

## JPEG 2000 の版管理応用への適用性検討

西川 裕規<sup>†</sup> 原 潤一<sup>†‡</sup> 尾上 孝雄<sup>†</sup>

<sup>†</sup> 大阪大学大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

<sup>‡</sup> (株) リコーGW 開発センター第五開発室 〒107-8544 東京都港区南青山 1-15-5 リコービル

E-mail: <sup>†</sup> {nishikawa.yuuki, jun, onoye}@ist.osaka-u.ac.jp

あらまし 本稿では更新された画像をバージョン管理しサーバに保存する際、バージョン間の差分画像を利用することにより記憶容量やデータ転送量を削減する版管理アーキテクチャを提案する。特に、画像を圧縮保存する方法として JPEG 2000 を応用することを検討する。差分を取らない場合と 3 種類の差分手法を用いた差分画像について、他の画像圧縮形式と記憶容量について比較評価した。この結果、差分手法を選択することで、JPEG 2000 が他の差分方法に対して最大で 7.28 倍の高い圧縮性能を達成した。

キーワード JPEG 2000 版管理 差分画像 静止画像 PNG JPEG-LS ZIP

## Applicability of JPEG 2000 to Revision control of still image

Yuuki NISHIKAWA<sup>†</sup> Jun'ichi HARA<sup>†‡</sup> and Takao ONOYE<sup>†</sup>

<sup>†</sup> Graduate School of Information Science and Technology, Osaka University

1-5 Yamadaoka, Suitashi, Osaka, 556-0871 Japan

<sup>‡</sup> 5th Development Department, Controller Development Center, MFP Business Group,

RICOH company, Ltd. 1-15-5 Minamiaoyama, Minatoku, Tokyo, 107-8544 Japan

E-mail: <sup>†</sup> {nishikawa.yuuki, jun, onoye}@ist.osaka-u.ac.jp

**Abstract** Revision control system of still images is proposed, in which the memory capacity and the data transfer through the network can be reduced by using difference image between versions. As for compaction of images, JPEG2000 algorithm is employed mainly because of high coding efficiency and its feature of scalability. Experiments of the proposed system are carried out by using several calculation schemes of difference image. As a result, JPEG 2000 attains at most 7.28 times as high coding efficiency as the other algorithms.

**Keyword** JPEG 2000, revision control, still image, difference image between versions, PNG, JPEG-LS, ZIP

### 1. はじめに

デジタル画像の応用範囲の拡大に伴い、幅広いアプリケーションに対応可能な新たな画像符号化方式への期待が高まっている。このような状況下で、2000 年に JPEG 2000 符号化方式[1]が標準化された。JPEG 2000 符号化方式は解像度や画質においてスケーラビリティを有しており、さらにタイル分割による編集の容易性や汎用性を持つ。そのため監視カメラ[2]やデジタルカメラ、デジタルペーパーで使用されている。近年ではデジタルシネマへの応用も期待されている。

一方、画像メディアの普及に伴い、美術品や蔵書のデジタルアーカイブ、医療画像、衛星の観測画像などの高解像度画像の管理、編集の必要性が高まっている。また、JPEG 2000 符号化方式のワンソースマルチユース機能を利用して、新聞などの画像データをサーバに蓄積し、そのデータをクライアントの環境に合わせた解像度や画質を用いて送信するサービスが存在する。

現在、一般的な画像の管理手法は、更新した画像を

特定の画像圧縮方式で順次保存、蓄積していくものである。この手法では、記憶容量やデータ転送量は画像圧縮形式に依存する。さらに、複数の解像度表示などが必要な場合、別ファイルとして管理するため、管理システムを複雑にする要因となっている。

一方、独自のアプリケーション内で処理内容を保存管理する特許も存在する[3]。しかしながら、この手法はシステム独自の編集アプリケーションに適用範囲が限定され、外部アプリケーションで作成した画像を管理することができない。このため、複数クライアントで編集、管理するための汎用性に乏しく、実用的ではない。

そこで本研究では、更新される前後の画像について差分を取り、これを利用して記憶容量やデータ転送量の削減を図る版管理アーキテクチャを提案する。さらに、このアーキテクチャにおいて、JPEG 2000 符号化方式を利用することにより管理編集システムの簡易化やデータ転送量の削減、汎用性の向上を目指す。

