行わない。また、 $2^{-\lceil \frac{q}{4} \rceil}$ に 1 を加えた場合には、指数部から 1 を引き、値を合わせる。この部分は、表 1 のようになる。

この部分を式で表すと以下ようになる。

表 1 $2^{-\frac{q}{4}}$ のとる値
Table 1 The value for $2^{-\frac{q}{4}}$.

q	0...00	0...01	0...10	0...11	1...00	1...01	1...10	1...11
$-\frac{q}{4}$	0	-0.25	-0.50	-0.75	0	0.75	0.50	0.25
負の値に+1		0.75	0.50	0.25				
$2^{-\frac{q}{4}}$	2^0	$2^{-1} \times 2^{0.75}$	$2^{-1} \times 2^{0.50}$	$2^{-1} \times 2^{0.25}$	2^0	$2^{0.75}$	$2^{0.50}$	$2^{0.25}$

$$Y = \begin{cases} X_f \times 2^{(1-\frac{q}{4})} \times 2^{(X_e - [\frac{q}{4}] - 1)} & (q > 0 \text{ かつ } \frac{q}{4} \neq 0) \\ X_f \times 2^{-\frac{q}{4}} \times 2^{(X_e - [\frac{q}{4}])} & (q < 0 \text{ かつ } \frac{q}{4} \neq 0) \\ X_f \times 2^{(X_e - [\frac{q}{4}])} & (\frac{q}{4} = 0) \end{cases} \quad (35)$$

また、逆に q が負のときに、 $-\frac{q}{4}$ から 1 を引くことでも、同様の方法で実現できる。以下については、式 (35) を元に説明を行うが、基本は同じであるので、対応するところを変更することで、どちらにも適用可能である。

ここで、以下のような関数 r を定義する。

$$r = \begin{cases} -(1 - \frac{q}{4}) & (q > 0 \text{ かつ } \frac{q}{4} \neq 0) \\ \frac{q}{4} & (q < 0 \text{ かつ } \frac{q}{4} \neq 0) \\ 0 & (\frac{q}{4} = 0) \end{cases} \quad (36)$$

式 (36) を用いて、式 (35) を表すと以下のようになる。

$$Y = \begin{cases} X_f \times 2^{-r} \times 2^{(X_e - [\frac{q}{4}] - 1)} & (q > 0 \text{ かつ } \frac{q}{4} \neq 0) \\ X_f \times 2^{-r} \times 2^{(X_e - [\frac{q}{4}])} & (q < 0 \text{ かつ } \frac{q}{4} \neq 0) \\ X_f \times 2^{(X_e - [\frac{q}{4}])} & (\frac{q}{4} = 0) \end{cases} \quad (37)$$

このとき、 r は、0、0.25、0.50、0.75、のいずれかの値をとることができる。よって、式 (34) と比べると、テーブルが半分ですむ。テーブルの参照方法は、 X_f と r をパラメータとして使うことにより、実現できる。

次の問題点として、出力を浮動小数点形式とすると、仮数部は正規化されていなければならない。しかし、式 (37) の、 $X_f \times 2^{-r}$ は、必ずしも正規化されたものとは言いえない。ここで、浮動小数点の形式として、仮数部の条件が 0.5 以上 1.0 未満というものを用いたとすると、式 (37) の、 $X_f \times 2^{-r}$ は、0.5 以上、約 1.68 未満となってしまう。

この対策として、正規化を行い、指数部へ補正を行わなければならない。そこで、テーブルに正規化された値と、補正ビットを用意し、もし、補正ビットが立っているなら、指数部に補正を加えることによって、浮

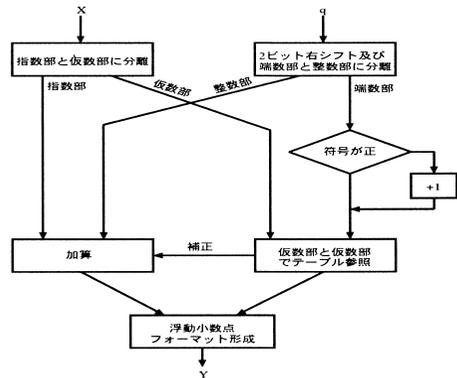


図 4 Y の算出
Fig. 4 The generation of Y.

