J-076

JPEG2000 Precinct 構造による低遅延 HDTV 符号化方式

Low delay HDTV coding based on JPEG2000 precinct structure.

佐野　雄磨† 　内藤　整‡ 　渡辺　裕†
Yuma SANO† 　Sei NAITO‡ 　Hiroshi WATANABE†

1　はじめに

近年，ディ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，例えば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．これまでに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化歪みや，符号化効率の低下という問題点があった．そこで本稿では分割の単位をタイルではなく，Precinct を用いることによって符号化する手法について検討する．

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．このタイル化歪みの問題に対して，JPEG2000 の Part2 では，タイルをそれぞれ 1 サンプル分重複させることによって歪みの改善を図る SSO-WT(Single Sample Overlap Wavelet Transform) という方法が定められている．

3　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 [2]．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 [3][4]．

4　タイルを用いた符号化遅延低減

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，入力画像の読込みであるといえる．例えば MPEG など通常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込みが完了するのに最低でも 1 フレーム時間 (約 33msec) 待たされることになる．これに対して，タイルを用いて入力画像を空間的に分割し，読み込んだタイル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．

4.1　実験

HDTV 解像度の画像をタイル分割を行って符号化する実験を行う．今回は，標準画像 OpeningCeremony(1920x1080) を入力画像とするが，最終的なターゲットは動画なので，1 フィールド (1920x540) を符号化の対象と考える．これを垂直方向にタイル分割し，並列符号化処理を行うことを考える．まず、分割タイルサイズと符号化ビット数の関係をシミュレーション実験により調査した結果を図 1 示す．ビットレートは 0.8〜1.6bpp(50〜100Mbps) を想定しており，タイルサイズは走査線数で示した．

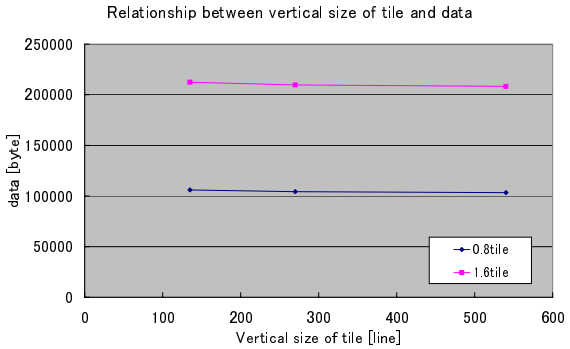

図 1　画像読込みラインと符号量の関係

図 1 からわかるように，画像を分割すれば，少ないライン数を読み込んだ時点でその領域の符号化を開始でき，低遅延化に有効であるものの分割損としてヘッダ量が増加してしまう．図 1 が示すとおり，これによる効率低下は無視できるレベルであるが，分割した領域が独立して符号化されることにより，低ビットレート符号化時には境界歪みが顕著となる可能性がある．そこで，この境界歪が生じないような手法が必要である．

5　Tile-Precinct 変換

4.1 のように，タイル分割を行うことによって，符号化遅延を低減することが可能である．しかし，タイルは境界部分の不連続性による歪を伴う可能性がある．Part2 の SSO-WT のようにサンプル数を重複させれば、境界歪は低減することが可能だが符号化効率が悪化する．そこで，低遅延符号化と不連続性除去の二つを両立させる改善手法として，本稿では “Tile-Precinct 変換” を提案する．

† 早稲田大学大学院 国際情報通信研究科
‡(株)KDDI 研究所

5.1 変換方法

まず，符号化遅延を低減するために，タイルに分割して独立に並行して符号化する．その後，符号化の終了したタイルのコードストリームを，タイル符号化による構造から Precinct を単位としてパケット化した構造に変換する．そして，変換したコードストリームを受信側へ伝送する．受信側では，受信した各コードストリームを合成して復号する．これにより，ミラーリング処理を用いて符号化された JP2 ストリームが，受信側ではミラーリング処理を行わずに復号される．言い換えれば，各タイルで独立に符号化されたものが，お互いのタイルを参照して復号されることになる．この “Tile-Precinct 変換” をブロック図で表したものを図 2 に示す．