本稿では特に、更新画像において差分を取らない画

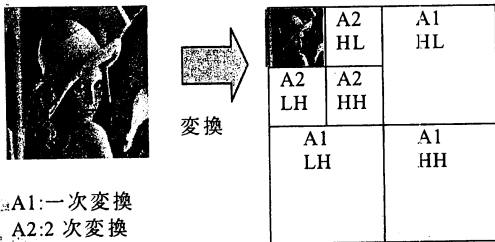
像と差分画像をそれぞれJPEG 2000 符号化方式で圧縮し、これを他の画像圧縮方式と比較評価することによってJPEG 2000 符号化方式の版管理への適用性を評価する。

## 2. JPEG 2000 符号化方式

従来のJPEGでは離散コサイン変換を用いているのに対し、JPEG 2000符号化方式では離散ウェーブレット変換[4]を用いることにより、ブロック歪みを発生させることなくJPEGの約2倍の圧縮効率を実現している。また、JPEG 2000符号化方式では画像の解像度レベルを段階的に上げることができるうえ、タイル領域、プレジント領域、コードブロック領域という小領域に階層的に分類することができるため、スケーラビリティを有している。なお、一般にJPEG 2000符号化方式は低周波成分の割合が多い画像に対して高い圧縮率を示す。

### 2.1. サブバンド分割

JPEG 2000 符号化方式では2次元ウェーブレット変換によって低周波数領域と高周波数領域に分割される。係数は縦、横半分にダウンサンプリングされ、図1のようにサブバンドとよばれる領域に分割される。これにより、JPEG 2000 符号化方式は一つのファイルで複数のサブバンド領域に対応する解像度の画像を得ることが可能となり、スケーラビリティを有している。



- A1:一次変換
- A2:2次変換
- L:低周波成分
- H:高周波成分

図1 離散ウェーブレット変換

## 3. 差分画像を利用した版管理アーキテクチャ

提案する版管理アーキテクチャを図2に示す。このアーキテクチャにおいて、クライアントは、常にあるバージョンの画像を保持しているものとする。以下、画像の蓄積と閲覧についての手順を説明する。

### 3.1. サーバへの蓄積と管理

クライアント側が更新画像をサーバに保存する場合、1~4の手順により、差分画像か更新画像かどちらかを選択し、画像を保存蓄積することによって、記憶容量とデータ転送量を削減することができる。

1. サーバの最新バージョンを確認し、クライアントの保持するバージョンと一致していれば次の処理に進む。そうでなければ3.2節で示す手順で、サーバの保持する最新バージョンの画像を取得する。
2. 1で得た最新バージョンの画像に対し、クライアントの更新画像との差分を取り、これを差分画像とする。

3. 更新画像を転送する場合と差分画像を転送する場合のデータ量を計算し、データ量の小さい方を転送する。同時に、転送する画像が差分画像か更新画像かを示す情報を送信する。
4. サーバ側は受け取った画像が差分画像であれば、差分画像と現在の最新バージョンの画像の加算により得られた画像を最新のバージョンとしてサーバに保存し、差分画像を前のバージョンとして保存する。一方、受け取った画像が更新画像であれば、受け取った画像を最新バージョンとして保存する。

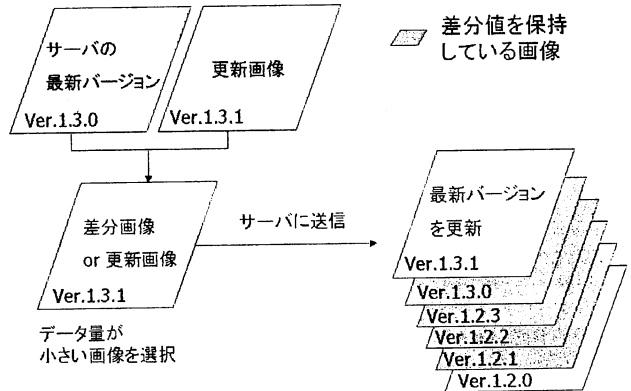


図2 提案する版管理アーキテクチャ

### 3.2. クライアントの画像取得

クライアントから、サーバ上に存在する各バージョンの画像を要求し、取得する手順は以下の1~5のとおりである。

1. 現在クライアントが保有しているバージョンと取得したいバージョンをサーバに送信する。
2. サーバはクライアントが保有しているバージョンよりも要求するバージョンが新しい場合、保有しているバージョンからクライアントの要求するバージョンまでの差分画像のデータ量の合計を計算する。一方、クライアントが保有しているバージョンよりも要求するバージョンが古い場合、要求されるバージョンから最新バージョンまでの差分画像のデータ量の合計を計算する。
3. サーバ上の最新バージョンの画像に対し、要求されるバージョンまでの差分画像を順次用いて、クライアントが要求するバージョンの画像を作成する。また、この画像のデータ量を計算する。
4. 2と3で得られる画像のうち、データ量が少ない画像をクライアントに送信する。同時に、送信する画像が差分画像か更新画像かを示す情報を送信する。
5. クライアントは、送信された画像が差分画像である場合、差分画像を順次用いて要求するバージョンの画像を得る。一方、送信された画像が更新画像であれば、これを要求するバージョンとする。

