

MPEG-2 ビットストリーム分離・合成機能を用いた スケーラブル映像符号化方式の検討

永吉 功[†] 花村 剛^{††} 笠井 裕之^{†††} 富永 英義^{†††}

[†] 早稲田大学 理工学部 電子・情報通信学科
〒169-8555 東京都新宿区大久保 3-4-1

^{††} 株式会社 メディアグルー
〒169-0072 東京都新宿区大久保 2-4-12 新宿ラムダックスビル 8 階

^{†††} 早稲田大学 国際情報通信研究センター
〒169-0051 東京都西早稲田 1-3-10

isao@tom.comm.waseda.ac.jp

本稿では、要求品質に対して柔軟な映像提供を可能にする MPEG-2 トランスコーダに着目し、トランスコーダの入力/出力間の差分映像信号を保存し合成することで符号化器/復号器の外部機能によるスケーラブル映像符号化を実現するための新たな符号化方式について検討する。まず、符号化器/復号器によるスケーラビリティの実現方法における問題点について述べ、トランスコーダの入力/出力間の差分映像信号の符号化機能を追加することで階層符号化と同等の機能を実現する方法を述べる。次に、再量子化処理前後での DCT 係数ブロックの変化が再量子化出力係数値から類推可能な点に着目し、係数の変化差分値の分類による変化成分情報の符号化方式と、本符号化情報を利用した再量子化前の係数情報の生成方式を提案する。さらに、評価実験より、提案方式は分離前後で総符号量が大きく増加することなく、場合によっては再圧縮する効果をもたらす、効率的な符号化方法であることを示す。また本提案方式の符号化性能について理論的に考察し、圧縮符号化の一つの方式としても有効であることを述べる。

キーワード スケーラブル映像符号化, トランスコーダ, ビットストリーム分離・合成, MPEG-2

A Study on the Scalable Coding Algorithm by Separating and Composing of MPEG-2 Bitstream

Isao NAGAYOSHI[†] Tsuyoshi HANAMURA^{††} Hiroyuki KASAI^{†††} Hideyoshi TOMINAGA^{†††}

[†] Dept. of Electronics, Information and Communication Engineering, WASEDA University
3-4-1 Ohkubo Shinjuku-ku, Tokyo, 169-8555 JAPAN

^{††} Media Glue Corporation
Ramdax Bldg. 8th floor 2-4-12 Ohkubo Shinjuku-ku, Tokyo, 169-0072 JAPAN

^{†††} Global Information and Telecommunication Institute, WASEDA University
1-3-10 Nishi-Waseda Shinjuku-ku, Tokyo, 169-0051 JAPAN

isao@tom.comm.waseda.ac.jp

In this paper, we focus on the MPEG-2 transcoder to provide the flexible video transmission by converting the compressed video bit streams, and study on the new coding scheme for realizing bit stream scalability by preserving and re-using the differential data generated by transcoding process. First, we show problems in the scalability by encoding and decoding process, and propose the scalable coding method by separating and composing of MPEG-2 bit streams. Especially, we aim at the improvement of coding efficiency, we consider the separating and composing algorithm. Next, we point that it is possible to estimate the changing of DCT coefficient block that generated by re-quantization process, and show the coding algorithm of this changing data by classifying coefficient value and re-constructing method of original DCT block. Last, by simulation experiments, we show that the proposed method prevents the increase of total bit-count by separating process, moreover it brings re-compressing effect. And by cascading the bit stream separating processor, it can easily realize the multi layer scalability, we show the effectiveness of proposed scalable coding method.

keyword Scalable Video Coding, Transcoder, Bitstream Separating and Composing, MPEG-2

1. はじめに

MPEG 等の標準符号化方式を利用した映像通信サービスにおいて、一つの映像ソースに対して利用環境に応じて様々な品質を提供可能なスケーラブル映像通信サービスへの期待は高い。このような中、高品質符号化情報からの符号変換処理により要求品質に対する柔軟なスケーラビリティを実現する手段として、ビデオトランスコーダが検討されている [1]。トランスコーダには、符号化情報の生成装置、再生装置に特別な処理を追加せずにスケーラビリティを実現可能な点に大きな特徴がある。ここで、トランスコード処理過程で発生する入力映像情報の変化成分を出力映像情報とは別に保存し再利用することで、出力映像を基本階層、変化成分情報を高位階層とした階層符号化機能が符号化器/復号器の機能とは別にして実現される。

そこで、本稿では、符号化器/復号器の外部機能による階層符号化機能実現を目的として、トランスコーダ入出力間の映像変化成分を保持し、それを再利用することでスケーラブルな映像蓄積・配信を実現する新たな符号化方式を提案する。

