

F-17

音響信号のサンプリングレート・スケーラブル・ロスレス符号化

守谷健弘 神明夫 池田和永 森岳至 金子孝夫
 Takehiro Moriya Akio Jin Kazunaga Ikeda Takeshi Mori Takao Kaneko

1. まえがき

ブロードバンドサービスが普及し始めて、高いサンプリングレート (96 kHz) や高精度の振幅語長 (24 bit) の音楽信号の歪のない蓄積や配信の要望が高まっている [1]。一方で、これまでのフォーマット (48 kHz, 16 bit) や MPEG-4 などの高圧縮の符号化の要請も大きい。本稿ではサンプリングレートと振幅語長を階層的に分解し、さらに歪を許す高圧縮符号化を含む、スケーラブルなロスレス符号化を提案し、実用的な配信システムでの利点を定量的に検証する。

2. 符号化

2.1 スケーラビリティ

ビットレートやサンプリングレートのスケーラビリティは MPEG-2 や MPEG-4 [2] ですでに標準化されているように、圧縮ビット列の一部だけを取り出しても意味のある信号が再生できる機能で、通常階層的符号化が使われる。例えば、復号側 (利用者) の要求に合わせた部分的な復号が可能となり、また異なる階層の情報毎に別のクラスの packets を割り当てることができれば、ネットワークの要求に合わせた情報伝送容量の制御が可能となる。複数の符号化データを用意したり、トランスコーディングする必要がなくなる。

2.2 階層符号化の枠組み

本稿ではサンプリングレート、振幅語長の両方を考慮したスケーラブルな歪のない符号化を提案する。図 1 はその構成例として 4 段階の階層符号化を示す。まず原音は 96 kHz サンプル、振幅語長 24 bit であり、次の段階で振幅語長を 16 bit とする。下位 8 bit に対応する情報を符号化する。この場合は雑音に近く、圧縮効率は低いので圧縮はしない。次にダウンサンプルにより 48 kHz サンプルの信号を作り、同時にこれをアップサンプルしてダウンサンプル前との誤差信号を作成する。この誤差信号は 96 kHz サンプルの時間領域の信号であるが、周波数領域の高域半分のみ成分しか持っていないので圧縮符号化が可能である。同時に 48 kHz サンプルの信号を原音として MPEG-4 の高圧縮符号化 (この場合 TwinVQ) と、その差分を符号化する。これで 4 種類の独立の圧縮データ系列が作成できる。なお図 1 中のビットレートは表 1 と同じレートを書き込んだものである。

復号側あるいは伝送時には、図 1 の 4 つの系列の下から優先的に利用することで 4 種類の再生波形が得られる。一番下のみでは高圧縮符号化、2 番目まであわせると 48 kHz, 16 bit の原音、3 番目まで使うと 96 kHz, 16 bit の原音、すべてを使うと 96 kHz, 24 bit 原音が再構成できる。

2.3 圧縮符号化の構成

上記の階層符号化では上位の信号との誤差信号を符号化するが、いずれも振幅は原音より小さくなることを利

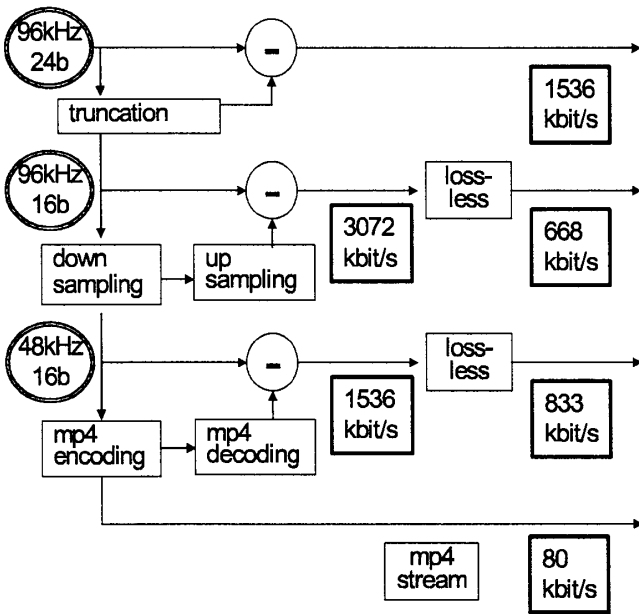


図 1 階層符号化の構成

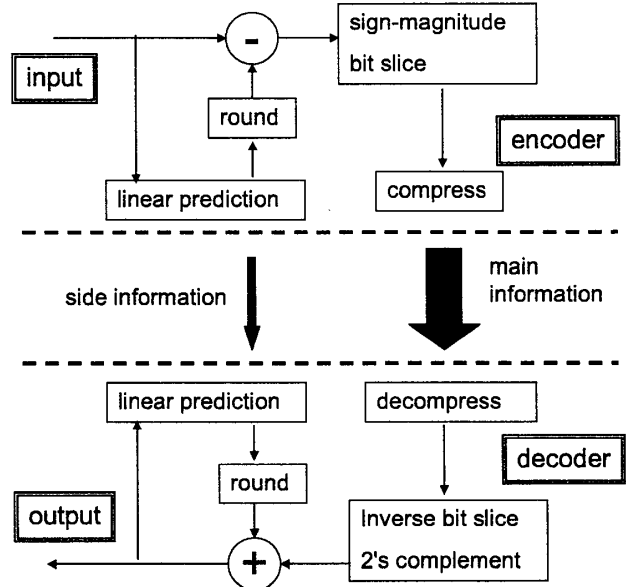


図 2 基本圧縮符号化

用して、圧縮効率をあげることが可能である。また誤差信号は必ずしも周波数領域で平坦ではなく波形サンプル間の相関があるので線形予測によって平坦化し、振幅を小さくすることが可能である。これらを考慮した基本圧縮符号化を図2に示す。