### 3.3. 版管理における JPEG 2000 の利点

JPEG 2000 は 2.1 節で説明したように、分解レベルに応じた解像度を得ることができるため、複数の解像度レベルの画像を別ファイルで持つ必要はなく、版管理アーキテクチャを簡略化できる。さらに差分を取るにあたって、任意の階調値に対応しているため、差分や編集によって、画像階調値が増加しても、問題がない。

## 4. JPEG 2000 の適用性検討

前節で述べた版管理アーキテクチャにおける JPEG 2000 符号化方式の適用性を以下の点で評価する。

- ・ 差分画像の有効性
- ・ JPEG 2000 の各差分画像における他画像圧縮形式との比較

この版管理アーキテクチャは編集、管理を前提とするため、全ての画像は可逆圧縮で保存するものとする。さらに、複数のバージョンを一覧表示する可能性があったため、サーバは複数の解像度の画像ファイルを持つ必要がある。このことを考慮して、本研究では、原画像に加え、原画像の 1/2 の解像度の画像を同様の方法でサーバに蓄積する。そして、この二つの画像の合計の記憶容量を評価指標とする。

### 4.1. 比較対象となる圧縮形式

版管理アーキテクチャへの JPEG 2000 の適用性を検討するにあたり、差分画像を扱うため、8(bit/pixel)以上の階調値を持ち、可逆圧縮が可能である画像圧縮形式を選択した。以下、各画像圧縮形式について説明する。

#### JPEG-LS

JPEG-LS[5]は有限長ゴロム符号化を利用することにより、比較的少ない計算量で高い圧縮率を実現している。しかしながら、普及率は低い。

#### PNG

PNG[6]は GIF の特許問題に関連して策定されたパテントフリーの画像圧縮形式である。色数の少ない画像に対して高い圧縮率を実現する。また、フィルタを選択することにより、コンポーネント数が多い画像に対して圧縮性能を向上させる効果がある。

#### TIFF

画像データを解像度や、色数、符号化方式の異なる色々な形式で一つのファイルにまとめて格納できるため、編集用の画像として使用されている。単体では圧縮処理は行われない。

#### ZIP

ZIP はスライド辞書法 [7] により、以前に符号化した入力データ列を辞書として使用する圧縮方式である。ある範囲で同じデータ列が多く繰り返される場合に高い圧縮率を実現する。しかしながら、画像を閲覧するには画像のデコードに加えて ZIP 自体の解凍作業が必要となる。

## 4.2. 差分手法

各画像圧縮形式において、8(bit/pixel)のグレースケール画像を原画像とする。また、原画像の 1/2 の解像度の画像を一覧表示用に保存にする。これらの画像の差分計算に以下に挙げる 3 つの手法を用いる。

### 4.2.1. 単純差分

原画像の画素値を  $I_0(x,y)$ 、更新後の画像の画素値を  $I_1(x,y)$  とし、差分画像の画素値を  $D(x,y)$  とすると  $n$ (bit/pixel) の画像の差分を行った場合、 $D(x,y)$  の範囲は以下の式で示される。

$$-(2^n - 1) \leq D(x,y) \leq 2^{n-1} - 2$$

画素値は 0 以上でなければならないため、 $2^n$  を加算することにより、レベルシフトを行う。その結果、(4.1) 式によって単純差分で得られる画素値  $D_0(x,y)$  が計算される。なお、本研究では 8(bit/pixel) の画像を扱うため、 $n=8$  となる

