

## JPEG2000 ビューアの階層的復号機能に関する検討

## A study on layered decoding method of JPEG2000 viewer

松尾 賢治†  
Kenji MATSUO

橋本 真幸†  
Masayuki HASHIMOTO

小池 淳†  
Atsushi KOIKE

## 1. まえがき

現在, Web ページ用の画像符号化方式には JPEG が幅広く使用されている. しかし, JPEG は Web ページ画像の閲覧に適した画像符号化方式であるものの, 特に医療画像診断や電子チラシ広告および中古車情報サイトなど, 拡大表示や部分表示を頻繁に要求されるインタラクティブな画像閲覧が求められる分野では有効な符号化方式ではなかった. これに対して, 近年国際標準規格化された JPEG2000 は, 従来の JPEG の問題点を克服し, 1 つの符号化データから部分画像の表示およびサムネイル画像の表示を行える符号化方式である. これは, JPEG2000 の符号化処理がタイル単位で行われ, またその符号化データが解像度および品質等に関して階層構造を有しているためであり, これらの特徴を応用することによって, 特に高解像度の画像閲覧を効率良く行えると考えられる. しかし, タイル単位処理および階層化特徴の実際の応用方法は JPEG2000 の規格外であるため, 高解像度画像の閲覧方式はその実装方法が課題となっていた. したがって, 本論文では, JPEG2000 を利用したインタラクティブな高解像度画像表示方法の一実現を目的として, 要求に応じて部分表示および拡大表示が高速かつ容易に行える高効率画像表示方式を提案する.

## 2. JPEG2000

JPEG2000 は直交変換方式にサブバンド分割を再帰的に繰り返す Wavelet 変換を採用している. そのため, 解像度レベルごとに符号化処理が行われ, その結果, 解像度に関して階層的な構造を持った符号化データが作成される. また, JPEG2000 では画像をタイルと呼ばれる小領域単位に分割し, そのタイル画像単位で独立した符号化処理を行える. したがって, 図 1 に示す通り, JPEG2000 の符号化データは各タイルに関して解像度レベルごとに階層化された構造を有している. また, このような階層構造を持った JPEG2000 の符号化データは JPEG とは違って, 符号化データ全体を復号すること無く, 関心のある領域の画像を任意の解像度で効率良く復号表示できるという特徴を持っている. したがって, 機器の表示解像度に応じて, またユーザの要求に応じて, 関心のある領域の最適な解像度の画像を高速に表示することが可能である.

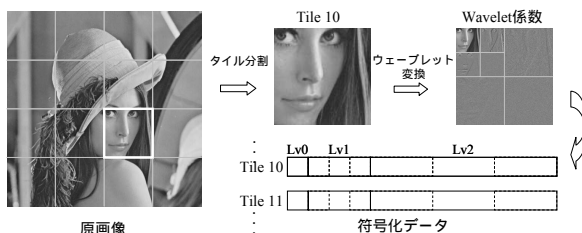


図 1. JPEG2000 符号化データの階層構造

## 3. 提案方式

JPEG2000 の特徴を利用したインタラクティブな画像表示方式を提案する. また, その実装上の問題点を検討して, 提案方式を実現する画像ビューアを開発する. 前記のように, JPEG2000 の符号化データは階層構造を有するという特徴から, 解像度レベルを任意に指定して画像を復号表示することができる. ここでは, インタラクティブな画像閲覧の観点から, 一度ある解像度レベル  $L_v$  の画像を表示してから, さらに 1 つ上位の解像度レベル  $L_{v+1}$  の画像を拡大表示する場合について考える. それぞれの復号処理を独立な処理として扱う場合, 両方の処理において解像度レベル  $L_v$  の符号化データに対する復号処理の一部を重複して行う必要がある. これでは, 拡大表示を多用する画像閲覧を行う場合, 処理効率が悪く, 画像ビューアとしてのインタラクティブ性に欠けることとなる.