動小数点形式にのったフォーマットを出力できるようになる。この全体の流れを、図 4 に示す。

この近似値の計算方法では、精度はテーブル参照のために X_f のうち何ビットを用いるか、および、テーブルに格納されている近似値のビット数により定まる。

3.2.2 Z の算出

次に、式 (31) の計算方法について考える。 X と同様に、 Y も浮動小数点形式で表され、仮数部 Y_f 、指数部 Y_e に分けられ、以下のように表せる。

$$Y = Y_f \times 2^{Y_e} \quad (38)$$

よって、 $Y^{\frac{3}{4}}$ は、以下のように表せる。

$$Y = Y_f^{\frac{3}{4}} \times 2^{\frac{3}{4}Y_e} \quad (39)$$

ここで、 $\frac{3}{4}Y_e$ は、次のように、シフトと加算のみで行える。

$$\frac{3}{4}Y_e = Y_e - \frac{Y_e}{4} \quad (40)$$

しかし、 Y_e が 4 の倍数でない限り、 $\frac{Y_e}{4}$ は実数値をとる。そのため、 q のときと同様、整数部と端数部とに分ける。

$$\frac{3}{4}Y_e = Y_e - \left[\frac{Y_e}{4} \right] - \frac{Y_e}{4} \quad (41)$$

$$Z = Y_f^{\frac{3}{4}} \times 2^{-\frac{Y_e}{4}} \times 2^{(Y_e - [\frac{Y_e}{4}])} \quad (42)$$

この式は、式 (34) と同じ形式であり、 Y の導出に用いた方法と同様の手法を利用できる。

3.3 最適な量子化ステップの探索方法

初期量子化ステップは 1 グラニュール中の最大値に

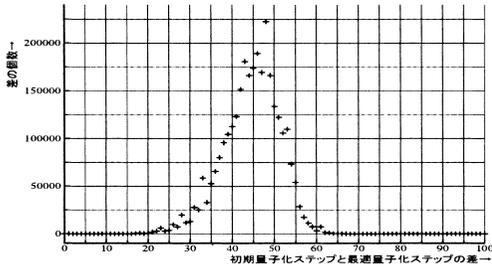


図 5 初期量子化ステップと終了量子化ステップの差の分布
Fig. 5 The histogram of the value between the initial to the last quantization step.

表 2 初期量子化ステップと終了量子化ステップの差の統計
Table 2 The standard deviation and the average between the initial to the last quantization step.

標準偏差	平均値
6.619	44.588

表 3 量子化ステップに関わる変数
Table 3 Important variables.

Q_s	初期量子化ステップ
Q_e	終了量子化ステップ
Q_a	初期量子化ステップと終了量子化ステップの差の平均値
Q_c	中心値 = $Q_s + Q_a$
Q_d	中心値からの偏差量

よって定まる。この最大値はある程度、そのグラニューールのデータの傾向を表しているといえる。よって、初期量子化ステップと、実際に符号化に使用される終了量子化ステップとの間にはある程度の相関性があることが考えられる。

この相関性を見るために、約 100 曲ほどの音楽データをエンコードしたときの、各グラニューールの初期量子化ステップと終了量子化ステップの差をまとめた。ここで用いる初期量子化ステップは、我々が設計したものをういた。文献 4) 等が算出する初期量子化ステップは、正確でないためである。その結果を図 5、表 2 に示す。

このデータから、初期量子化ステップと終了量子化ステップの差は 45 付近に集中して分布することが分かる。この値は用いるデータに左右されるが、同様の傾向は見られるはずなので、目的とする音声データの統計を用いれば同様の手法が用いることができるはずである。

よって、この近辺において一番量子化回数が減少するように設計することにより、量子化回数を削減できる。

ここで、表 3 のように定義する。これらの関係を、

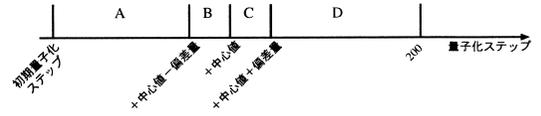


図 6 中心値 Q_c 、偏差量 Q_d 、初期量子化ステップ Q_s の関係
Fig. 6 The relationship of the important variables.

