

## JPEG2000 Precinct 構造を用いた低遅延 HDTV 符号化方式

佐野 雄磨<sup>†</sup> 内藤 整<sup>††</sup> 渡辺 裕<sup>†</sup>

<sup>†</sup> 早稲田大学 大学院 国際情報通信研究科

〒 169-0051 東京都新宿区西早稲田 1-3-10

<sup>††</sup> 株式会社 KDDI 研究所

〒 356-8502 埼玉県上福岡市大原 2 丁目 1 番 15 号

E-mail: †yuma@tom.comm.waseda.ac.jp

あらまし 近年、デジタル放送の開始に伴い映像素材は HDTV 化の流れにある。HDTV 素材の高品質伝送を目的として、高い圧縮率とスケーラビリティを持つ JPEG2000 による HDTV 符号化の研究が行われてきた。放送局間の素材伝送や番組配信においては伝送遅延の低減がユーザ要求として強く求められるが、ビデオフレームといった、符号化処理単位を大きく設定するにつれて符号化遅延は大きくなる。これに対し、入力画像を分割しそれぞれ独立して符号化し伝送するという手法が考えられる。これまでもタイルを用いた分割符号化が検討されてきたが、低ビットレート時に発生するタイル歪みや、符号化効率の低下という問題点があった。そこで本稿では、JPEG2000 の Precinct 構造を利用して、分割による境界歪を発生させずに低遅延で符号化する手法について検討する。

キーワード JPEG2000, HDTV, 低遅延符号化, タイル, precinct, ビット配分, バッファリング

## Low delay coding using JPEG2000 precinct structure

Yuma SANO<sup>†</sup>, Sei NAITO<sup>††</sup>, and Hiroshi WATANABE<sup>†</sup>

<sup>†</sup> Graduate School of Global Information and Telecommunication Studies, Waseda University  
Nishiwaseda 1-3-10, Shinjuku-ku, Tokyo, 169-0051 Japan

<sup>††</sup> KDDI Labs

Ohara 2-1-15, Kamihukkuokashi, Saitama, 356-8502 Japan

E-mail: †yuma@tom.comm.waseda.ac.jp

**Abstract** In recent years, HDTV is being used for digital video broadcast. For high quality transmission of HDTV, much research about encoding HDTV using JPEG2000 that provides high compression efficiency and scalability, has been conducted. Strict low delay is required for real time video transmission or TV program distribution. The larger the coding unit such as video frame, the more the coding delay. Encoding using a smaller coding unit such as with tile structure has been studied for low coding delay, however tile artifact at low bitrate is still a big problem. In this paper, we show that low delay coding using JPEG2000 precinct structure results in substantially reduced discontinuity at boundary of division.

**Key words** JPEG2000, HDTV, low delay coding, tile, precinct, bit allocation, buffering

## 1. はじめに

2003年12月から、地上波デジタル放送が一部の地域で始まった。デジタル放送によって、従来のアナログ放送の1チャンネル分と同じ周波数帯域幅でHDTVを配信することが可能になる。これに伴い、映像素材はHDTV化の流れにある。HDTV素材の高品質伝送を目的として、高い圧縮率とスケラビリティを持つJPEG2000 [1] によるHDTV符号化の研究が行われてきた [2]。

放送局間の素材伝送や番組配信においては伝送遅延の低減がユーザ要求として強く求められる。しかし、ビデオフレームといった、符号化処理単位を大きく設定するにつれて符号化遅延は大きくなる。例えば、MPEGなどの動画像符号化では、フレーム単位の符号化処理を適用しており、この場合、符号化に先立ち符号化対象となる画像データの読み込みが完了するのに最低でも1フレーム分(約33msec)待たされることになる。これに対し、入力画像を分割しそれぞれ独立して符号化することによって伝送遅延を低減するという手法が考えられる [3] [4]。これまでもJPEG2000のタイル構造を用いた分割符号化が検討されてきた。しかし、入力画像をタイルに分割し、各タイルを独立に符号化するために、低ビットレート時にタイルの境界付近で発生する歪みや、符号化効率の低下という問題点があった。そこで本稿では、JPEG2000のPrecinct構造を利用して、分割による境界歪を発生させずに低遅延で符号化する手法について検討する。