2. ビットストリーム分離・合成によるスケーラビリティ

一つの映像ソースから異なる品質の映像情報を同時に提供する機能として MPEG-2 では階層符号化処理によるスケーラビリティの実現方式が導入されている [2]。符号化情報を、基本となる画質を与える基本階層と基本階層の画質を向上させるための付加情報となる高位階層という形に階層構造を持たせることで、受信端末において受信する階層の選択により様々な映像品質を得ることが可能となる。しかしながら、符号化情報の生成の段階で階層化処理を行うので、符号化器/復号器の構造の複雑化、処理量の増大、符号化効率の低下 [3]、要求条件が符号化時に想定したものと異なったときに対処が困難なことによる柔軟性の問題等を有している。

本稿ではこのような機能をトランスコーダの拡張により実現する方法を提案する (図 1)。入力される非階層の MPEG-2 ビットストリーム (Bitstream A) はトランスコーダによってビットレートが削減され、ビットレートが下げられた MPEG-2 ビットストリーム (Bitstream B) を基本階層として出力する。また、Bitstream A と Bitstream B 間の映像信号の変化成分を符号化した変換差分ビットストリーム (Bitstream C) を高位階層として出力する。このように、非階層符号化情報 (Bitstream A) を入力して階層符号化情報 (Bitstream B, C) を生成する処理器をビットストリーム分離器と定義する。また、上記とは逆に階層符号から非階層符号への復元を行う処理器をビットストリーム合成器と定義する。

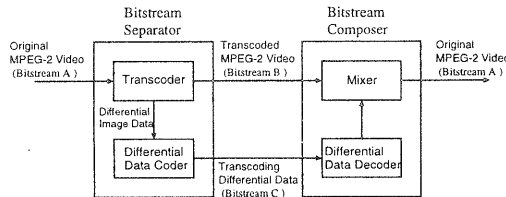


図 1 ストリーム分離・合成によるスケーラビリティ

ビットストリーム分離器はトランスコーダの機能を内包しているため、基本階層にあたる Bitstream B は MPEG-2 のフォーマットを維持しており、汎用的な復号器で復号再生が可能である。また目標ビットレートを調節可能なため高い柔軟性を有している。さらに、受信側において高位階層を受信

する処理器を復号器とは別に設けることで階層構造を有する特別な復号器を用いずに高位階層の復号画像を再生できる。

このように、符号化器/復号器とは別の処理器により階層化を実現することで、汎用性・柔軟性を持ったスケーラビリティが実現される。以下では、特に符号化効率の向上に焦点を当てて差分情報 (Bitstream C) の効率的な符号化アルゴリズムを述べる。

3. 差分映像信号符号化アルゴリズム

3.1 再量子化演算による係数値の変化

トランスコーダにおけるビットレート削減処理と差分情報の抽出方法を図 3 に示す。ビットレート削減処理は、入力ビットストリーム (Bitstream A) を可変長復号 (VLD)、逆量子化 (IQ) して得られる DCT 係数に対し、符号化時よりも粗いステップサイズで再量子化 (Q) を行うことで実現される。このときに係数ブロックが変化するので、この変化成分を Bitstream C として符号化する。具体的には、再量子化前の量子化係数ブロック B_1 と再量子化後の量子化係数ブロック B_2 との差分ブロック $\Delta B (= B_1 - B_2)$ をとり、 ΔB の符号化情報を Bitstream C として出力する。

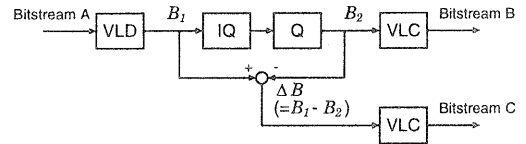


図 2 係数差分ブロックの符号化方法

逆量子化、再量子化処理は、逆量子化演算と再量子化演算を結合させた演算を用いて、再量子化入力量子化係数値 $k_1 (= B_1(u, v))$ から直接再量子化出力係数値 $k_2 (= B_2(u, v))$ を算出する。イントラの場合の演算を式 (1) に、インターの場合の演算を式 (2) にそれぞれ示す [1]。 (u, v) は係数ブロックにおける座標を表し $0 \leq u \leq 7, 0 \leq v \leq 7$ である。

$$k_2 = \left\lfloor k_1 \times \frac{MQ_1}{MQ_2} + \frac{1}{2} \times \text{sign}(k_1) \right\rfloor \quad (1)$$

$$k_2 = \left\lfloor \left\{ k_1 + \frac{1}{2} \times \text{sign}(k_1) \right\} \times \frac{MQ_1}{MQ_2} \right\rfloor \quad (2)$$