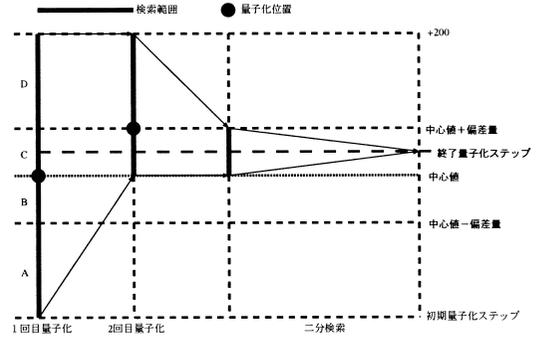


図 7 領域分割による検索範囲の絞り込み
Fig. 7 Example of the search space decomposition.

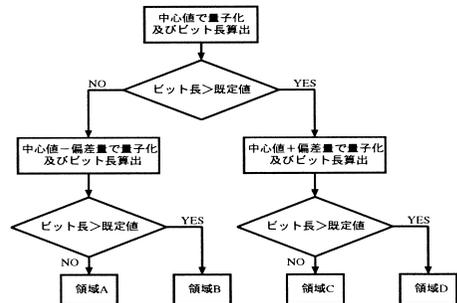


図 8 終了量子化ステップの該当領域の検索
Fig. 8 The flow chart to search the domain.

図 6 に示す。

ここで、仮に $Q_a = 45$ 、 $Q_d = 15$ とすると、図 5、表 2 から、80%程度のデータが、図 6 の BC の領域に入る。

よって、次のように領域を分割して検索することにより、BC 部分の量子化回数を減らし、全体の量子化回数の削減につなげることができる。その流れを図 7 に示す。

まず最初に、図 6 の ABCD のうち、どの領域に Q_e があるかを検索する。このために、 Q_c 、 $Q_c \pm Q_d$ の 2 カ所での量子化を要する。この部分の処理を、図 8 に示す。

このとき、ほとんどの場合、BC のどちらかの領域が選択される。この領域の検索範囲は Q_d となる。 Q_d は、ほぼつねに検索範囲全体の $\frac{1}{4}$ よりも小さいので、より少ない量子化回数で Q_e に達することができる。

AD の領域に Q_e がある場合には、量子化回数が増

えることがあるが、その領域にあることはまれなので全体的に見るならば、十分量子化回数は削減されると考えられる。

4. 性能評価

MP3 エンコーダに対してさまざまな改良を加えて

表 4 1 フレームあたり量子化ルーチンに平均呼び出し回数
Table 4 The number of calls per a frame for the quantization routine.

処理方法	量子化回数
図 2 の処理方法	100
図 3 の処理方法	30
今回提案した方法	9

表 5 全体処理時間の割合と量子化回数の変化
Table 5 The number of calls and the percentage of execution time for the quantization routine per the whole song.

8hz-mp3 の方法		
サンプル	量子化回数	全体処理時間の割合 [%]
ピアノ	440744	28.33
音楽 CD	1152445	29.76
録音テープ	1023362	27.39
英会話教材	507293	29.96
今回提案した方法		
サンプル	量子化回数	全体処理時間の割合 [%]
ピアノ	141751	14.96
音楽 CD	386900	14.86
録音テープ	409680	15.21
英会話教材	160835	14.56

きたが、それらの正当性と性能について評価する。音質の評価に関しては、この論文の付録で述べた方法で行った。また、量子化回数については、gprof を使用して計測した。