## 2. 従来の低遅延符号化手法

### 2.1 低遅延符号化の概念

まず、本研究ではJPEG2000の符号化をハードウェア上で行うことを想定している。ハードウェア上でJPEG2000の符号化を行う場合、ウェーブレット変換以降の処理が一瞬で行えると考えられるため、一連の符号化過程の中で最も時間がかかるのは、入力画像の読み込みであるといえる。符号化処理単位を1フレームとすれば、フレームのサイズが大きくなるに従って、読み込み時間は大きくなり、結果として符号化遅延が増大する。これに対し、入力画像を空間的に分割し、読み込んだ領域から順次符号化を開始することによって符号化の遅延時間を減少させる方法が考えられる。JPEG2000では、この分割領域としてタイルを用いて符号化することが可能である。

### 2.2 タイル化

JPEG2000では、入力画像を矩形のタイルに分割可能である [1]。分割された各タイルは、タイル境界を越えての画素参照は行わず独立に符号化、復号される。サイズの大きい画像を符号化する際には、タイルを用いて並列的に符号化、復号することによって処理を軽減し、遅延を減少することが可能である。しかし一方で、低ビットレート時には、タイルの境界付近は画素値の不連続性が生じるという欠点がある [5] [6]。

### 2.3 タイル化歪

入力画像から分割されたタイルは、それぞれ独立

に符号化される。wavelet変換の際に、タイルの境界の係数については、係数を折り返すことによってタイル外部への係数値を得てwavelet変換を行う。一方で、タイル分割しない場合は、タイル境界にあたる位置の係数は折り返し処理を行わず、フィルタ長の係数を参照してwavelet変換を行う。タイル分割を行うと、接する2つのタイルで折り返し処理によってwavelet変換が行われるために、その境界に不連続性が生じ、画質の劣化の原因となる。特に低ビットレート時にその傾向は顕著である。図1にタイル化歪の様子を示す。画像を中央に水平な歪が確認できる。



図1 タイルの分割法(左)とタイル化歪(右)

JPEG2000のタイル分割を用いて、HDTV素材の低遅延符号化を行う場合、このタイル化歪を除去または低減することが必要となる。本稿では、分割の単位をタイルではなくPrecinctを用いることにより、境界の不連続性による歪を発生させない低遅延符号化手法を検討する。次章で、その手法について説明する。

## 3. Precinctを用いた低遅延符号化

### 3.1 Precinct

JPEG2000の符号化手順の中で、コードブロックは集められてPrecinctという領域を形成することが可能である。Precinctは単なるコードブロックの集合であり、Precinctサイズはコードブロックサイズを考慮して設定する必要がある [1]。タイルとは異なり、Precinctを設定することによって符号化や画質には影響を与えることはない [7]。また、Precinctの大きさは、2のべき乗である必要があるが、そのサイズは各解像度レベルで異なる値をとることが可能である。また、JPEG2000では、タイルパートごとにパケットの順序を決めることが可能である。Precinctは、パケット順序を決める単位の一つであり、Precinctごとにパケットを連続的に配置可能である。特定のPrecinct情報を得たり、空間的なランダムアクセスなどを行う際に有用である [8] [9]。以下に、タイルによる分割とPrecinctによるの違いを示す。

入力画像の上下二等分をタイルとして設定した場合のサブバンド分割の様子を図2に、Precinctとして設定した場合のサブバンド分割の様子を図3に示す。サブバンド分割を3段階した後の最低域成分におけるタイルもしくはPrecinctが図の右側の黒色部分にあたる。タイルはそれぞれのタイル内でサブバンド分割が行われ、Precinctは画像全体でサブバンド分割された後、Precinctとして分割される。

