

ロスレス・オーディオ符号化MPEG-4 ALSの 高性能化

圧縮率の改善のために多チャンネル信号を考慮した新たな線形予測方法を考案しました。また処理時間短縮のためにアルゴリズムの簡略化とソフトウェアの最適化を行いました。評価実験の結果、MPEGの参照ソフトウェアよりも、圧縮率は平均で0.1%改善し処理時間は6倍高速化されました。提案法は国際標準準拠なので実用的です。

かまもと めたか もりや たけひろ
鎌本 優 / 守谷 健弘
はらだ のぼる チャバ コーシュ
原田 登 / Csaba Kos

NTTコミュニケーション科学基礎研究所

MPEG-4 ALSについて

新たな国際標準MPEG-4 Audio Lossless Coding (ALS) が2006年3月に出版されました⁽¹⁾。これは、NTTをはじめベルリン工科大学(ドイツ)、リアルネットワークス社(アメリカ)、インフォコム研究所(シンガポール)らの、技術提案や相互検証などの努力によるもので、国際的な投票を経て制定されました^{(2)~(5)}。この標準はHD-Audio (High-Definition Audio: 高品質音響信号)、つまり、サンプリング周波数が高く、振幅語長が大きく、チャンネル数が多い信号を完全に元どおりに復元できることを保証しながら圧縮するためにつくられました。MPEG-4 ALSは技術が公開になっている国際標準なので、世界トップレベルの性能であるという以外にも、相互運用性、長期保守、特許関係が明快というような、他のフリーソフトなどにはないメリットがあります。他のMPEG標準と同様にビットストリームの意味やデコードおよび再構成の方法はすでに決められています。エンコードする方法にはまだ改善の余地があります。

そこで本稿では私たちが行ってきた、MPEG-4 ALSの枠組み内での圧縮率改善と処理速度向上について説明していきます。

圧縮率を改善させるために、多チャンネル信号向けの新しい線形予測分析方法を考案しました。従来のMCC (Multi-Channel Coding: マルチチャンネル符号化) 法は、親チャンネルと子チャンネルそれぞれのチャンネルごとに独立に線形予測分析された予測残差信号に対して、重みつき減算処理が行われました。新しい分析方法では減算処理後の差信号、つまり符号量が小さくなるように処理されます。言い換えると、従来法によるパラメータは伝送される符号量を最小化していませんでしたが、考案法は実際に送られる符号量を最小化する基準でパラメータを求めます。

処理時間を削減するために、エンコーダの中で、MCC, LTP (Long-Term Prediction: 長期予測), BS (Block-Switching: ブロック分割) の、どの部分でどれだけ時間がかかっているかを調べました。既存のこれらのツールは、圧縮率は十分改善できていましたが、エンコードにかかる演算量もかかっ

ていました。そこで、私たちはMCC, LTP, BSそして他の部分も含め、MPEG-4 ALSの参照ソフトウェアに対してアルゴリズムの簡略化を行いました。また、ソフトウェアの実装についても改善を行い、さらにSSE (Streaming Single instruction multiple data Extensions: ストリーミングSIMD拡張命令) を用いてエンコーダとデコーダの実行速度を向上させました。

ロスレス符号化のフリーソフトは世の中にいくつか出回っており、新たな技術が実装されたりプログラムにバグが見つかったりした場合などは、新たなバージョンがリリースされます。それとは対照的に、国際標準技術は簡単には改定できません。それゆえ、たとえ0.1%でも標準準拠のままで圧縮率を改善させることには意味があります。努力の積み重ねで圧縮限界を超える可能性が生まれます。

簡単なMPEG-4 ALSの基本技術を解説した後で、私たちが行ってきた圧縮率の改善と速度向上について説明し、それから評価結果について示します。

MPEG-4 ALSの基本技術

■線形予測符号化

MPEG-4 ALSの技術は時間領域の線形予測に基づいています。図1はエンコードとデコードの処理を表しています。携帯電話やIP (Internet Protocol) 電話などで使われる音声符号化にも線形予測は使われています。P次の線形予測分析は予測係数 a_k ($k=1, 2, \dots, P$)を算出し、これらの予測係数は入力信号 $x(n)$ と過去のサンプルからの予測信号 $\hat{x}(n)$ との差である予測残差信号 $e(n)$ を最小化します。式で表すと次のようになります。

$$\begin{aligned} e(n) &= x(n) - \hat{x}(n) \\ &= x(n) - \left(-\sum_{k=1}^P a_k \cdot x(n-k) \right) \\ &= \sum_{k=0}^P a_k \cdot x(n-k) \end{aligned}$$

ただしここで、 $a_0=1$ とします。

整数値の予測残差信号と予測係数から変換され量子化されたPARCOR (PARTIAL auto-CORrelation: 偏自

相関) 係数⁽⁶⁾がデコーダへ伝送されます。デコーダは送られてきたビットストリームから、逆フィルタを使って原信号をロスレス (無歪) で再合成し出力します。

予測残差信号の振幅値はゴロム・ライス符号やBGMC (Block Gilbert-Moore Code: ブロックギルバート・ムーア符号) でエントロピー符号化されます⁽⁷⁾。これらの符号は小さな振幅値に短い符号を割り当てています。予測残差信号の振幅は入力信号の振幅よりも小さくなるので、エントロピー符号化がよく効くようになります。また、各フレームの最小符号量は信号のエネルギー最小化で近似できます。