ここで、再量子化処理特有のレート歪み特性 [4] を考慮して、符号化効率が劣化する領域を避けるように再量子化パラメータを選択する。再量子化パラメータ MQ_2 を、符号化時 (再量子化前) の量子化パラメータ MQ_1 と整数値 m を用いて、イントラの場合は式 (3)、インターの場合は式 (4) のように算出して、 MQ_2 の値域を特定の値に制限する。

$$MQ_2 = \begin{cases} MQ_1 & (m = 0) \\ 2m \times MQ_1 + 1 & (m \neq 0) \end{cases} \quad (3)$$

$$MQ_2 = (m + 1) \times MQ_1 \quad (4)$$

ここで、式 (3)、式 (4) のように MQ_1, MQ_2 の関係が与えられるときの式 (1)、式 (2) の演算において k_1, k_2 間の関係として以下の式 (5)、式 (6) が成立する [5]。

$$\text{I: } k_2 = 0 \quad (0 \leq k_1 \leq m) \quad (5)$$

$$\text{II: } k_2 < k_1 \quad (k_1 > m) \quad (6)$$

性質 II は再量子化後の係数値は必ず再量子化前よりも値が減少することを保証する。すなわち、 B_2 内の有意係数は再量子化演算により全て値が変化したものとなるから、 ΔB

においてこれらの係数と同一位置に必ず差分値情報が存在する。また、 B_2 内の有意係数を参照すれば差分値情報が対応する係数が分かるため、ブロック内位置の特定のための情報が不要となる。

一方、性質 I より m 以下の係数値は再量子化後には零係数としてまとめられるため、 B_2 内の零係数には元からの零係数(非有意係数)と再量子化によって零係数に変化したものが混在する。すなわち、 B_2 の零係数に対応する ΔB の係数値の有意/非有意は分からないため、 B_2 の零係数に対応した ΔB 中の有意係数には係数値とともに位置を特定するための情報が必要となる。

そこで、上記のような性質に基づき、再量子化演算処理で発生する差分量子化係数ブロックの符号化アルゴリズムの方針を以下に示す。

- 方針 1 ΔB 内の有意係数を、 B_2 に対応する係数値が 0 か否かで分類
- 方針 2 零係数に対応する係数は、 ΔB 内の連続する零係数の個数と係数値で二次元ランレングス符号化
- 方針 3 非零係数に対応する係数は、係数値を再量子化前の係数値を予測したときの予測誤差として符号化

3.2 差分量子化係数の符号化方法

ΔB の符号化処理は、 ΔB を B_1 と B_2 を参照しながらジグザグスキャンし、方針 2 に該当する B_2 内の零係数に対応する有意係数を抽出し、符号化した後に、残りの方針 3 に該当する有意係数を符号化する。

まず、 B_2 内の零係数に対応する ΔB の係数の符号化方法について述べる。その手順を以下に示す。

1. B_2 , ΔB をジグザグスキャンの順番で読みだす。 $B_2(u, v)$, $\Delta B(u, v)$ によって以下の 2-1, 2-2, 2-3 の 3 通りの処理に分かれる。
 - 2-1. $\Delta B(u, v) = 0$ の場合
 ΔB 内零係数カウンタ c を 1 だけ増加し、3. の処理へ進む。
 - 2-2. $\Delta B(u, v) \neq 0$ かつ $B_2(u, v) \neq 0$ の場合
 3. へ進む。
 - 2-3. $\Delta B(u, v) \neq 0$ かつ $B_2(u, v) = 0$ の場合
 カウンタ c の値を run として、 $\Delta B(u, v)$ を $level$ とした二次元ランレングスイベント ($run, level$) を生成し、可変長符号化。カウンタ c を 0 にリセット。 run は ΔB 内の連続する 0 の個数を表す。
3. ブロック末尾ならばランレングスの終りを示す符号 $End_of_Runlength$ を符号化し終了。そうでなければ 1. へ戻る。

次に、方針 3 に該当する係数の符号化方法を述べる。以下にその手順を示す。

1. B_2 , ΔB をジグザグスキャンの順番で読みだす。
2. $\Delta B(u, v) \neq 0$ かつ $B_2(u, v) \neq 0$ ならば、 $B_2(u, v)$ から $B_1(u, v)$ を予測したときの予測誤差 $B_{err}(u, v)$ を式 (7) より算出し、 $B_{err}(u, v)$ を可変長符号化する。

$$B_{err}(u, v) = B_1(u, v) - \left[B_2(u, v) \times \frac{MQ_2}{MQ_1} + \frac{1}{2} \right] \quad (7)$$