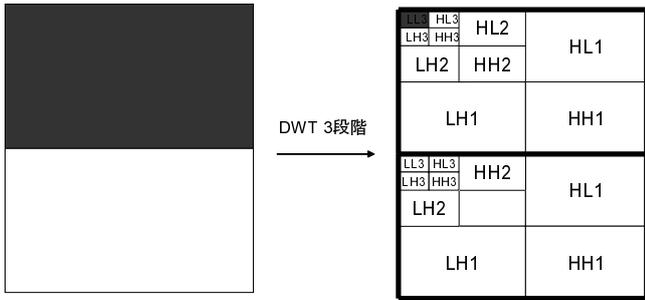


図 2 タイル構造のサブバンド分割

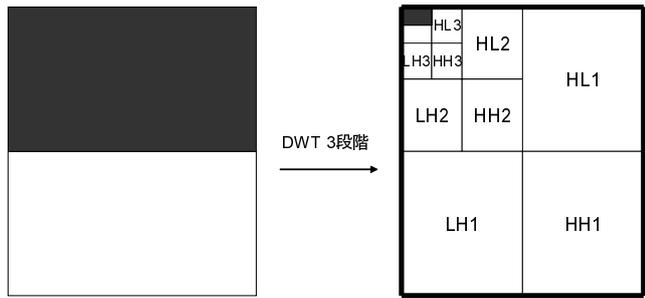


図 3 Precinct 構造のサブバンド分割

### 3.2 Wavelet 変換の有効タップ長

DWT(DiscreteWaveletTransform) での低域成分は、DWT ローパスフィルタを多段にかけることによって得られる [10]。ここで、最低域成分の 1 画素が符号化されるのに参照した画素数のことを「有効タップ長」とする。有効タップ長は DWT のレベル数とフィルタのタップ数から求めることが可能である。9 × 7 フィルタを用いた時の、有効タップ長の様子を図示したものを図 4 に示す。

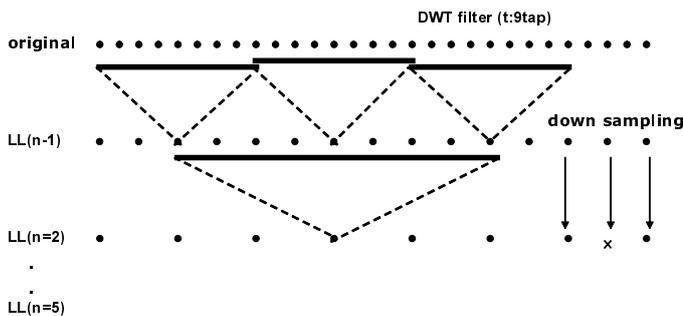


図 4 DWT フィルタ長と段階数

DWT のレベル数を  $n$ ，フィルタのタップ数を  $t$  とすると，有効タップ長は以下のように表される。

$$\text{有効タップ長} = (2^n - 1)(t - 1) + 1$$

例えば，5 × 3 整数フィルタを用いて，5 段階 DWT を行うと有効タップ長は

$$(2^5 - 1)(5 - 1) + 1 = 125$$

となる。すなわち，注目画素の周辺 125 画素がその画素の符号化に影響を与えることになる。以上より，画像を分割して符号化する際に，この有効タップ

長よりも多くオーバーラップさせて分割すれば，注目領域に関しては分割の影響なく DWT を行うことが可能である。つまり，分割しないで符号化した場合と同じ結果が得られる。