PARCOR係数はフィルタの安定判別に便利な係数で、ゴロム・ライス符号で圧縮されて送られます。予測次数は0 (予測しない) から1023次まで適応的に変化できます。また、頭出しフレームの最初の数サンプルには予測次数漸増方式が使われています⁽⁸⁾。

ここで、この標準準拠のビットスト

リームには、線形予測の分析方法は関係ありません。LD (Levinson-Durbin: レビンソン・ダービン) 法が参照ソフトウェアには実装されていますが、Burg (バーク) 法, Covariance-Lattice (共分散格子) 法⁽⁹⁾ やL-PLP (Laguerre-based Pure Linear Prediction) 法⁽¹⁰⁾ で分析しても標準準拠のままのビットストリームを出力することができます。しかしながら、従来のステレオ線形予測⁽¹¹⁾は根本的に方式が異なるのでMPEG-4 ALSには適用できません。というのも、ステレオ予測ではフィルタの安定性を保証することや次数を適応的に変化させることが困難なので、ステレオ予測の構成は標準には組み込まれませんでした。

■長期予測

上記で述べてきたような通常の線形予測は、近傍の相関を利用する短期の予測です。音声信号やオーディオ信号にはピッチと呼ばれる長期の相関を持つことがあります。この相関は予測残差信号に対して長期予測を適用することにより、振幅値を縮小し、圧縮率を向上させる可能性があります。図2に表されるように、短期予測の残差信号に対して複数タップの長期予測が適用されます。振幅値を縮小するために、最適な時間差とそのときの重み係数が計算されます。重み係数はゴロム・ライス符号化され時間差と共に補助情報として送られます。

式で表すと次のようになります。エンコーダ側では、短期予測後の残差信号 $e(n)$ に対して、さらに長期予測が以下の式で行われます。

$$\tilde{e}(n) = e(n) - \left(\sum_{j=-2}^2 \gamma_j \cdot e(n - \tau + j) \right)$$

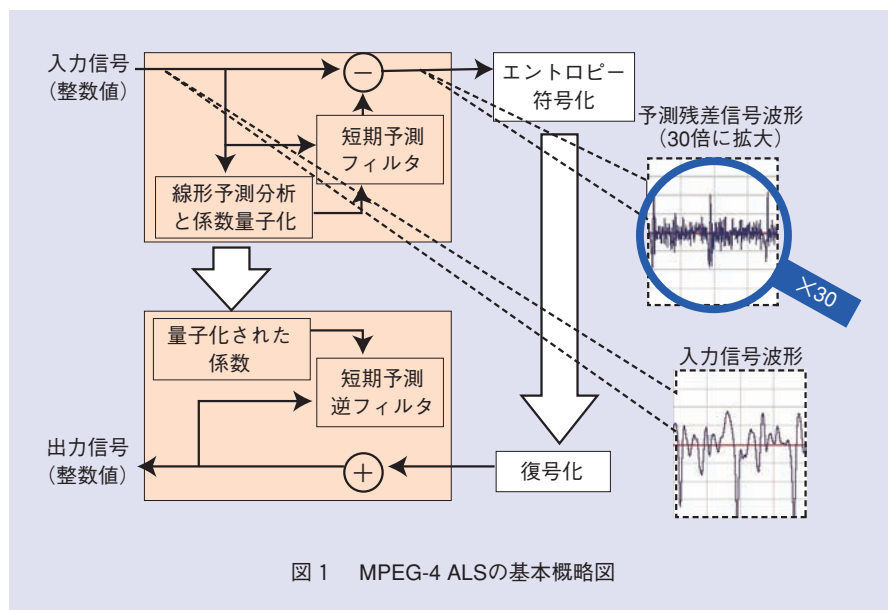


図1 MPEG-4 ALSの基本概略図

ここで、 τ は時間差、 γ_j ($j=-2, -1, \dots, 2$)は重み係数で実際には量子化されています。 $\tilde{e}(n)$ は減算処理後の差信号です。最適な τ や γ_j はそれぞれ自己相関関数や二乗誤差最小化基準などにより求められます。また、差信号 $\tilde{e}(n)$ はその後MCCの入力信号として用いられ、そのまま符号化されて伝送されたりします。

デコーダ側では、逆の演算が行われます。すなわち伝送された $\tilde{e}(n)$ から、

$$e(n) = \tilde{e}(n) + \left(\sum_{j=-2}^2 \gamma_j \cdot e(n - \tau + j) \right)$$

により、元の残差信号 $e(n)$ が復元され、この残差信号と短期の線形予測の再合成に用います。

■マルチチャンネル符号化

多チャンネル信号にはチャンネル間に相関があることが多々あります。チャンネル間予測をすることで、短期予測後や長期予測後の残差信号の振幅値を小さくできる可能性があります。マルチチャンネル予測では、チャンネル間相関を効果的に利用するために、チャンネル間の組合せ（親子関係）を探索します。効率のよい組合せのペア（親子）に対して複数タップの重み付き減算処理が行われます。親子チャンネル間の相対的な時間差を持った重み付き減算処理も追加で行えます。この場合は時間差も補助情報として送られます。いずれにせよ、タップの重み係数はLTPと同様に、量子化され、さらにゴロム・ライス符号化されて送られます。