3. ブロック末尾ならば終了。そうでなければ 1. へ戻る。

以上の手順により、 $B_2(u, v) \neq 0$ に対応する ΔB の係数はジグザグスキャンの順番で符号化出力される。

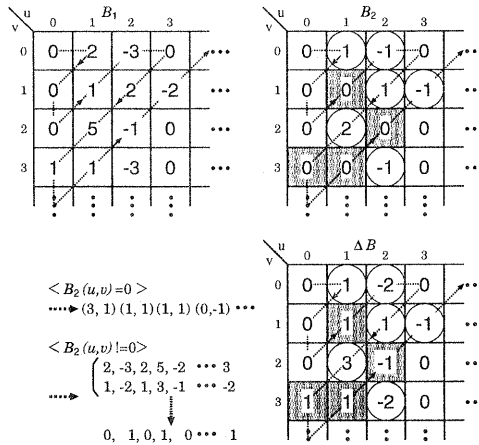


図 3 係数差分ブロックの符号化方法

3.3 復号方法

復号処理では再量子化後の係数情報と差分係数情報を組み合わせて、再量子化前の係数ブロック B_1^* を再構成する。本処理は B_2 の復号処理直後に行われる。

まず、Bitstream C よりランレングス符号を復号し、 B_1^* において run の値からカウントアップし特定されるジグザグスキャン位置に $level$ の値を挿入して、零係数へ変換された係数の再量子化前の値が復元される。ここで、 run の値に $B_2(u, v) \neq 0$ の係数はカウントされていないので、復号された B_2 を参照して、これらの係数位置をスキップする。

$End_of_Runlength$ 符号を検出して全てのランレングス情報を復号し終えたら、 $B_2(u, v) \neq 0$ の係数に対応する差分情報を復号する。本係数情報は、ジグザグスキャンの順番で符号化されているので、 B_2 中の非零係数をジグザグスキャン順に参照し、復号値 $\bar{B}_{err}(u, v)$ と再量子化後係数値 $B_2(u, v)$ から式 (7) を用いて、再量子化前の係数値 $B_1^*(u, v)$ を逆算して求めて、 B_1^* 内の非零係数と同一位置に挿入する。

ここで、 $\bar{B}_{err}(u, v)$ は B_2 内の非零係数にジグザグスキャン順に対応しているため、ブロック内位置を特定する情報がなくても再量子化前の値を復元できる。

4. 評価実験

本稿で提案するビットストリーム分離・合成方式に関して、変換差分ビットストリームの提案符号化アルゴリズムについて評価する。

4.1 実験条件

評価対象方式として、 ΔB を MPEG-2 の符号化方式に従った二次元ランレングス符号化した方式 (方式 1)、提案アルゴリズムに従った ΔB をスキャンし MPEG-2 と同一の符号化表を用いて可変長符号化した方式 (方式 2)、提案アルゴリズムを用いて、さらに確率分布の測定をもとに可変長符号割当を行った方式 (方式 3) の 3 つの方式を比較・考察する。入力ビットストリームは MPEG-2 MP@ML (N=15, M=3) の符号化レート 15 [Mbps] で符号化されており、テストシーケンスとして Bus と Table Tennis を使用し、符号化フレーム数は 150 枚とした。

Bitstream A, B, C それぞれのビットレートを R_A , R_B , R_C [Mbps] とする。実験では、Bitstream B の目標ビットレートを 2 [Mbps] から $(R_A - 1)$ [Mbps] まで 1 [Mbps] きざみでとり、 R_A と $R_B + R_C$ を比較して評価する。

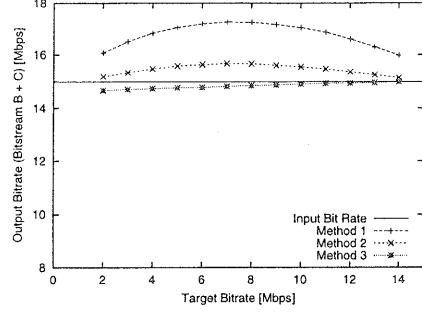
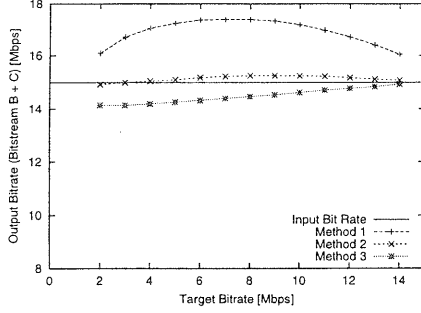