### 3.3 提案手法の概要

提案手法の概要を図 5 に示す。

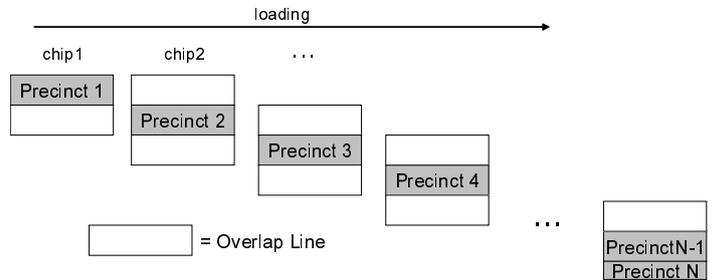


図 5 提案手法の概要図

提案手法を送信側，受信側のについてそれぞれの動作を説明する。

#### a) 送信側

伝送の単位として Precinct を用いる。図 5 のように，入力画像の上端から水平に Precinct を設定する。そして，チップを複数使って，順次 Precinct を読み込んでいく。まず，1 番目のチップは，第 1 Precinct とオーバーラップ分のラインを読み込み符号化を行う。符号化終了後，第 1 Precinct のデータだけを受信側へ伝送する。その間，2 番目のチップは第 2 Precinct とその上下のオーバーラップ分のラインを読み込み符号化を行う。符号化終了後第 1 Precinct と同様に，注目 Precinct のデータのみを送信する。その後も，残っているチップや符号化の終了したチップを用いて，同様に符号化を行う。

#### b) 受信側

受信側では，Precinct ごとにデータを受信する。JPEG2000 では，Precinct 構造をとる場合，全ての Precinct がそろわなくては，画像全体を復号することは出来ない。しかし，一つ後の Precinct を受信すれば，その前に受信した Precinct に関しては，復号しそのデータを得ることが可能である。したがって，受信側では，注目している Precinct を復号するために一つ次の Precinct の受信を待たなくてはならないが，しかし，Precinct のサイズはフレームと比較して小さく，フレームを符号化単位とした時に比較して，最初のデータを得るまでの遅延時間が小さくなるといえる。オーバーラップをとってれば，注目 Precinct に境界歪が生じることはないので，Precinct を小さく取ることによって，符号化遅延をますます低減することが可能である。

### 3.4 ビット配分

前節で述べたように，提案手法では，Precinct を上から順番に読み込んだものから符号化を行う。そのため，ある Precinct を読み込み符号化する時点では，それ以降の Precinct の画像情報はまったくわからない。通常，入力画像全体で符号化する場合は，画像全体で wavelet 変換を行い，画像全体の画像特性に基づいて，RD 最適化によって各 Precinct にビット配

分される。しかし、提案手法の場合、他 Precinct の画像特性もわからないため、ある Precinct を符号化する時点で最適なビット配分を行うことは困難である。仮に各 Precinct に同じビットレートを割り当てて符号化を行うと、Precinct によって情報量が異なるため、情報量が少ない Precinct に余分なビットが割り当てられたり、情報量が多い Precinct に十分にビットを割り当てられないという事態が発生し、結果的に画像全体で符号化したときと比較して、画質が劣化する。