$$D_0(x,y) = I_0(x,y) - I_1(x,y) + 2^n \quad \dots (4.1)$$

上記の式で求められた差分画像は以下のパラメータで保存する。

JPEG 2000...9(bit/pixel)で保存。解像度レベル 2

JPEG-LS...9(bit/pixel)で保存

PNG...16(bit/pixel)で保存。

TIFF...16(bit/pixel)で保存。圧縮なし

ZIP...圧縮なし TIFF を ZIP で圧縮

なお、この差分は、更新画像の階調値が増加しても利用できるという利点がある。

### 4.2.2. xor 差分

この手法は更新前後の画像の画素値において、ビットごとに xor を計算し、この結果を差分値として保存するものである。

原画像の画素値を  $I_0(x,y)$ 、更新後の画像の画素値を  $I_1(x,y)$  とし、差分画像の画素値を  $D_{xor}(x,y)$  とすると、 $D_{xor}(x,y)$  は(4.2)式によって求められる。xor 差分画像では、階調値は変化しないため、すべて 8(bit/pixel) で保存する。また、画像を復元する場合は差分画像と更新画像のビット単位の xor を取れば良い。

$$D_{xor}(x,y) = I_0(x,y) \text{ xor } I_1(x,y) \quad \dots (4.2)$$

### 4.2.3. 絶対値差分

この手法は、差分の画素値を  $n/2$ bit だけレベルシフトし、この絶対値を保存する手法である。ここで述べる絶対値とは、ビット演算における絶対値である。計算された差分値の最上位ビットを切り捨てた値を差分画像の画素として保存する。

このようにして原画像の画素値を  $I_0(x,y)$ 、更新後の画像の画素値を  $I_1(x,y)$  とし、差分画像の画素値を  $D_{abs}(x,y)$  とすると、 $D_{abs}(x,y)$  は (4.3) 式によって求められる。絶対値差分画像では、階調値は変化しないため、すべて 8(bit/pixel) で保存する。画像を復元する場合は、算術シフトを行い、ビット幅を拡張した後に加算演算を行うことで可能である。

$$D_{abs}(x,y) = |I_0(x,y) - I_1(x,y) + 2^{N-1}| \quad \dots (4.3)$$

## 4.3. 対象画像

編集を行う対象画像として、4 種類の画像を用意した。図 3 に各原画像と、TIFF 形式（無圧縮）での記憶容量を示す。

- chart : テキスト、CG、自然画像の混合画像。  
 画像サイズ 1688×2347(pixel)  
 記憶容量 3966 (Kbytes)
- cmpnd : テキスト中心、CG 埋め込み画像。  
 画像サイズ 512×768(pixel)  
 記憶容量 393(Kbytes)
- gold : 自然画像  
 画像サイズ 720×576(pixel)  
 記憶容量 415(Kbytes)
- bike : CG に近い自然画像  
 画像サイズ 2048×2560(pixel)  
 記憶容量 5248(Kbytes)

これらの4種類の画像に対して、写真加工などに用いられる処理として01~10の処理を原画像に加えた。これらの処理を加えた画像を更新画像とし、単純差分、xor 差分、絶対値差分の差分画像を作成する。この差分画像において、JPEG 2000 を各圧縮形式と比較評価する。なお、01~06 の処理は全体処理を行っており、07~10 の処理は部分処理を行っている。

- |             |                 |
|-------------|-----------------|
| 01:移動、右     | 02:移動、右下        |
| 03:回転       | 04:ぼかし、全体       |
| 05:ノイズ      | 06:ぶれ、全体        |
| 07:ぼかし、一部   | 08:CG 埋め込み (左上) |
| 09:文字挿入 (大) | 10:文字挿入 (小)     |

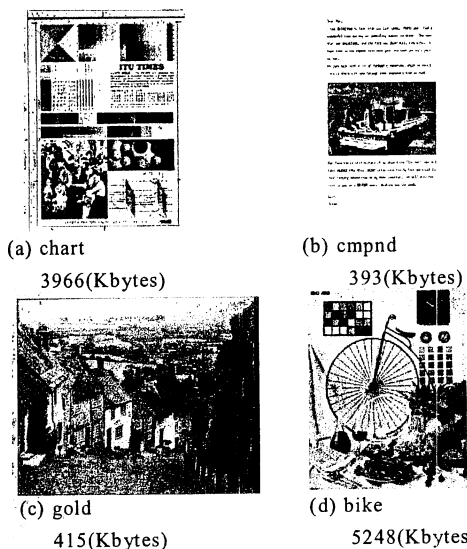


図3 対象となる原画像とその記憶容量

## 5. 結果と考察

前節で用意した二つの解像度レベルの画像を用いて、更新画像に加えてその単純差分、xor 差分、絶対値差分の3手法で差分画像を作成した。これらの4種類の画像の二つの解像度レベルに対して、各画像圧縮形式で画像を圧縮し、合計の記憶容量を比較した。以下、その結果と考察を述べる。