4.1 1 フレームあたり平均呼び出し量子化回数

1 フレームあたり平均呼び出し量子化回数については、gprof を使用して計測を行った。比較結果を表 4 に示す。図 2 の処理方法はハフマンテーブルの範囲内にならない最小の量子化ステップから、1 ずつインクリメントしながら量子化を行う。図 3 の処理方法は初期量子化ステップを概算して、二分探索法を用いた量子化ステップを算出を行う。しかし、図 3 では正確に量子化ステップを求めていないので、1 フレームあたり量子化平均呼び出し回数は多くなった。今回提案した方法では正確な初期量子化ステップを求め、音声データの統計をとりその傾向を利用して、最もデータが集中しているところで、検索範囲が狭まり、量子化回数が最小となるように設計を行った。今回提案した方法を利用して、1 フレームあたり量子化ルーチンに呼び出し回数は 9 回までに減らした。

4.2 全体処理時間の割合と量子化回数

今回提案した方法を適用したことによる、量子化回数の変化を表 5 に示す。この結果より量子化回数が大幅に減少していることが分かる。全体処理時間の割合を 15% ぐらいまで減らした。

4.3 処理時間

MP3 のエンコーダの実処理時間を測定した。その

表 6 実処理による高速化の結果
Table 6 The speed up of MP3 encoding with our method.

サンプル	実処理時間		
	ISO/IEC の方法 [s]	今回提案した方法 [s]	処理時間の削減率 [%]
1	337	204	28
2	112	80	29
3	451	338	26
4	995	723	28
5	394	298	25

表 7 今回提案した方法を適用したときの音質
Table 7 The evaluated quality of the encoded music.

	256 kbps		標準				今回提案した方法			
	L	R	L	R	L	R	L	R	L	R
int2	0.85	0.85	1.25	1.26	1.47	1.47	1.18	1.18	1.39	1.39
int	0.84	0.84	1.25	1.23	1.48	1.48	1.18	1.17	1.40	1.40
piano2	0.61	0.62	0.98	0.99	1.61	1.60	1.06	1.07	1.74	1.71
piano	0.44	0.43	0.64	0.63	1.45	1.46	0.64	0.63	1.45	1.47
sph1	0.95	0.94	1.34	1.34	1.41	1.42	1.31	1.31	1.38	1.39
vo2	0.91	0.91	1.34	1.25	1.46	1.37	1.29	1.26	1.41	1.38
wa24	0.70	0.65	0.99	0.94	1.41	1.44	0.99	0.93	1.40	1.42
ys24	0.99	1.00	1.44	1.43	1.45	1.43	1.41	1.42	1.42	1.42
ys25	1.03	1.02	1.64	1.66	1.60	1.62	1.54	1.55	1.50	1.52

結果を表 6 に示す。

4.4 音質の性能評価

今回提案した方法を適用したときの音質がどうかを検証を行った。その結果を表 7 に示す。

ここでは、各音声ファイルの 256 kbps のときの、ツールの評価値を 1 として規格化している。

この結果を見ると、ほぼ同等の音質を得られていることが分かる。若干の差はあるが、図 11 の特性と見比べると、この程度の差では、音質として認識できるレベルではないといえる。よって、今回提案した方法は正常に動作しており、十分実用になると考えられる。

5. ま と め

本論文では、現在において音声圧縮の国際標準規格である MP3 エンコーダの量子化部分を改良した。初期量子化ステップを計算して、さらに大量のデータ（音楽データ、英会話教材、ピアノ、録音テープ）を入力した際の統計的偏りを利用して、最適な量子化ステップ値を探索することができた。結果として、量子化回数を減らすことができ、大きな音質劣化がともなうことはなく、高速なエンコードが可能になった。