したがって, 本論文では拡大表示および部分表示などのインタラクティブな画像表示を行うための要素技術として, JPEG2000 符号化データの差分デコード方式を提案する. 図 2 の概要に示した通り, 差分デコード方式は一度ある解像度レベル  $L_v$  の画像を表示してから, さらに 1 つ上位の解像度レベル  $L_{v+1}$  の画像を拡大表示する場合, 上位解像度レベル  $L_{v+1}$  と下位解像度レベル  $L_v$  との差分の符号化データだけを復号処理し, その他の係数情報は, 既に復号済みの解像度レベル  $L_v$  の係数情報に付け加えることによって, 復号処理の重複を避ける方式である. 差分デコードを実現するためには, 既に復号処理された下位解像度レベルの係数情報を蓄積しておく必要がある. ここで JPEG2000 の復号処理の中で, 係数情報を蓄積できる場所としては次の 3 つの方法が考えられる. 1 つ目は, 符号化データそのままの状態に蓄積しておく方法である. この方法では, 差分デコードを行わない従来の方式とほぼ等価となり, 復号処理の重複を避けることはできない. 2 つ目は, RGB の復号画像の状態に蓄積しておく方法である. この方法では, もう一度 Wavelet 変換を行う必要があるため, 長い処理時間を要する. 3 つ目は逆 Wavelet 変換直後の YUV 係数の状態に蓄積する方法である. 表 1 に 1/10 圧縮された JP2 ファイルの各処理部における復号処理時間を示す. Wavelet 変換処理に全体の約 74% の処理時間が必要であるため, 逆 Wavelet 変換処理の重複を避けることが有効である. した

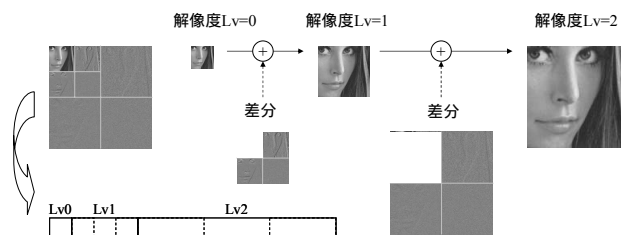


図 2. 差分デコード方式

表 1. JPEG2000 復号処理時間の内訳

エントロピー復号	8.9 %
逆量子化	2.4 %
逆ウェーブレット変換	73.7 %
YUV→RGB 色変換	5.6 %
その他	9.4 %

表 2. メモリ使用量

解像度	理論値
Lv=2	960 KB
Lv=3	3.75 MB
Lv=4	15 MB
Lv=5	60 MB

表 3. 差分デコード方式の復号処理時間 [msec]

提案方式(差分デコード有り)				
解像度	→ Lv2	Lv2→Lv3	Lv3→Lv4	Lv4→Lv5
処理時間	524.8	843.5	2071.8	6638.2
積算時間	(524.8)	(1368.3)	(3440.1)	(10078.3)
提案方式(差分デコード無し)				
解像度	→ Lv2	→Lv3	→Lv4	→Lv5
処理時間	502.9	918.6	2565.4	8096.2
積算時間	(502.9)	(1421.5)	(3986.9)	(12083.1)

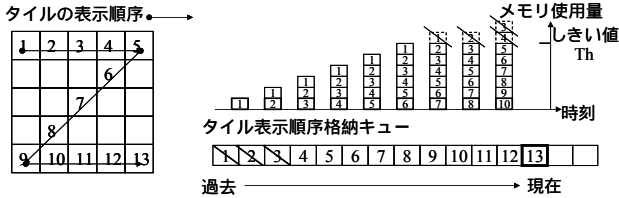


図 3. 蓄積係数の解放機能

がって、下位解像度レベルの係数情報は逆 Wavelet 変換直後の YUV 係数で保持することとする。

しかしながら、実装の観点から、下位解像度レベルの係数を逆 Wavelet 変換直後の YUV 係数で蓄積する場合、解像度レベルの上昇および閲覧済みタイル画像の個数の増加により、蓄積のためのメモリ使用量が飛躍的に増加し、復号装置のパフォーマンスが低下するという問題があった。例として、表 2 に解像度 2048×2560 のカラー画像の YUV 係数を 4 バイト float 型で蓄積する際に必要な使用メモリ量を記す。したがって、メモリ使用量を飛躍的に増加させないように、メモリ使用量が設定されたしきい値を上回った場合、蓄積係数を解放する機能を付け加える。ここで、蓄積係数の解放はタイル単位で行い、タイル内の全ての解像度レベルの蓄積係数を解放することとする。また、現在の画像表示位置の近傍にあるタイルは、次時刻に閲覧される可能性が非常に高いため、近傍のタイル画像の蓄積係数を解放するべきではない。したがって、解放対象とするタイルは、現在の表示位置から最も離れた場所にあるタイルであることが望ましく、それは最も過去に閲覧したタイルである可能性が高い。そのために、過去に閲覧したタイル番号を、キューに蓄積しておき、キューからは先入れ先出しの順にタイル番号を取り出すことによって、最も過去に閲覧したタイル番号を取り出すことができる。この蓄積係数解放アルゴリズムにより、効率良くメモリ解放を行え、係数蓄積のためのメモリ使用量が一定に保たれる。またしきい値を装置の使用可能メモリ量に比例した値に設定することによって、搭載メモリ大小によらず、全ての機器で本提案画像閲覧方式が実現可能となる。