ロスレス符号化技術は音響信号だけでなく、波面合成信号、生体信号、地震波信号のような多チャンネル時系列信号にも応用される可能性があります。これらの多チャンネル信号を効率よく圧縮するために、親チャンネルに重みをつけて子チャンネルに減算処理を行う方法を適用します。これらの処理は時系列信号の予測残差信号のチャンネル間相関に基づいて行います。この標準では3つのモードがあります。第1はチャンネル間相関を用いない独立符号化モード、第2は時間差なしの3タップの減算処理、第3はさらに時間差付きの3タップ減算処理を追加したモードです。その概念図を図3に示します。ロスレスで復元するために少なくとも1つのチャンネルは独立符号化で伝送されなければなりません。

3タップのモードでは以下のような演算が行われます。

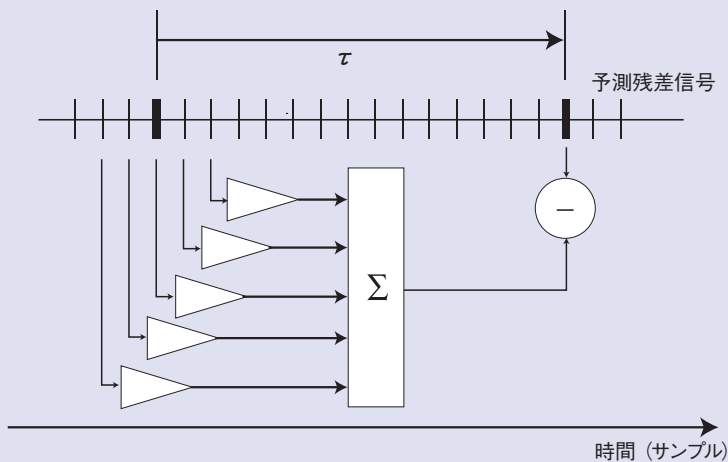


図2 5タップの時間差 τ 付きLTPフィルタ

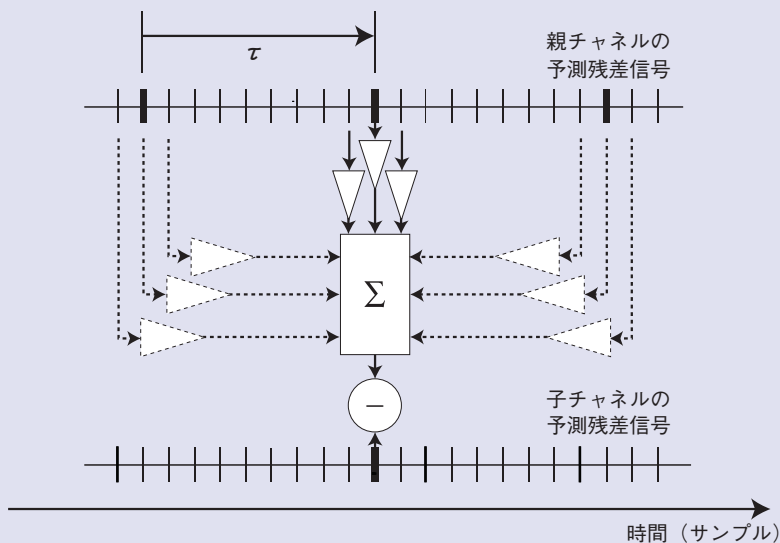


図3 時間差 τ 付きのMCCフィルタ

$$\tilde{e}^c(n) = e^c(n) - \left(\sum_{j=1}^1 \gamma_j \cdot e^r(n+j) \right)$$

ここで、 $\tilde{e}^c(n)$ と $e^c(n)$ はそれぞれ子チャネルの差信号と予測残差信号で、 $e^r(n)$ は親チャネルの予測残差信号です。親チャネルを表す情報と重み係数 γ_j ($j = -1, 0, 1$)が補助情報として送られます。デコーダでは元の予測残差信号が単純に逆演算で求められ、

$$e^c(n) = \tilde{e}^c(n) + \left(\sum_{j=1}^1 \gamma_j \cdot e^r(n+j) \right)$$

となり、復元された $e^c(n)$ は短期予測や長期予測の復元処理に使われます。6タップのモードでは

$$\tilde{e}^c(n) = e^c(n) - \left(\sum_{j=1}^1 \gamma_j \cdot e^r(n+j) + \sum_{j=1}^1 \gamma_{j+\tau} \cdot e^r(n+\tau+j) \right)$$

という演算が行われます。ここで τ は親チャネルと子チャネルの時間差で相互相関関数から推定できます。重み係数 γ_j ($j = -1, 0, 1, \tau - 1, \tau, \tau + 1$)は減算処理後の差信号のエネルギーが最小となるように求められます。これらの時間差と重み係数のパラメータが伝送されます。デコーダではこれまでと同様に逆演算で元の予測残差信号を得られます。

$$e^c(n) = \tilde{e}^c(n) + \left(\sum_{j=1}^1 \gamma_j \cdot e^r(n+j) + \sum_{j=1}^1 \gamma_{j+\tau} \cdot e^r(n+\tau+j) \right)$$