図4 ビットレート合計値 $R_B + R_C$ の変化 (入力符号化レート 15[Mbps])

4.2 実験結果

実験結果を図4に示す。方式1は、ビットレート合計値が入力時を大きく上回っており、分離処理における符号化効率の大幅な低下が見られる。特に、中間の出力レートにおいてピークとなり、増加符号量が入力符号量の約10%に達する。 ΔB には再量子化後の係数値によらずに全ての入力非零係数の変化差分値が含まれているために、 ΔB 中のランレングスイベント数自体は B_1 と変わらない。ここで、対応する非零の再量子化値に対応する係数差分値は復号時に B_2 を参照すれば位置を取得可能なためBitstream Cへは符号化不要である。すなわち、 B_2 の符号化情報の中にこれらの係数の位置情報がランレングス値として含まれているため、方式1では、Bitstream BとCで共有する情報成分により無駄な符号量が多く発生していると考えられる。

方式2は、本稿で提案した ΔB 内の係数の分類符号化方式により、Bitstream BとCで共有して符号化される情報を排除している。これにより入力レートとほぼ等しい出力レートを達成される。符号量の増大を抑制できることより提案アルゴリズムが符号化効率の点で有効であることが分かる。

方式3では、可変長符号化テーブルを差分係数情報の符号化用に最適設計することにより、入力よりも少ない符号量を達成できる。MPEG-2の符号化表では $level$ の値として $\pm 1 \sim \pm 2047$ までが定義されているのに対し、 ΔB のランレングスイベントは式(5)より $level$ の値域が $1 \leq level \leq m$ に制限されるために、MPEG-2のそれに比べて十分に小さな符号化表となる。そのため、符号化表が適切に設計されればMPEG-2よりも短い符号語でランレベルイベントを符号化できる。このように、差分情報の特性を考慮した符号割当を行うことで、さらなる符号化効率の改善が可能となる。

5. 提案差分情報符号化方式の符号化性能

4.の実験結果より、提案方式は分離出力後の総符号量の増大を抑制できるだけでなく、符号化表を適切に設計することで、入力符号量を下回る場合があることを示した。すなわち、提案方式は階層符号化機能の実現方式としてだけでなく、圧縮符号化の一手法としての性能が期待できる。

そこで、変換差分情報の符号化方式の性能を理論的に考察し、圧縮符号化の一手法としての評価を行う。DCT係数をラプラス分布でモデル化した上で4.1に示す各方式での変換差分情報の符号量を導出し、分離出力の合計符号量と入力符号量を比較して評価を行う。

5.1 差分量子化係数情報量の導出

原画像信号のDCT係数 x が符号化器においてステップサイズ Q_1 で量子化されているものとする。 x の分布を、平均が0、分散が σ^2 のラプラス分布と仮定すると、 x 量子化値(量子化係数値) k_1 の発生確率 $P_1(k_1)$ は次の式(8)により表される。

$$P_1(k_1) = \int_{(k_1 - \frac{1}{2})Q_1}^{(k_1 + \frac{1}{2})Q_1} p(x) dx \quad (8)$$

ただし、確率密度関数 $p(x)$ は式(9)により与えられる。

$$p(x) = \frac{1}{\sqrt{2\sigma}} e^{-\frac{\sqrt{2}}{\sigma}|x|} \quad (9)$$

ここで、密度関数 $p(x)$ は $x=0$ に対して対称であり、また量子化係数の絶対値に1ビットの符号ビットを追加することで正負が表現可能なため、本考察では $x \geq 0$ の領域についてのみ議論する。そこで、符号化された信号の符号量 R_1 は、式(10)のように表される。

$$R_1(Q_1) = N \sum_{k_1=1}^{\infty} P_1(k_1) L(k_1) \quad (10)$$

ここで、 N はDCT係数の個数であり、 $L(k)$ は量子化係数値 k を可変長符号化したときの符号量である。

MPEG-2では、量子化係数はジグザグスキャンされて一次元系列に並べられたあと、連続している量子化値が0である係数(ゼロ係数)の長さ(ラン長) run とそれに続く量子化値 k を組み合わせた事象(run, k)に対する二次元ハフマン符号化により量子化係数値を符号化する。すなわち、符号長 $L(k)$ は一意には定まらないので、本考察では各 run に対して平均値をとり $L(k)$ を表現する。そこで、ランレベルイベント(run, k)のハフマン符号長を $len(run, k)$ と表したときに符号長 $L(k)$ は次の式(11)により表される[6]。

$$L(k) = \frac{\sum_{run=0}^{63} \{P(0)\}^{run} \times len(run, k)}{\sum_{run=0}^{63} \{P(0)\}^{run}} \quad (11)$$