そこで、本稿では前フィールドのビット分布を用いて、それ以降のフィールドのビット配分を最適化することを考える。提案するビット配分手法を図 6 に示す。

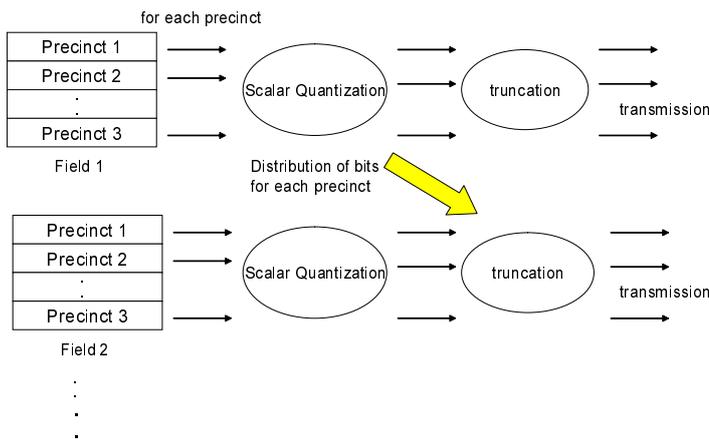


図 6 提案ビット配分手法の概念図

まず、最初のフィールドに関しては、前フィールドの情報も無く、各 Precinct に最適なビット配分を行うのは不可能であるため、各 Precinct について、想定ビットレートで符号化を行う。その際、JPEG2000 の符号化処理の中で、まずスカラー量子化を行い、その後ポスト量子化によって、想定ビットレートに調整する。ここでスカラー量子化後の各 Precinct の符号量を記録しておく。各 Precinct のスカラー量子化後の符号量は、その Precinct の情報量の指標となるため、この値を記録し、注目 Precinct の符号量が全体の符号量のどれくらいの割合になるかを算出する。第 2 フィールド以降は、一つ前のフィールドの注目 Precinct のスカラー量子化後の符号量の全体に対する割合を、1 フィールドに割り当てられる符号量にかけて、その Precinct に配分するビット量を決める。1 フィールドごとの符号量は、フィールドサイズとビットレートによって求められる。フィールドの縦の長さを  $L_v$ 、横の長さを  $L_h$ 、ビットレートを  $R$  とすると、1 フィールドに割り当てられる符号量  $B$  は、

$$B = L_v \times L_h \times R$$

と表される。第  $n$  フィールドの第  $k$  Precinct の符号量の割合を  $X_{n,k}$  とすると第  $n+1$  フィールドの第  $k$  Precinct に割り当てられる符号量  $P_{n+1,k}$  は以下のように計算可能である。

$$P_{n+1,k} = B \times X_{n,k}$$

動画では、前フィールドと次フィールドの画像

情報は類似していると考えられるので、これによって画像全体の特徴を考慮したビット配分を行うことが可能である。

### 3.5 ビット配分の実験

前節で述べた、ビット配分手法について実験を行った。入力画像には、ITE 標準画像の Streetcar (1920 × 1080, 24bit) を用いた。符号化の条件として、9X7 フィルタを用い、3 段階 DWT を行い、ビットレートは 0.8bpp とした。さらに、Precinct サイズを 2048 × 64 とし、1 フィールドに 9 Precinct 存在するように設定した。また、プログレッションは PCRL である。本手法では、フィールドベース符号化を行うため、まず第 1 フィールドを上端から順次読み込み符号化を行う。そして、各 Precinct でスカラー量子化を行い、その後の符号量を記録した上で、ポスト量子化を行い符号量を調整する。表 1 に第 1 フィールドの各 Precinct の量子化後の符号量の割合を示す。

表 1 スカラー量子化後の各 Precinct の符号量の割合

Precinct	符号量の割合	Precinct	符号量の割合
第 1	0.0656	第 6	0.1576
第 2	0.0935	第 7	0.1558
第 3	0.1209	第 8	0.0679
第 4	0.1448	第 9	0.0326
第 5	0.1613		

1 フィールドに割り当てられるビット量は、  
 $1920 \times 540 \times 0.8 = 839440$  bit

である。従って、第 2 フィールドで各 Precinct に割り当てられるビット量は表 2 のようになる。

表 2 第 2 フィールドで各 Precinct に割り当てられる符号量

Precinct	符号量 (bit)	Precinct	符号量 (bit)
第 1	54411	第 6	130719
第 2	77552	第 7	129226
第 3	100279	第 8	56318
第 4	120102	第 9	27039
第 5	133788		

また、画像全体で符号化しビット配分した時の各 Precinct のビット量を表 3 に示す。

表 3 フィールド全体で符号化した場合の各 Precinct の符号量

Precinct	符号量 (bit)	Precinct	符号量 (bit)
第 1	54304	第 6	128656
第 2	79224	第 7	128360
第 3	98960	第 8	62048
第 4	118768	第 9	27296
第 5	132280		