線形予測は音声符号化でよく使われる全極型の20次の予測を行い、予測係数はPARCOR係数で量子化し、補助情報として伝送している。予測係数や予測値は実数となるが、予測誤差を算出する際には予測値を四捨五入して整数化する。

入力信号より振幅が小さくなった予測誤差信号について、2の補数表示から、極性絶対値表示の数値に変換する[3][4]。これにより、振幅を表す上位の桁の2進数値は“1”よりも“0”が多くなる。さらに時間領域の振幅ビット列を時間軸でならべ、水平方向にビットをスライスしてバイト単位のシンボルとして切り出す。これにより上位の桁(MSBより)に対応するシンボルの大部分は“0”になり、“0”の連続であることを指定するといった簡単な符号化で圧縮可能である。下位の桁(LSBより)に対するシンボルはランダムに近く、ほとんど圧縮できない。今回の数値実験ではこの水平方向にスライスしたシンボルにZIPを適用している。

復号化処理は符号化処理の逆操作となり、符号器と同じ予測と四捨五入により歪のない信号が再構成される

3. 圧縮性能の比較

表1は左端の欄のサンプリングレートと語長の条件を入力としたときの圧縮ビットレートを表している。図1のブロック図とも対応しており、上から3つはすべてロスレス符号化であり、4番目は歪のあるMPEG-4の圧縮符号化である。またすべて入力はステレオ信号で、2種類の音楽に対する平均値を表示している。右端は原音のビットレート、右から2番目は各条件毎に独立にスタンドアロンの符号化で圧縮した場合のビットレート、右から3番目は図1に示される3段階の階層をもつスケラブル符号化での圧縮ビットレートである。いずれも基本圧縮ツールとして図2の符号化を使っている。実はこの数値は右から4番目のスケラブル符号化の差分ビットレートを累積した数値となっている。

表1 圧縮性能の比較

	differential data rate (kbit/s)	accumulated data rate (kbit/s)	stand-alone compression (kbit/s)	original data rate (kbit/s)
96k-hz, 24bit	1536	3117	2668	4560
96k-hz, 16bit	668	1581	1420	3072
48k-hz, 16bit	833	913	881	1536
lossycoding	80	80	80	80
total	-	3117	5177	9248

ここで、利用者の好みやネットワーク環境に応じて各種条件の音楽を提供できるサーバー・クライアント型の蓄積配信システムを想定してこの表を解釈する。そのシステムのサーバー側での蓄積方法として下記の3種類の方法を考える。なお②、③は従来法と位置づけられる。

- ① 本稿で提案するスケラブル符号化
- ② すべての条件の波形を予め作って個別に圧縮しておき、要求に応じて選択する方式、
- ③ 最高のサンプリングレートと最大の語長の信号のみを圧縮しておき、要求があるたびに伸張、サンプリング・語長変換、再圧縮を行う。

①と③の場合のサーバー側での蓄積容量の違いは、最高のサンプリングレートと語長条件に対する累積ビットレート(右から3列目)とスタンドアロン符号化のビットレート(右から2列目)で比較できる。スタンドアロンの場合が圧縮効率が高いが、10%程度の差である。次に①と②の場合の蓄積要領を比較する場合、この表の3種類の歪なしすべてを加算したものとなる。ただしスケラブル構成の場合は最高サンプリングレートと語長の圧縮ビット列が下位のサンプリングレートと語長の符号化を含むので、最高サンプリングレートと語長に必要なビットレートのみで、すべてに対応できる。一方スタンドアロンの場合はすべて独立に圧縮ファイルが必要となる。これは表1の一番下の総計で示され、②の場合は、①のスケラブル符号化の場合のより60%も多くなる。

また、サーバーの演算量の観点からは①と②は選択だけであるのに対し、③は非常に大きな演算量を必要とする。①の場合、クライアント側での再構成処理は②、③の場合より少し負荷がかかるが、大きな差とはならない。

このように、蓄積容量と演算量を総合的に比較すると提案するスケラブル符号化が優れており、もちろん符号化条件が192kHzまでに拡張されると、この違いは非常に大きくなる。またスケラブル符号化の各階層の情報を別のカテゴリのパケットに対応させることで、ネットワーク容量の変動に応じて、ルータによるパケットの取捨選択が可能となる。

4. まとめ

サンプリングレートと振幅の表現語長のスケラブルなロスレス符号化の構成を提案し、サーバー・クライアント型の蓄積配信システムでの優位性を具体的に示した。ブロードバンドの利用した高付加価値の配信で、かつ従来の利用者や多様なネットワーク環境やその時間的変化にも柔軟に対応できる符号化の枠組みとして有望である。

参考文献

- [1] M. Hans and R.W. Schafer, "Lossless Compression of Digital Audio," *IEEE Signal processing magazine*. Vol.18 No.4, pp.21-32, 2001.
- [2] ISO/IEC 14496-3; 2001 (MPEG-4 Audio)
- [3] T. Moriya, N. Iwakami, T. Mori and A. Jin, "A Design of Lossy and Lossless Scalable Audio Coder," *Proc. ICASSP'2000*, pp. AE-P1.11, 2000.
- [4] T. Moriya, A. Jin, T. Mori, K. Ikeda, and T. Kaneko, "Lossless scalable audio coder and quality enhancement," *Proc. ICASSP'2002*, #2440, 2002.