またここで、これまでと同様に、復元された予測残差信号 $e^c(n)$ は短期予測や長期予測の復元処理に使われます。

■階層的ブロック分割

ALSでは圧縮性能改善のために、図

4に表されるような、階層的ブロック分割方法も使用可能です。エンコーダはフレームごとに最適なサブブロックの組み合わせを選択できます。g0モード(ブロック分割しない場合)では1つのブロック長だけしか選べませんが、g1モードでは1つの長いブロック長か2つの短いブロック長かを選ぶことができます。32分割のブロック長がg5モードでは選べます。定常な信号は長いブロック長で有利で、非定常な信号は短いブロック長にすれば圧縮性能を改善できます。

■浮動小数点信号への対応

浮動小数点信号の入力に対しても、ALSでは基本技術である整数値信号の圧縮符号化を利用します⁽¹²⁾。浮動小数点信号はオーバーフローやアンダーフローの危険性がないため、プロのミキシング用途などで用いられます。しかしながら浮動小数点信号は見た目のサンプル値が無相関になるため圧縮が効かなくなります。そこで我々は浮動小数点信号を整数系列と余りの誤差

系列に分解する方法を考案し、圧縮を実現しました。また、元々の入力信号はAD変換後の整数値信号であり、それが何らかの編集をされて浮動小数点信号になることが多々あります。このような場合はフレーム内で見れば整数値が定数倍された信号が浮動小数点信号として表現されているとみなすことができます。この定数を推定し、割り算することで、ほぼ整数値信号として扱えるようになり、圧縮性能が改善します。誤差系列の部分はマスクLempel-Ziv符号化で効率よく圧縮できます。

このように、標準にはさまざまな符号化パラメータが存在します。最適なパラメータを求めれば、標準準拠のままに圧縮性能をさらに向上させることができます⁽¹³⁾。

MPEG-4 ALSの高性能化

■マルチチャネル符号化に対応した線形予測分析

これまでの線形予測分析法は、図5のように、予測残差のエネルギーがチャ

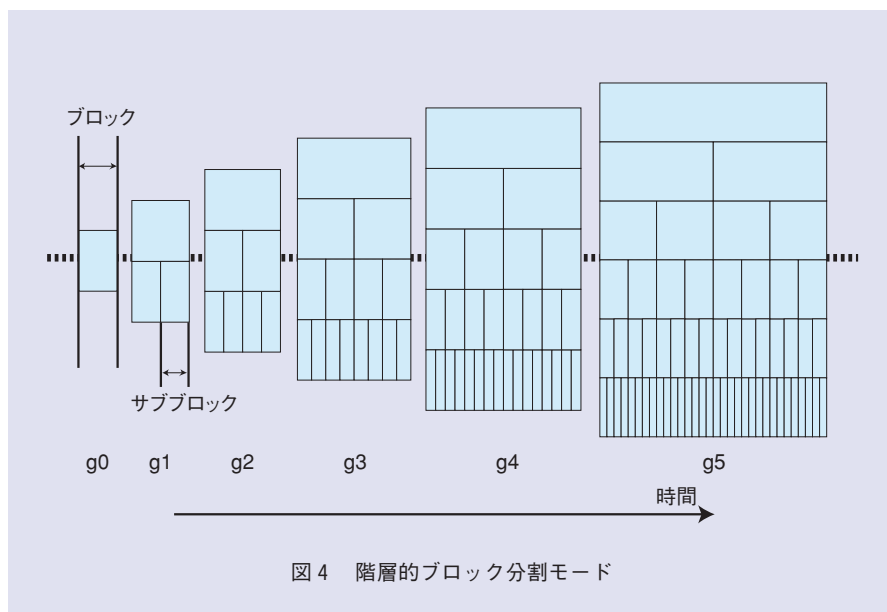


図4 階層的ブロック分割モード

ネルごとに最小となるように線形予測係数を分析していました。つまり、親チャネルの線形予測係数 a_k^r ($k=1, 2, \dots, P^r$)と子チャネルの線形予測係数 a_k^c ($k=1, 2, \dots, P^c$)は予測残差信号のエネルギー $E^r = \sum_{n=1}^N (e^r(n))^2$ と $E^c = \sum_{n=1}^N (e^c(n))^2$ のそれぞれが、個々に最小となるようにして求められていました。ただしここでNは1フ

レーム当りのサンプル数です。しかしながら、マルチチャネル符号化を使う場合には子チャネルは減算処理後の信号が送られるので、図6のように子チャネルでは減算処理後のエネルギー $\tilde{E}^c = \sum_{n=1}^N (\tilde{e}^c(n))^2$ が最小となるような基準で分析したほうが、圧縮率が改善できる可能性があります。というのも、伝送されるのは $\tilde{e}^c(n)$ であって

$e^c(n)$ ではないからです。言い換えると、子チャネルの線形予測残差信号が、親チャネルの線形予測残差信号にできるだけ似るように子チャネルの線形予測係数を求めます。

ここでは、分かりやすく説明するために時間差のない1タップの減算処理とします。すると、

$$\tilde{e}^c(n) = e^c(n) - \gamma \cdot e^r(n) \quad (1)$$

となり、 \tilde{E}^c を最小化するために γ は

$$\gamma = \frac{\sum_{n=1}^N e^c(n) \cdot e^r(n)}{\sum_{n=1}^N e^r(n) \cdot e^r(n)} \quad (2)$$