### 5.1. 更新画像

4.3 節で述べた画像について、更新画像をそのまま圧縮した画像の記憶容量を表1、2に示す。但し、表1

は01~06の全体処理についての平均値、表2は07~10の部分処理に対しての平均値を示す。

この結果より、cmpndのようにテキストの多い画像においては、JPEG 2000の圧縮効率が低いことがわかる。それ以外の画像においては、全体的にJPEG 2000の圧縮効率が低いことがわかる。

従って、管理や閲覧を前提とする版管理アーキテクチャにおいて、テキスト画像以外の更新画像をそのまま保存する場合についてはJPEG 2000を用いることで記憶容量を削減することができる。

### 5.2. 差分画像

以下、単純差分とxor差分、絶対値差分の結果と考察を述べる。

#### 5.2.1. 単純差分

単純差分の手法で作成した差分画像の記憶容量の結果を表3、4に示す。但し、表3は01~06の全体処理についての平均値、表4は07~10の部分処理に対しての平均値を示す。

表1と3、表2と4を比較すると、全体処理が加わった画像においては、更新画像される画像をそのまま保存する方が圧縮率が高く、部分処理が加わった差分については差分画像を保存する方が圧縮率が高い傾向がある。

このことより、更新画像をそのまま保存する場合にはJPEG 2000を用いることによって記憶容量が削減できる。また、差分画像を保存する場合には全体処理に関してJPEG 2000を用いることによって記憶容量が削減できる。しかしながら、部分処理を加えた画像に対しての単純差分において、JPEG 2000はJPEG-LSなど他の画像圧縮形式の圧縮効率が及ばないこともある。

#### 5.2.2. xor 差分

xor 差分の手法で得られた各差分画像を圧縮した画像の記憶容量を表5、6に示す但し、表5は01~06の全体処理についての平均値、表6は07~10の部分処理に対しての平均値を示す。

これより、単純差分に比べ、各処理によって画像圧縮形式の性能は上下していることがわかる。これは、xor 差分において、原画像の画素値によって差分の値が変動するため、差分画像における画素値の分布の特徴が変化するためだと考えられる。JPEG 2000の圧縮効率は、表5、6のように、殆どの場合において単純差分よりも低下した。

#### 5.2.3. 絶対値差分

絶対値差分で作成した差分画像の記憶容量の結果を表7、8に示す。但し、表7は01~06の全体処理についての平均値、表8は07~10の部分処理に対しての平均値を示す。これより、単純差分に比べ、各処理によって画像圧縮形式の性能は全体的に向上している。これは、各画像を8(bit/pixel)で保存しているためであると考えられる。また、cmpndにおいては単純差分の手法に比べ最大で7.28倍圧縮効率が向上する。このように、cmpndのような二値画像と写真の混合画像において、絶対値差分を用いるとJPEG 2000の圧縮効率が向上することがわかる。また、表7、8より、他の画像においても、他画像圧縮形式より圧縮効率が低いことがわかる。

### 5.3. 考察

全体の結果より、JPEG 2000 は更新画像をそのまま保存する場合や全体処理が行われた差分画像において、他画像圧縮形式に比べ記憶容量を削減できることがわかる。また、部分処理が行われた差分画像においては、単純差分や xor 差分を用いる場合では JPEG 2000 の圧縮効率は他の画像圧縮形式よりも低下することがわかる。

一方、絶対値差分を用いることで、テキスト画像などにおいて単純差分よりも圧縮効率が最大で 7.28 倍改善し、他画像圧縮形式と比較すると JPEG 2000 は他の画像圧縮形式より高い圧縮性能を示している。

このことより差分画像の管理、蓄積については、絶対値差分を用いた差分画像を JPEG 2000 で保存することが有効であると考えられる。

### 6. まとめと今後の課題

本研究では、差分画像を利用したアーキテクチャを提案し、このアーキテクチャにおける、JPEG 2000 の適用性を評価した。編集を施した更新画像とその差分画像である、単純差分、xor 差分、絶対値差分の 3 種類の差分画像に対して、他画像圧縮形式と比較評価した。その結果、全体処理を行った画像や更新画像をそのまま保存する場合は、JPEG 2000 の圧縮効率が高い。また、部分処理を行った画像に対しては、絶対値差分を用いることによって JPEG 2000 が他の画像圧縮形式よりも圧縮効率が高い。

今後の課題としては、特定の可逆フィルタをかけた後、画像を保存するなどにより、JPEG 2000 の適用範囲を広げて行くことが考えられる。また、これに伴い画像の種類によって、どの差分方法を取るのが適切か決定づける評価関数を見つけることなどが考えられる。

### 文 献

[1]“JPEG 2000 Part 1 Final Publication Draft,” ISO/IEC JTC1/SC29/WG1 July 2002.