表 2 と表 3 より、提案手法によって、画像全体で符号化しビット配分した時とほぼ同様のビット配分が行われていることがわかる。

次に、画像全体で符号化され理想的なビット配分が行われた画像と、提案手法によって符号化された画像の品質を比較した。

	PSNR [dB]
理想的なビット配分	: 33.286
提案手法	: 33.278

この結果から、提案手法は理想的なビット配分に近いPSNRを取ることがわかる。また、オーバーラップをとって符号化を行ったために、境界の不連続性による視覚的な画質の劣化が生じていない。従って、低ビットレート時も境界の不連続性を発生させることなく画質を維持することが可能である。

### 3.6 バッファリング

本手法では、各 Precinct には、最初のフレームを除いて異なるビット量を割り当てて符号化し、伝送する。そのため、CBR(Constant Bit Rate) 回線で伝送する場合には送信側、受信側で、バッファリングが必要となる[11]。バッファリングは、伝送遅延の一因となりうるが、符号化の単位をフレームとした時と比較して、Precinct のサイズが小さいため、大きな遅延が生じないようにバッファを設計することが可能である。提案手法を用いた場合のバッファモデルを考える。図7に受信側のバッファモデルと送信側のバッファモデルを示す。

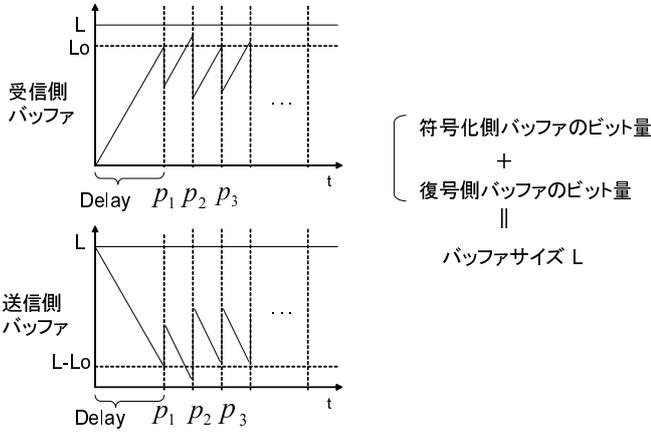


図7 受信側と送信側のバッファモデル

$p_k$  は  $k$  番目の Precinct が符号化、もしくは復号される時点を表す。伝送ビットレートを  $R$ 、フィールドレートを  $F$ 、各フィールドの Precinct 数を  $P$  とすると、

$$p_k - p_{k-1} = \frac{1}{F \times P}$$

と表すことが出来る。また、Delay は受信側のバッファリングによる遅延時間で、 $L_o$  とは、

$$Delay = \frac{L_o}{R}$$

という関係を持つ。また、 $L_o$  はバッファサイズ  $L$  を用いて、 $L_o = iL$  と表すことが出来る。(  $0 < i < 1$  )

このバッファリングモデルにおいて、あるフィールドの  $n$  番目の Precinct の符号化が終わった時、送信側バッファには、

$$x = \sum_{j=1}^n (j) - \frac{(n-1)R}{FP} + (L - L_o)$$

のビット量が存在する。

( $d(j)$  は  $j$  番目の Precinct のビット量、 $L - L_o$  は送信側バッファの初期値)

送信側バッファと受信側バッファのビット量の合計をバッファサイズ  $L$  となるので、受信側のビット量  $B(n)$  は、以下のように表される。

$$\begin{aligned} B(n) &= L - x \\ &= L - \sum_{j=1}^n (j) + \frac{(n-1)R}{FP} - (L - L_o) \\ &= L_o - \sum_{j=1}^n (j) + \frac{(n-1)R}{FP} \end{aligned} \quad (1)$$