次に、上記量子化係数 k_1 を再量子化ステップサイズ Q_2 で再量子化したときの符号量について考察する。3.1で示したように Q_2 は整数値 m と Q_1 により式(3)、式(4)を用いて算出され、再量子化値 k_2 は逆量子化式と再量子化式を結合した式(1)、式(2)により算出される。このとき、式(5)、式(6)から整数値 m により k_1 の再量子化値 k_2 の有意/非有意の境界が与えられるため、再量子化によって減少する符号量 R_{red} を式(12)のように表現する。

$$R_{red}(Q_1, m) = N \sum_{k_1=1}^m P_1(k_1) \times \{L(k_1) - L(0)\} + N \sum_{k_1=m+1}^{\infty} P_1(k_1) \times \{L(k_1) - L(k_2)\} \quad (12)$$

式 (10), 式 (12) より再量子化出力係数の符号量 R_2 は式 (13) により表される.

$$\begin{aligned} R_2(Q_1, m) &= R_1(Q_1) - R_{red}(Q_1, m) \\ &= N \sum_{k_1=m+1}^{\infty} P_1(k_1) \times L(k_2) \end{aligned} \quad (13)$$

次に, 各方式における再量子化前後間の係数差分情報の符号量 R_{diff} を導出する.

(1) 方式 1

方式 1 では係数差分値に対し MPEG-2 の符号化方式と同様のジグザグスキャンと二次元ランレングスにより符号化処理を行う. したがって, 方式 1 での係数差分情報の符号量 R_{diff} は式 (14) により表される.

$$R_{diff}(Q_1, m) = N \sum_{k_1=0}^{\infty} P_1(k_1) \times L(k_1 - k_2) \quad (14)$$

(2) 方式 2, 方式 3

方式 2, 方式 3 では, 係数差分値は入力係数値を参照して m を境界にした分類し符号化される. まず, $k_1 \leq m$ である係数は再量子化前の係数ブロック中の連続するゼロ係数をラン長とし, これに k_1 の値を組み合わせた二次元ランレイベントを用いてランレングス符号化される. 本ランレングス情報の符号量 R_{rl} は式 (15) のように表される.

$$R_{rl}(Q_1, m) = N \sum_{k=1}^m P_1(k_1) \times L_{rl}(k_1) \quad (15)$$

式 (15) において, $L_{rl}(k_1)$ はランレングス符号化される係数の係数値 k_1 の符号量でありラン長分布が再量子化前に対してほとんど変わらないと仮定すると, 式 (11) と同様にして, 差分ランレングスイベントの符号語長 $len_{rl}(run, k)$ を用いて式 (16) に表現される.

$$L_{rl}(k_1) = \frac{\sum_{run=0}^{63} \{P(0)\}^{run} \times len_{rl}(run, k)}{\sum_{run=0}^{63} \{P(0)\}^{run}} \quad (16)$$

一方, 残りの $k_1 > m$ の係数は, k_2 から k_1 を予測したときの予測誤差 Δk を算出し, Δk を可変長符号化して符号化される. Δk に対する符号語長を $len_{res}\{\Delta k(k_1, k_2)\}$ とすると, 符号量 R_{res} は式 (17) により算出される.

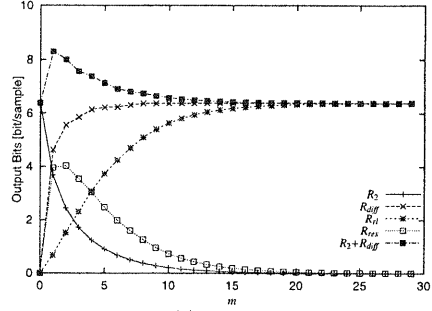
$$R_{res}(Q_1, m) = N \sum_{k_1=m+1}^{\infty} P_1(k_1) \times len_{res}\{\Delta k(k_1, k_2)\} \quad (17)$$

したがって, 係数差分値の符号量 R_{diff} を式 (15) と式 (17) の和として式 (18) に導出する.