として求められます。式(1)と(2)より、 \tilde{E}^c は次のように書き換えられます。

$$\begin{aligned} \tilde{E}^c(n) &= \sum_{n=1}^N (\tilde{e}^c(n))^2 = \sum_{n=1}^N (e^c(n) - \gamma \cdot e^r(n))^2 \\ &= \sum_{n=1}^N \left(\sum_{k=0}^{P^c} a_k^c \cdot x^c(n-k) \right)^2 \\ &\quad - \frac{\left(\sum_{n=1}^N e^r(n) \cdot \sum_{k=0}^{P^c} a_k^c \cdot x^c(n-k) \right)^2}{E^r} \end{aligned} \quad (3)$$

この式(3)を最小化するような線形予測係数 a_k^c ($k=1, 2, \dots, P^c$)は偏微分

$$\frac{\partial \tilde{E}^c}{\partial a_i^c} = 0 \quad (i=1, 2, \dots, P^c)$$

となるような係数として求めればよく、正規方程式

$$\mathbf{a} = \mathbf{v}^{-1} \cdot \mathbf{w} \quad (4)$$

を解くことで求められます。ここで、

$$\mathbf{a} = (a_1^c, a_2^c, \dots, a_{P^c}^c)^T$$

$$\mathbf{v} = \begin{pmatrix} R(0) - \frac{C(1,1)}{E^r} & \dots & R(P^c-1) - \frac{C(1,P^c)}{E^r} \\ \vdots & \ddots & \vdots \\ R(P^c-1) - \frac{C(1,P^c)}{E^r} & \dots & R(0) - \frac{C(P^c,P^c)}{E^r} \end{pmatrix}$$

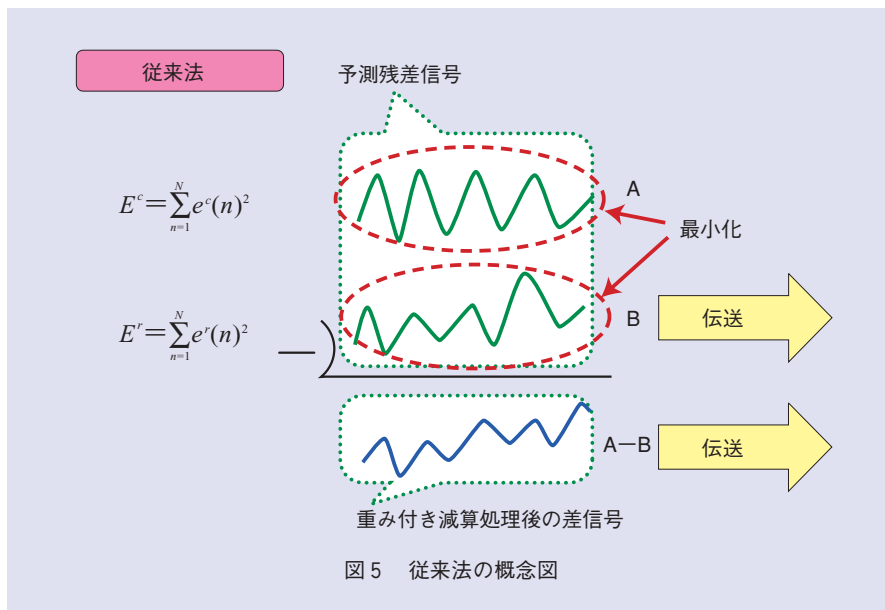


図5 従来法概念図

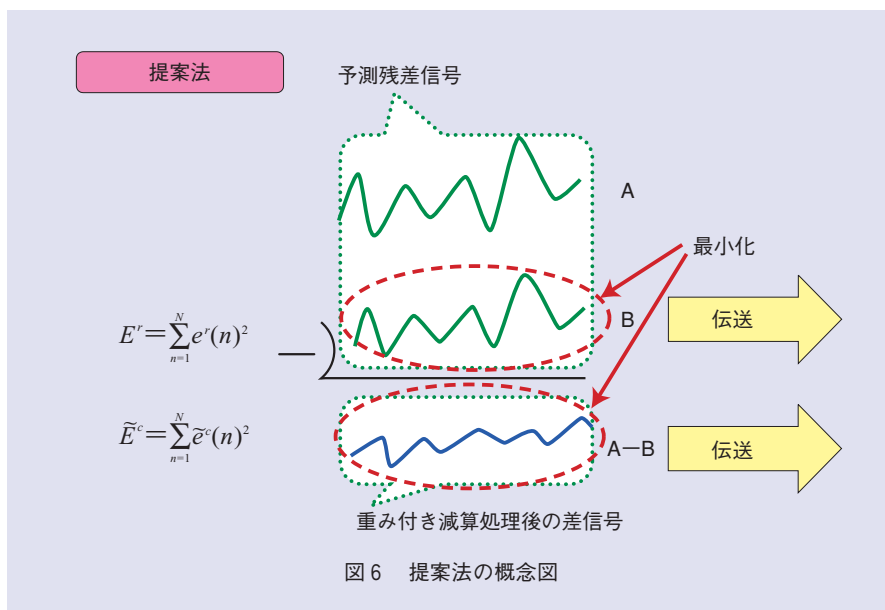


図6 提案法概念図

$$w = \begin{pmatrix} R(1) - \frac{C(0, 1)}{E^r} \\ \vdots \\ R(P^c) - \frac{C(0, P^c)}{E^r} \end{pmatrix}$$

$$R(\tau) = \sum_{n=1}^N x^c(n) \cdot x^c(n-\tau)$$

$$c(\mu, \nu) = \left(\sum_{n=1}^N e^c(n) \cdot x^c(n-\mu) \right) \cdot \left(\sum_{n=1}^N e^c(n) \cdot x^c(n-\nu) \right)$$