参 考 文 献

- 1) Moving Picture Experts Group: MPEG Home page. <http://www.cseit.it/mpeg/>
- 2) International Standards Organization: ISO Web Page. <http://www.iso.ch/>
- 3) ISO/IEC 11172-3: Coding of Moving Pictures and Associated Audio for Digital Strage Media at up to about 1.5Mb/s -part 3: Audio (Aug. 1993).
- 4) <http://www.8hz.com/mp3>
- 5) 貴家仁志, 村松正吾: マルチメディア技術の基礎 DCT (離散コサイン変換) 入門, CQ 出版社 (1997).
- 6) 午後のこ〜だ。
<http://homepage1.nifty.com/herumi/soft.html>
- 7) 酒居, 光成, 成田, 石田, 藤井, 庄司: MP3 エンコーダの高速化, 情報処理学会計算機アーキテクチャ研究会, Vol.2001, No.76, pp.144-25 (2001).
- 8) Sakai, K., Fujiwara, I. and Ae, T.: Extended VLIW Processor for Real-Time Imaging, Real-Time Imaging V, *Proc. SPIE*, Vol.4303, pp.43-50 (2001).

付 録

A.1 音質評価手法

MP3 エンコーダに対して改良を加えたことによる、性能の変化を調べる必要がある。MP3 の場合、プロ

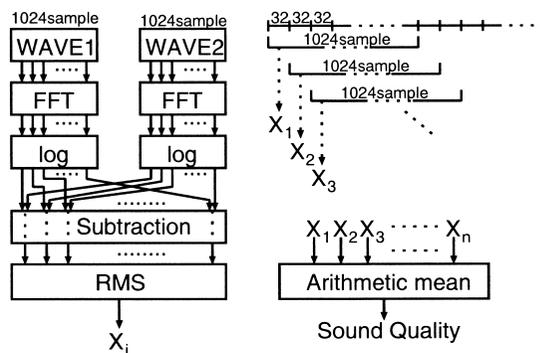


図 9 音質評価プログラムの構成

Fig. 9 The configuration of the quality evaluation program.

グラムの実行速度だけでなく、音質の面でも性能を見なければならぬ。しかし、MP3 の仕様として、デコーダの音質の性能評価方法は compliance test として定められているが、エンコーダについては何も規定がない。

そこで、エンコーダの性能評価に用いることのできるツールを作成した。このツールの行う処理と、用い方を示して行く。

A.1.1 評価処理概要

このツールは、2 つの wave ファイル間の音質差を評価値として出力することができるものである。その方法は 2 つの wave ファイル間の、各周波数成分ごとの差分を、周波数方向と、時間方向に平均をとり、この平均値を評価値として出力するものである。

このプログラムの構成を図 9 に示す。

A.1.2 音質差比較方法

まず、PCM の wave ファイルを 2 つ用意しそれぞれの wave ファイルの先頭から 1024 個の PCM サンプルを取り出す。この取り出したサンプルをそれぞれ、1024 点 FFT にかけて周波数分解を行う。ここで、FFT の出力値の log をとる。これは、小信号時の誤差が大信号時の誤差に飲み込まれないようにするための対策である。

次に、各周波数成分ごとの差を 2 つの wave ファイルの間で求める。この差を、周波数方向に二乗平均をとることにより、周波数方向の平均を出す。これを X_i と置く。

X_i の算出を、wave ファイルの 32 サンプル間隔で、毎回 1024 サンプルずつとりながら行う。この作業を、wave ファイルの終わりまで続ける。その後、 X_1 から X_n までの、相加平均をとり、評価値として利用する。

この評価方法を式で表すなら、次のようになる。

$$t_i, r_i \quad (i = 0, 1, \dots, n)$$

$$T_j = [t_{32j}, t_{32j+1}, \dots, t_{32j+1023}]$$

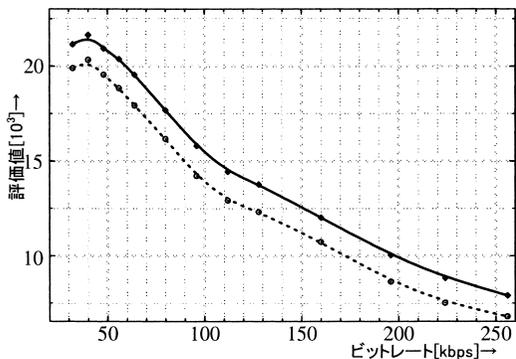


図 10 Compliance test の特性

Fig. 10 The characteristics of the compliance test for the different quality encoding.