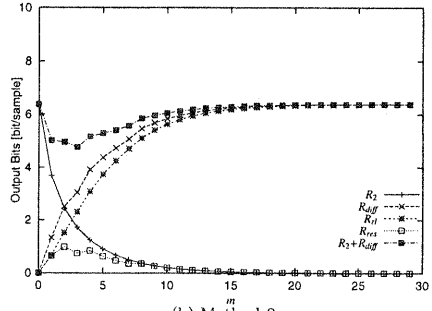
$$R_{diff}(Q_1, m) = R_{rl}(Q_1, m) + R_{res}(Q_1, m) \quad (18)$$

方式 2 では, ランレングスイベントに対して MPEG-2 と同一の符号語を割り当てるので, 符号語長 $len_{rl}^{(2)} = len(run, k)$ となる. また $len_{res}^{(2)}(\Delta k)$ については, Δk の値域に応じた固定長符号とする.

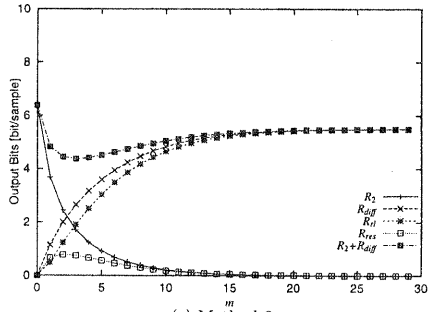
方式 3 では, $len_{rl}^{(3)}(run, k)$ と $len_{res}^{(3)}(\Delta k)$ とともに発生確率の分布に基づくハフマン符号により導出する.



(a) Method 1



(b) Method 2



(c) Method 3

図 5 m に対する各種符号量の変化 ($Q_1 = 2$)

5.2 符号化性能評価

5.1 より導出される各符号量を算出し, 提案方式の符号化特性を検証する. $\sigma = 10.0$, $Q_1 = 2$ としたときの m に対する $R_2(Q_1, m)$, $R_{diff}(Q_1, m)$, $R_{rl}(Q_1, m)$, $R_{res}(Q_1, m)$ の変化を図 5 に示す. $m = 0$ においては再量子化を行わないので, このときの符号量は入力符号量 R_1 に相当する. ただし, 方式 1 については R_{diff} のうち, $k_1 \leq m$ の入力係数に対する変化差分情報量を R_{rl} として, $k_1 > m$ に対する変化差分情報量を R_{res} としている.

図 5 より, 方式 1 は $m = 1$ から 10 の範囲で分離出力総符号量 $R_2 + R_{diff}$ が入力符号量を上回ることが分かる. R_{rl} は, R_2 の指数関数的な減少とほぼ対称的に増加する傾向を示しており m が低い領域では R_{rl} は少ない. しかし, R_{res} が占める符号量が特に多いために, 結果的に分離後の総符号量は入力時を上回り, 符号化効率が低下する. したがって, 符号化効率の改善には R_{res} に要する符号量を削減することが重要であることが分かる.

方式 2, 方式 3 では $k_1 > m$ の係数の変化情報に対して, ラン情報を符号化せず係数値のみを符号化対象とすることで,

必要な符号量を削減している。これにより R_{res} の符号量が大きく改善されることが図 5 に示されている。また、式 (7) に示す演算を行うことで、値の大きな入力係数に対しても必要な符号量を少なくできる。このように $k_1 > m(k_2 \neq 0)$ の係数に対する変化値は、再量子化値を利用して不要な符号を排除し、再量子化値だけでは再現できない情報成分のみを符号化対象とすることで、通常の二次元ランレングス符号化方式に比べて 1/4 程度の符号量で表現可能となる。

さらに、発生確率分布をもとに符号化テーブルを再設計することによって、特にランレングスの符号量 R_{rl} が大幅に削減されることが方式 3 の結果よりわかる。 m が大きい領域でも $R_2 + R_{diff}$ が R_1 を下回っており、二次元ランレングスの値域の縮小による符号語長の削減が再圧縮に効果的であることが分かる。

次に、 m とともに Q_1 を変化させ、入力符号量 R_1 と m に対する分離出力総符号量 $R_2 + R_{diff}$ の特性を図 6 に示す。図 6 より、 R_1 が大きいほど各方式の違いが顕著に現れ、提案方式は、入力係数が多い高レートで符号化されたビットストリームに対して特に高い効果を発揮することがわかる。

提案方式は、係数の変化差分値に対して、量子化等の非可逆処理は行わないので、復号処理後には再量子化前と全く同一の係数情報が再現される。また、可変長符号の割り当てに、式 (3)、式 (4) で用いた整数値 m を利用することで、割り当て対象の値域を制限し、導出される可変長符号長を制限する。その結果、全く同一の係数情報が、MPEG-2 で用いられている二次元ランレングス符号化方式よりも少ない符号量で表現可能となる。