となります。

式(4)のVは対称行列ですがテプリック型ではないので、コレスキー分解で解くことになります。この方法により求めた線形予測係数を用いたフィルタはときどき不安定になることがあります。その場合は、 E^c を最小化する通常の従来法(LD法)を適用することにして、提案法をあきらめます。また、従来法で生成される符号量と提案法で生成される符号量を比較して、小さいほうを伝送します。これにより、圧縮性能は必ず改善される、つまり、悪くはならないことが保障されます。

演算量の削減

(1) 自己相関関数の計算

自己相関関数は短期の線形予測分析に必要です。予測次数や分析フレームのサンプル数が増加すると、計算量も増加します。予測次数が大きいときには、演算量削減のために、FFT(Fast Fourier Transformation: 高速フーリエ変換)を用います。同様に、LTPを適用する際にもっとも時間がかかる部分は時間差を求める際の自己相関関数の計算です。192 kHz以上のサンプリングレートの信号では、短期予測の最大次数に1024を足した範囲で最適な時間差を探します。このような環境

ではFFTにより自己相関関数の計算は大幅に処理時間が削減されます。

(2) マルチチャンネル符号化の簡略化

MCCのエンコード・デコード処理は65536チャンネルまで対応できるように設計されています。一般的にチャンネルの親子関係の探索には時間がかかります。音楽用途では2チャンネルのステレオ信号が重要です。ステレオ信号に限って言えば、単純に両方のエネルギーを比較して、エネルギーの小さいチャンネルを親チャンネルにすれば良いのです。また、マイクロホンアレー信号であれば時間差の付いた6タップの処理が有効ですが、ステレオ信号では3タップの減算処理で十分な性能が得られます。

(3) ブロック分割の簡略化

最適なサブブロックを見つけるには、実際の符号量を比較する必要があります。ブロックの階層に応じて計算量も増加します。そこで、演算量削減のために、中間状態での計算結果から符号量を推定し、近似解でサブブロックを決めることにしました。

評価実験

■実験条件

これまで述べてきた改善策を評価するために、音楽信号による実験を行いました。CD(Compact Disc)品質の著作権フリー(著者Y.K.が作曲)の13曲(ステレオ、サンプリングレート44.1 kHz、振幅語長16 bit、演奏時間64分、ファイルサイズ642 MB)と、RWC音楽データベース⁽¹⁴⁾の中から、英語の歌詞の20曲(RWC-MDB-P-2001 No.81-100, 85分, 855 MB)を用いました。これらは市販されているポピュ

ラー音楽とほぼ同等です。エンコードやデコードにかかる時間は“timeit.exe”により、全ファイルのトータルの時間を計測しました。速い方が高性能です。処理時間は演奏時間で正規化しました。50%は実時間の2倍速を意味します。Microsoft Windows Server 2003, AMD Opteron プロセッサ2.4 GHz, メモリ2 GBを搭載した計算機を用いました。また圧縮率(小さいほうが高性能)はすべてのファイルの合計を用いて、

圧縮率(%) =

$$\frac{\text{圧縮後のファイルサイズ}}{\text{元のファイルサイズ}} \times 100$$

と決めました。

■結果

実験結果を以下にまとめています。

MCCにおけるエンコード時間と圧縮率の関係を図7に示しています。図中の点は上から下へ、異なる最大予測次数7, 15, 31, 63, 127, 255を表しています。私たちの改善したソフトウェア(赤線: NTT with MCC)は参照ソフトウェア(青線: RM with MCC)よりエンコード時間を約6分の1に削減できています。また、新しい線形予測分析(ピンク線: NTT with MCC and new predictor)により、従来のMCCよりも圧縮率を0.1%改善できています。

デコーダでは図8にあるように、参照モデル(青線: RM with MCC)よりも効果的な実装のおかげでNTT版(赤線: NTT with MCC)は処理時間を約半分に節約できています。さらにMCCを考慮した線形予測分析を用いた方法(ピンク線: NTT with

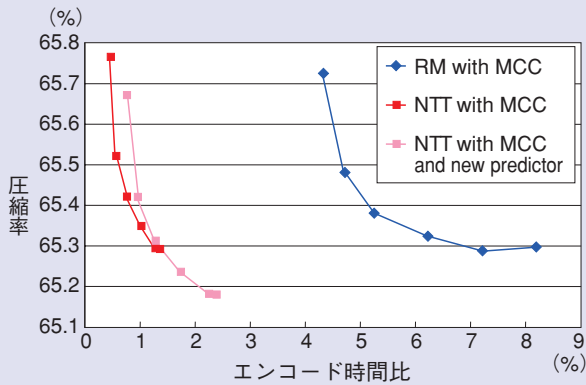


図7 MCCを用いたときのエンコード処理時間と圧縮率の比較 (左が速く, 下が小さいファイルサイズ)