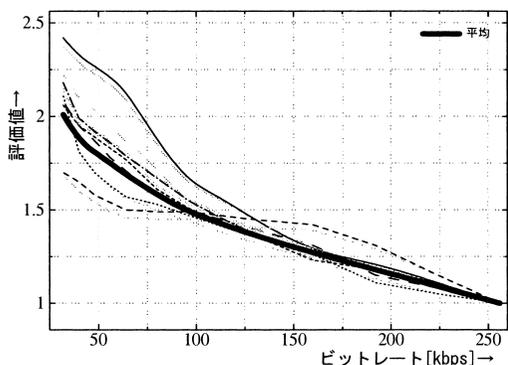


図 11 提案方法の特性

Fig. 11 The characteristics of our method.

$$\begin{aligned}
 R_j &= [r_{32j}, r_{32j+1}, \dots, r_{32j+1023}] \\
 OUTT_j &= FFT(T_j) \\
 OUTR_j &= FFT(R_j) \\
 OUTT_j &= [outt_{j,0}, outt_{j,1}, \dots, outt_{j,1023}] \\
 OUTR_j &= [outr_{j,0}, outr_{j,1}, \dots, outr_{j,1023}] \\
 RMS_j &= \sqrt{\frac{1}{1024} \sum_{k=0}^{1023} \left| \log \frac{outt_{j,k}}{outr_{j,k}} \right|^2} \\
 RESULT &= \frac{32}{n} \sum_{j=0}^{\frac{n}{32}} RMS_j \quad (43)
 \end{aligned}$$

A.1.3 この評価方法の意義

基本的にデコーダの評価方法として定められている，compliance test と同じ手法を用いても，音質の評価を行うことはできるが，音質の評価値が必ずしも音質に比例しない場合があるという問題があった．その結果を，図 10 に示す．図 10 に示した 2 本の線は，2 曲に対して，MP3 のビットレートを変えることにより，音質を劣化させ音質差をつくり，それについて com-

pliance test で評価を行ったものである．320 kbps に対する劣化値を，グラフとして描いている．これを見ると，2 曲に対して極低ビットレートにおいて，評価値が音質に追従していない様子が見てとれる．

この対策として，我々は Compliance test で，時間方向にのみ誤差の平均をとっていたのを，時間方向と周波数方向の誤差の平均をとるようにした．

A.1.4 評価手法の正当性

この評価方法が音質を評価するのに十分な性能を有しているかどうか，その評価を行った．評価には，compliance test の評価に使ったのと同じように，複数の異なった曲に対して，MP3 のビットレートを変えることにより音質を劣化させ音質差をつくり，その音質を測定することで行った．この結果を図 11 に示す．この結果では，どの場合でも評価値が音質に追従している様子を見ることができる．よって，十分音質評価に使えるだけの性能を持っているといえる．

(平成 13 年 12 月 14 日受付)

(平成 14 年 4 月 16 日採録)



李 芝剛

2001 年東海大学工学部通信工学科卒業．同年同大学大学院工学研究科電気工学専攻博士前期課程入学，現在に至る．音声圧縮の研究に従事．



神山 智章

2002 年東海大学工学部通信工学科卒業．同年同大学大学院工学研究科電気工学専攻博士前期課程入学，現在に至る．音声圧縮，回路設計等に関する研究に従事．



清水 尚彦 (正会員)

1985 年上智大学大学院理工学研究科博士前期課程修了．同年 (株) 日立製作所入社，1994 年上智大学大学院理工学研究科博士後期課程修了．博士 (工学)．1995 年より東海大学工学部，現在，東海大学電子情報学部助教授．コンピュータアーキテクチャ，アプリケーション指向アーキテクチャ，オペレーティングシステム等に興味を持つ．電子情報通信学会，ACM，IEEE 各会員．