このように、提案差分係数符号化アルゴリズムは、トランスコーダを用いた階層符号化情報の提供手段としてだけでなく、DCT と量子化をベースにした符号化方式における可変長符号化方式の新たな手法としても有効といえる。

6. まとめ

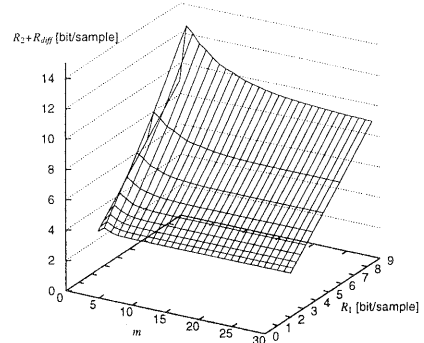
本稿では、MPEG-2 ビデオビットストリームを対象としたトランスコーダの入力/出力間の差分画像信号を符号化することで、トランスコーダを用いた階層符号化機能を提供方法を述べ、差分画像信号の符号化方法を提案した。

まず、差分画像信号の符号化処理器をトランスコーダの機能拡張として実装することで非階層から階層構造への変換を行うビットストリーム分離器と、元の非階層符号化情報を生成するビットストリーム分離器を導入し、ビットストリーム分離・合成による階層符号化機能の実現方法を述べた。次に、両者のアルゴリズムを述べ、再量子化係数値の変化の性質を利用した差分画像の符号化方法を提案した。さらに、評価実験を行い提案方式により符号化効率の低下を抑制可能であり、符号化性能を理論的考察から圧縮符号化アルゴリズムの一手法としての有効性を示した。

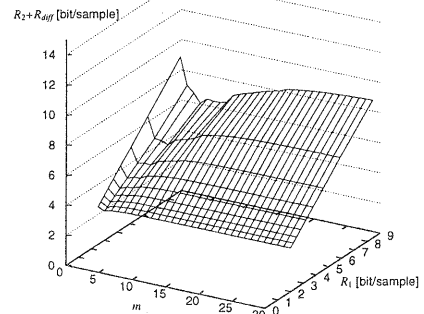
ビットストリーム分離合成機能は、トランスコーディングにより階層符号化機能を実現することで、これらの二つの機能を融合する技術としてデジタル映像通信の分野で有用なものとなると考えられる。今後の課題としては、ビデオ以外の音声や同期情報などを含めた多重化ストリームを対象とする提案映像通信システムの実現が挙げられる。

文 献

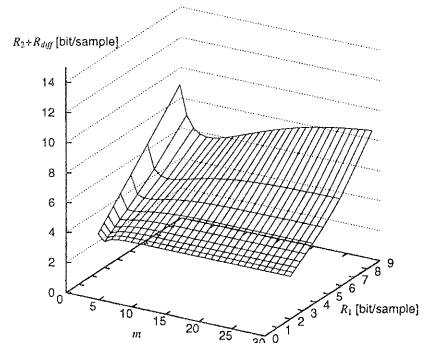
- [1] 西村 敏, 笠井 裕之, 高屋 和幸, 亀山 涉, 榊 直昭, 花村 剛, 富永 英義: “リアルタイム MPEG-2 ビデオトランスコーダソフトウェアの開発”, 情処研報, AVM25-5, pp. 25-30 (1999).
- [2] 花村 剛, 富永 英義: “スケーラブルビデオ符号化”, テレビ技報, ITE 17-57 (1993).
- [3] 笠井 裕之, 児玉 明, 富永 英義: “階層符号量配分率の動的更新制御による動画像階層符号化符号量制御



(a) Method 1



(b) Method 2



(c) Method 3

図 6 (R_1, m) に対する出力総符号量の変化

- 方式”, 信学論 B, 第 J82-B 巻, no.8, pp. 1475-1468 (1999).
- [4] 永吉 功, 笠井 裕之, 花村 剛, 亀山 涉, 富永 英義: “再量子化特性を利用した MPEG-2 トランスコーダ量子化制御方式の検討”, 信学技報, IE-99-75, pp. 9-16 (1999).
 - [5] 永吉 功, 花村 剛, 笠井 裕之, 富永 英義: “MPEG-2 ビットストリーム分離・合成によるスケーラブル映像符号化方式の検討”, 信学技報, IE-2000-95, pp. 69-76 (2000).
 - [6] 角野 真也, 柴藤 稔, 横矢 直和: “MPEG-2 再符号化における再量子化誤差の検討”, 信学技報, IE-99-32 (1999).