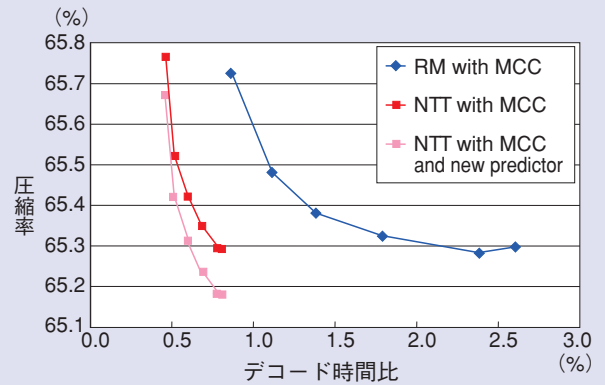


図8 MCCを用いたときのデコード処理時間と圧縮率の比較 (左下が高性能)

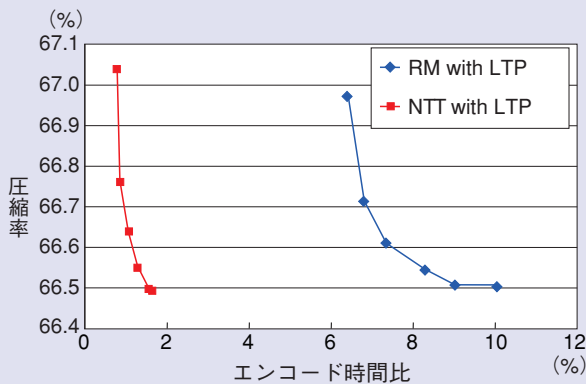


図9 LTPを用いたときのエンコード処理時間と圧縮率の比較 (左が速く, 下が小さいファイルサイズ)

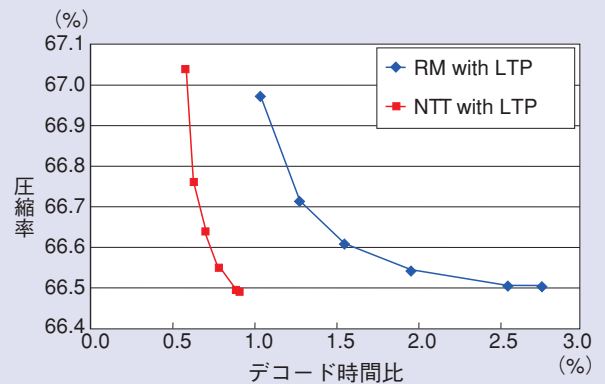


図10 LTPを用いたときのデコード処理時間と圧縮率の比較 (左下が高性能)

MCC and new predictor) では, 従来のNTT版 (赤線) と比べて, 時間を増やさずに, 圧縮率が改善できています。言い換えると, 新手法で生成されたビットストリームは, 同じ圧縮率であれば, より少ない予測回数になるので, 速くデコードできることとなります。

LTPにおけるエンコード時間の比較を図9に示します。簡略化されたエンコーダ (赤線: NTT with LTP) は参照モデル (青線: RM with LTP) よりも約5分の1の速さで処理ができ

ています。デコード速度は図10に示します。こちらは参照モデル (青線: RM with LTP) よりもNTT版 (赤線: NTT with LTP) が最適化された実装により約2倍速で処理できています。

ブロック分割を用いたときのエンコード時間と圧縮率の比較を図11に示します。私たちの改善したソフトウェア (ピンク線とオレンジ線: NTT with BS level 1 および2) は参照ソフトウェア (水色線と緑線: RM with BS level 1 および2) よりエンコード時間を約4

分の1に削減できています。分割レベルが増えるとその差は大きくなっていきます。デコーダでの結果は図12にあります。NTT版 (赤線, ピンク線, オレンジ線: NTT) は参照モデル (青線, 水色線, 緑線: RM) よりデコードの処理時間を約3分の1に短縮できています。

まとめると, 私たちによって改善されたNTT版のソフトウェアはMPEG-4 ALSに準拠したビットストリームを出力します。さらに図7~12にあるように, 参照ソフトウェアよりも短い時間

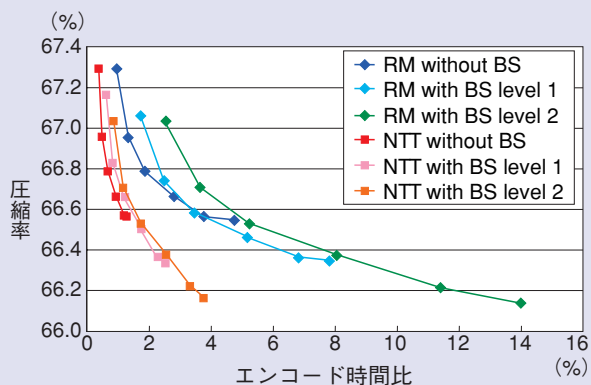


図11 BSを用いたときのエンコード処理時間と圧縮率の比較 (左が速く、下が小さいファイルサイズ)