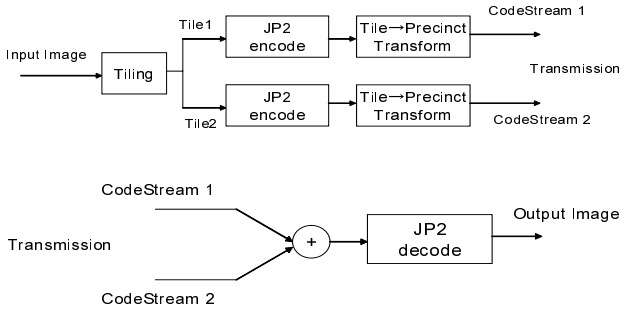

図 2 “Tile-Precinct 変換” におけるエンコーダ（上），デコーダ（下）のブロック図

5.2 変換実験

今回は，“Tile-Precinct 変換” について lena(512x512 24bpp) を用いて実験を行った．9x7 フィルタで 3 段階ウェーブレット変換を行い，ビットレートを 0.3bpp として量子化を行った．タイルサイズを 512x256 としたタイル符号化による画像と “Tile-Precinct 変換” を行った画像の境界付近の様子を図 3 にそれぞれ示す．

図 3 通常の Tile 符号化 (PSNR:32.155dB)（左） Tile Precinct 変換 (PSNR:31.192dB)（右） の画像比較

タイルで符号化した画像の PSNR は 32.155dB で，“Tile-Precinct 変換” の画像の PSNR は 31.192dB であった．“Tile-Precinct 変換” は通常のタイル変換に比べて若干 PSNR は低下するが，図 3 から，境界付近 (画像中心部) の歪に関しては視覚的に良好になっている様子がわかる．さらに，“Tile-Precinct 変換” の影響を検証するために，タイルで符号化したものと “Tile-Precinct 変換” を行ったものそれぞれのついて，原画像との差分をとって比較した．図 4 はそれぞれの境界付近の差分情報を示したものである．

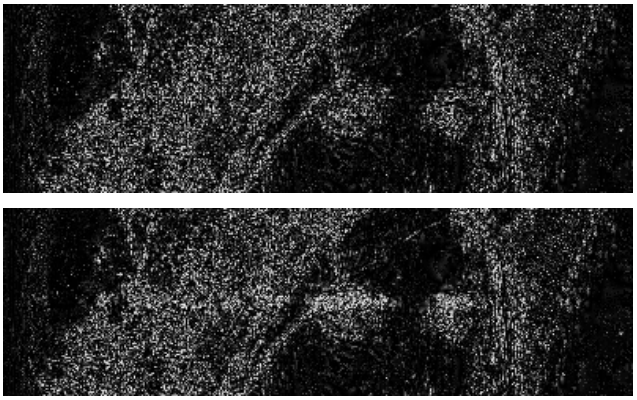

図 4 通常の Tile 符号化の差分（上），Tile Precinct 変換の差分（下）の画像比較

図 4 からわかるように，“Tile-Precinct 変換” は，本来参照していない画素を参照してウェーブレット逆変換を行い復号するので，普通の Tile 符号化を行った場合と比較して境界付近に原画像との差分が多く存在している様子がわかる．しかし，逆にこの境界付近に分布している誤差によって不連続性は低減され，その誤差も視覚的にはほとんどわからない．タイル符号化において，各タイルは独立に符号化するが，元々繋がっていた画像なので当然その相関性は高い．従って，ミラーリング処理を行って復号するよりも，タイル符号化されていても，他タイルの画素情報を用いて復号した方が不連続性の除去には有効であることがわかった．

6 まとめと今後の課題

本稿では，JP2 符号化においてタイル構造を利用することにより，HDTV 解像度の画像について符号化遅延を低下できることを再確認した．さらに，タイルを用いて符号化した JP2 ストリームに対して，“Tile-Precinct 変換” を行うことによって，タイル化特有の境界歪を低減できることを示した．また，“Tile-Precinct 変換” を行うことによって，Precinct 構造のストリームを得るため，これをさらに他のプログレッションに変換可能であり，スケーラビリティの上でも非常に有用であると言える．今回は，タイル構造ストリームをタイル化を行っていないものとして復号したが，復号の際に境界付近のウェーブレット係数に対して適切に重み付けすることによって，境界付近の誤差は低減できると予想される．

参考文献

[1] ISO/IEC JTC1/SC29/WG1 (ITU-T SG-8) N1646R Coding of Still Pictures: “JPEG2000 Final Committee Draft Version1.0, ” Mar,2000.

[2] R. Rosenbaum, D. Taubman, ”Merging images in JPEG2000-domain”, IASTED-VIIP2003, Benalmadena/Spain, September 08-10, 2003

[3] 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.

[4] 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.