次に、 $n$  番目の Precinct について、オーバーフローとアンダーフローを防ぐ条件をそれぞれ以下に示す。

c) アンダーフロー

$$\begin{aligned} B(n) + \frac{R}{FP} - d(n+1) &\geq 0 \\ \Leftrightarrow B(n) + \frac{R}{FP} &\geq d(n+1) \end{aligned} \quad (2)$$

d) オーバーフロー

$$\begin{aligned} L > B(n) + \frac{2R}{FP} - d(n+1) \\ \Leftrightarrow d(n+1) > B(n) + \frac{2R}{FP} - L \end{aligned} \quad (3)$$

(2)(3) 式をまとめると

$$B(n) + \frac{2R}{FP} - L < d(n+1) \leq B(n) + \frac{R}{FP} \quad (4)$$

となる。(1) 式を (4) 式に代入すると

$$\begin{aligned} L_o - \sum_{j=1}^n (j) + \frac{(n+1)R}{FP} - L < d(n+1) \\ \leq L_o - \sum_{j=1}^n (j) + \frac{nR}{FP} \end{aligned} \quad (5)$$

となり、 $n+1$  番目の Precinct のビット量が満たすべき条件が求められた。この中で、前章のビット配分で各 Precinct に割り当てられるビット量は決まっているので、全ての Precinct について (5) 式を満たすようなバッファ量  $L$  を求め、バッファを設計することが可能である。

想定する条件として、伝送ビットレート=50Mbps、フィールドレート=60fps、Precinct 数=9、 $L_o = \frac{5}{7}L$  を考える。この条件の下、全ての Precinct で (5) 式を満たすバッファモデルの 5 フィールド分のバッファ占有量の例を、Streetcar, Europeanmarket, Openingceremony の 3 種類の標準画像について図8, 9, 10に示す。

図8, 9, 10のバッファモデルでは、 $L$  は 350kbit であり、送受信含めた遅延は  $0.7 \times 10^{-2}$ sec となる。従って、バッファを用いることによって符号化遅延は生じるが、符号化単位を Precinct としているため、遅延時間は少なくおさえることが可能である。

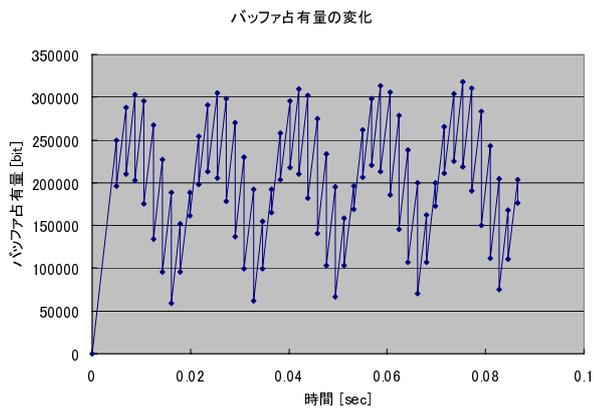


図 8 バッファ占有量 (Streetcar)

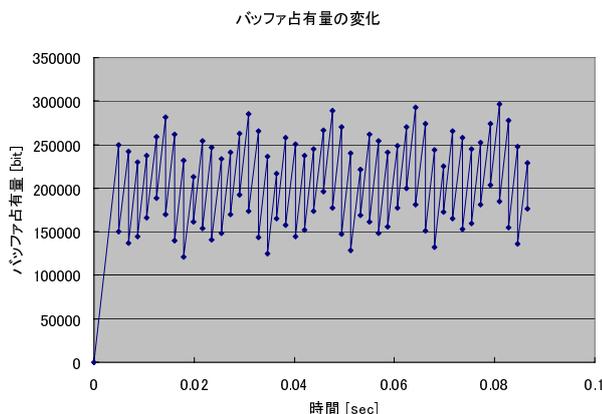


図 9 バッファ占有量 (Europeanmarket)