[2]Toshihisa Nakai, Noriyuki Sato, “Internet Video and WWW Access from Mobile Terminals – Access Protocol Extension Enabling Real Time Applications –,” No.1, Vol.67, Apr. 2004.

[3]中西 英俊, “画像ファイルのバージョン管理装置” 特許公報(B2), 特許番号第 29994916 号, Dec.1999.

[4]M. Antonini, M. Barlaud, P. Mathieu, and I. Dubieties, “Image coding using the wavelet transform,” IEEE Trans. Image Processing, vol. 2, pp. 205–220, Apr. 1992.

[5]ISO/IEC JTC1/SC29/WG1(FCD 14495-publicdraft), “FCD 14495, Lossless and near-lossless coding of continuous tone still images(JPEG-LS)” 1997.

[6]<http://www.libpng.org/pub/png/>

[7]Jacob Ziv, Abraham Lempel, “A Universal Algorithm for Sequential Data Compression” IEEE Transactions on Information Theory 1977.

表 1 全体処理の更新画像における記憶容量の平均値(Kbyte)

画像名	画像圧縮形式				
	TIFF	PNG	ZIP	JPEG 2000	JPEG -LS
chart	4958	2255	2764	<b>1722</b>	2023
cmpnd	492	<b>115</b>	136	139	145
gold	519	289	393	<b>222</b>	271
bike	6560	3762	4757	<b>2766</b>	3368

表 2 部分処理の更新画像における記憶容量の平均値(Kbyte)

画像名	画像圧縮形式				
	TIFF	PNG	ZIP	JPEG 2000	JPEG -LS
chart	4958	2078	2690	<b>1559</b>	1774
cmpnd	492	92	117	111	<b>91</b>
gold	519	294	407	<b>234</b>	280
bike	6560	3798	4928	<b>2797</b>	3362

表 3 全体処理の単純差分における記憶容量の平均値(Kbyte)

画像名	画像圧縮形式				
	TIFF	PNG	ZIP	JPEG 2000	JPEG -LS
chart	9860	3421	3991	<b>2137</b>	2462
cmpnd	988	<b>166</b>	185	<b>166</b>	173
gold	1040	422	451	<b>302</b>	320
bike	13080	5597	5994	<b>3425</b>	4042

表 4 部分処理の単純差分における記憶容量の平均値(Kbyte)

画像名	画像圧縮形式				
	TIFF	PNG	ZIP	JPEG 2000	JPEG -LS
chart	9980	239	297	<b>139</b>	144
cmpnd	990	31	35	29	<b>27</b>
gold	1046	46	56	71	<b>29</b>
bike	13203	441	507	<b>226</b>	248

表 5 全体処理の xor 差分における記憶容量の平均値(Kbyte)

画像名	画像圧縮形式				
	TIFF	PNG	ZIP	JPEG 2000	JPEG -LS
chart	9860	3122	5399	<b>2692</b>	3166
cmpnd	988	<b>144</b>	225	190	190
gold	1040	418	701	<b>395</b>	434
bike	13080	5404	8876	<b>4689</b>	5601

表 6 部分処理の xor 差分における記憶容量の平均値(Kbyte)

画像名	画像圧縮形式				
	TIFF	PNG	ZIP	JPEG 2000	JPEG -LS
chart	9980	189	304	<b>179</b>	183
cmpnd	990	<b>24</b>	36	33	28
gold	1046	34	59	<b>33</b>	<b>33</b>
bike	13203	353	554	<b>312</b>	343

表 7 全体処理の絶対値差分における記憶容量の平均値(Kbyte)

画像名	画像圧縮形式				
	TIFF	PNG	ZIP	JPEG 2000	JPEG -LS
chart	9860	2838	4915	<b>2235</b>	2581
cmpnd	988	164	202	<b>141</b>	166
gold	1040	344	566	<b>273</b>	334
bike	13080	4552	7482	<b>3520</b>	4225

表 8 部分処理の絶対値差分における記憶容量の平均値(Kbyte)

画像名	画像圧縮形式				
	TIFF	PNG	ZIP	JPEG 2000	JPEG -LS
chart	9980	178	294	<b>128</b>	147
cmpnd	990	23	35	<b>16</b>	23
gold	1046	33	57	<b>28</b>	29
bike	13203	307	541	<b>222</b>	256