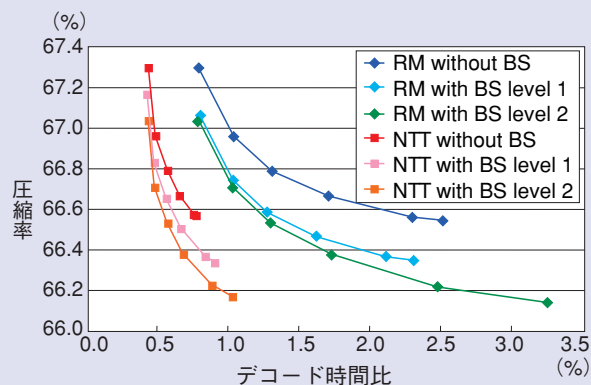


図12 BSを用いたときのデコード処理時間と圧縮率の比較 (左下が高性能)

でエンコード・デコードの処理ができています。

まとめ

私たちはMPEG-4 ALSに準拠したソフトウェアの改善を行いました。評価実験の結果、MPEGの参照ソフトウェアよりも約6倍速いエンコード処理をしながら圧縮率を0.1%改善することができました。デコーダでは効果的なソフトウェアの実装により、処理時間を約半分に短縮できました。このNTT版ソフトウェアは、国際標準準拠のビットストリームを入出力できるので実用的です。この標準はさまざまなアプリケーションに利用されることが期待できます。また、圧縮されたファイルが100年後でも復元できるようにメンテナンスを続けていかねばなりません。音響信号の蓄積、プロ用の編集、携帯プレーヤ以外にも、医療分野や環境分野への応用も考えられます。

参考文献

(1) ISO/IEC 14496-3:2005/Amd.2:2006, Information technology—Coding of audio-visual objects—Part 3: Audio, Amendment 2: Audio Lossless Coding (ALS), new audio profiles and BSAC extensions, edition 2006-03-

15.
 (2) T. Moriya, N. Harada, Y. Kamamoto, and H. Sekigawa: "MPEG-4 ALS—International Standard for Lossless Audio Coding," NTT Technical Review, Vol.4, No.8, pp.40-45, 2006.
 (3) 鎌本・守谷・西本・嵯峨山: "チャンネル間相関を用いた多チャンネル信号の可逆圧縮符号化," 情処学論, Vol.46, No.5, pp.1118-1128, 2005.
 (4) 鎌本・守谷・原田・西本・嵯峨山: "ISO/IEC MPEG-4 Audio Lossless Coding (ALS) におけるチャンネル内とチャンネル間の長期予測," 信学論, Vol.J89-B, No.2, pp.214-222, 2006.
 (5) 原田・守谷・関川・白柳・鎌本: "ISO/IEC MPEG-4 Audio Lossless Coding (ALS) におけるIEEE 754浮動小数点信号の可逆符号化," IEICE, Vol.J89-B, No.2, pp.204-213, 2006.
 (6) J. Makhoul: "Linear prediction: A tutorial review," Proc. of IEEE, Vol.63, No.4, pp.561-580, 1975.
 (7) T. Liebchen and Y. Reznik: "MPEG-4 ALS: an emerging standard for lossless audio coding," Proc. of Data Compression Conference 2004, pp.439-448, 2004.
 (8) T. Moriya, D. Yang, and T. Liebchen: "Extended Linear Prediction Tools for Lossless Audio Coding," Proc. of ICASSP 2004, pp.III-1008-1011, 2004.
 (9) J. Makhoul: "Stable and efficient lattice methods for linear prediction," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. 25, No.5, pp.423-428, 1977.
 (10) A. Biswas and B. den Brinker: "Lossless compression of digital audio using Laguerre-based pure linear prediction," Proc. SPS 2004 (4th IEEE Benelux Signal Processing Symp.), pp.49-52, 2004.
 (11) F. Ghido and I. Tabus: "Adaptive design of the preprocessing stage for stereo lossless audio compression," AES 122nd Convention Paper, #7085, 2007.
 (12) N. Harada, T. Moriya, H. Sekigawa, and K. Shirayanagi: "Lossless compression of IEEE floating-point audio using approximate

common factor coding and masked-LZ compression," AES 118th Convention Paper, #6352, 2005.
 (13) T. Moriya, N. Harada, and Y. Kamamoto: "Performance-complexity tradeoffs of the MPEG-4 ALS lossless coding standard," Proc. of IEEE 40th Asilomar conference, WA7a-4, 2006.
 (14) M. Goto: "Development of the RWC music database," Proc. of ICA2004 (18th International Congress on Acoustics), pp.I-553-556, 2004.



(左から) 鎌本 優 / 守谷 健弘 / 原田 登 / Csaba Kos

音響信号だけでなく生体信号(脳波や心電図など)や環境信号(地震波やセンサーなど)も圧縮して元どおりに復元できる技術として広まっていくことを期待しています。より広い用途に適用可能な方法を考案していきます。

◆問い合わせ先

NTTコミュニケーション科学基礎研究所
 守谷特別研究室
 TEL 046-240-3593
 FAX 046-240-3145
 E-mail kamamoto.yutaka@lab.ntt.co.jp