#### 4. 提案方式の実装およびシミュレーション

提案方式の有効性を示すために計算機シミュレーションを行った。JJ2000 Ver4.2 を使用して、差分デコード方式とメモリ解放機能を実装し、Java アプレット形式の画像ビューアを作成した。まず、差分デコードの実装の有無による有効性を比較するために、Pentium4 2.8GHz の計算機を使用し、解像度 2048×2560 のカラー画像 bike を、タイル化無しで色変換を行って分解数 5 の 1/10 圧縮にして作成した JP2 ファイルの復号処理時間を測定した。その結果を表 3 に示す。表より、下位解像度レベルの符号化データが既に復号済みであった場合は、重複する復号処理が最大限に省かれた差分デコード方式がより高速に表示できることが確認された。したがって、拡大表示を繰り返すインタラクテ

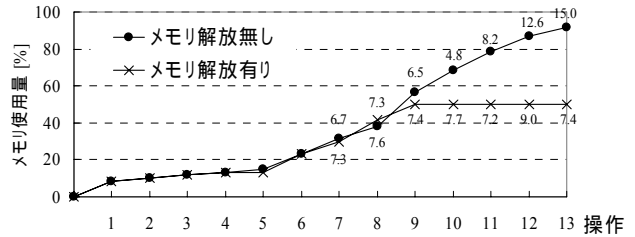


図 4. 蓄積係数の解放機能

ィブな画像閲覧を行う場合は、差分デコード方式によりトータルの復号処理時間を短縮できる。

次に、メモリ解放機能の実装有無による比較を行った。その結果を図 4 に示す。8192×8192 の巨大地図画像を 512×512 にタイル化し、分解数 5 で 1/10 に圧縮して JP2 ファイルを使用した。縦軸はメモリ使用量の全体に対する割合を示す。操作 1~5 で解像度レベル 1 から 5 までの拡大表示を行った後、操作 6~13 はラスタースキャンの順でタイルを閲覧した結果で、グラフ中の記載値は各操作の復号処理秒数を示す。しきい値をビューア全体の使用可能メモリ量の 50% に設定した場合、操作 9 以降でメモリ解放機能により蓄積係数が効率良く解放され、ビューアのメモリ使用量がしきい値以下の一定値に保たれていることが確認できる。また解放後の処理時間は、蓄積係数が解放されたタイルを表示するためには、もう一度最下位の解像度レベルから復号処理を行う必要があり、余分に処理時間がかかると考えられるが、今回の実験のタイル閲覧順序では、蓄積係数解放済みのタイルを閲覧する操作が含まれていなかったため、処理時間に解放処理前後で差異が無かった。一方、メモリ解放機能無しの方式は、操作 15 および 16 で搭載メモリの 90% を越すメモリが使用されているため、メモリスワップが発生し、処理時間が長くなるという結果を得た。

#### 5. まとめ

JPEG2000 のタイル分割処理と階層構造を利用し、拡大表示と部分表示をインタラクティブに行える画像復号表示方式を提案した。また、提案画像表示方式を実現する画像ビューアを試作し、シミュレーションにより有効性を確認した。その結果、提案方式は、下位解像度レベルの係数を利用した差分デコード方式とその蓄積係数を効率良く解放するアルゴリズムにより、要求に応じた表示および高速な画像閲覧が可能な方式であることを確認した。

謝辞 本研究は通信・放送機構(TAO)の受託研究「インテリジェント映像技術の研究開発」に基づいて行われた。

参考文献 [1] ISO/IEC 15444-1, "Information technology – JPEG2000 image coding system – Part 1: Core coding system", ISO/IEC JTC 1/SC 29 WG1, Jan.2001.