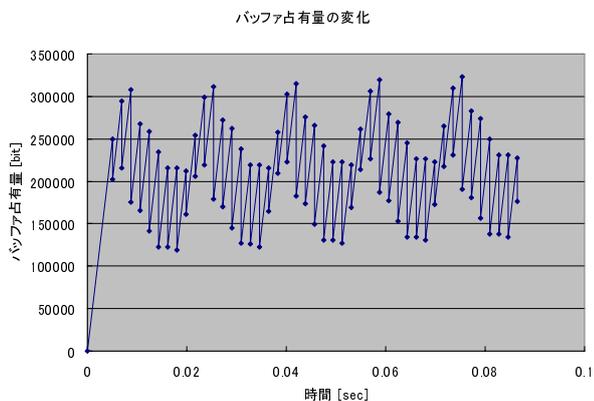


図 10 バッファ占有量 (Openingceremony)

### 3.7 今後の課題

本提案手法では、各Precinctのビット配分を行うために、前フィールドのスカラー量子化後の各Precinctの符号量割合を参照した。そして、スカラー量子化後の符号量割合から、各Precinctへ割り当てるビット量を算出することが可能であることを示した。しかし、スカラー量子化による符号量制御は、ポスト量子化によるレート制御と比較して精度が落ちる。そこで、今後は、ポスト量子化による符号パスの切捨て点情報も考慮に入れることで、精度の高いビット配分を検討する必要がある。

## 4. ま と め

本稿では、HDTV解像度の映像素材をJPEG2000のPrecinct構造を利用することによって低遅延かつ境界歪の生じない符号化手法を提案した。その上で、問題となる各Precinctへのビット配分、そしてバッファリングモデルについて中心に検討した。提案手法により、理想的なビット配分に近い結果が得られることがわかった。さらに、提案手法に基づいたバッファを設計することによって、低遅延で固定ビットレートで伝送可能であることを示した。

今後の課題として、ポスト量子化処理を考慮した、より精度の高いビット配分を実現させることが挙げられる。

## 文 献

- [1] ISO/IEC JTC1/SC29/WG1 (ITU-T SG-8) N1646R Coding of Still Pictures: "JPEG2000 Final Committee Draft Version1.0," Mar,2000.
- [2] A. Descampe, F Devaux, "A Flexible, Line-Based JPEG2000 Decoder for Digital Cinema," IEEE Region 8 Student Paper Contest, 2004
- [3] D. Chaver, M. Prieto, L. Pinuel, F. Tirado, "Parallel Wavelet Transform for Large Scale Image Processing," International Parallel and Distributed Processing Symposium, 2002
- [4] DHS2000, <http://teal.kddilabs.jp/DHS2000/>
- [5] K. Berkner, E.L. Schwartz, "Removal of Tile Artifacts Using Projection Onto Scaling Functions for JPEG 2000," Proceedings of IEEE ICIP'2002, Rochester New York, 2002.
- [6] M. Hashimoto, K. MATSUO, A. KOIKE, "JPEG2000 encoder for reducing tiling artifacts and accelerating the coding process," International Conference on Image Processing, 2003, MP-L1.7
- [7] R. Rosenbaum, D. Taubman, "Merging images in JPEG2000-domain", IASTED-VIIP2003, Benalmadena/Spain, September 08-10, 2003
- [8] J. C. Rountree, T. J. Flohr, M. W. Marcellin, "Report on core experiment codeff5: scan-based processing mode," ISO/IEC JTC1/SC29/WG1 N1594, March, 2000.
- [9] T. J. Flohr, M. W. Marcellin, J. C. Rountree, "Scan-based processing with JPEG-2000," Proc. of SPIE, Appl. of Digital Image Proc., pp 347-355, July, 2000.
- [10] 田邊, 石川, 渡辺, 富永, "Motion JPEG2000 映像配信のためのマルチ解像度変換法" 情報処理学会研究報告, vol.2004, No.25, Page109-114, 2004
- [11] 小野, 渡辺, "国際標準画像符号化の基礎技術" コロナ社, 1998