

# 劣化型データ埋め込み法におけるファイルサイズの低減に関する検討

齊藤 卓也<sup>†</sup> 姜 錫<sup>†</sup>

<sup>†</sup> 北海道大学大学院情報科学研究科

**概要** 近年、インターネットにおいてデジタルコンテンツの不正コピーや不正流出などがさかんに行われるなどして、その問題が深刻化している。こうした不正行為に対抗するためコンテンツの流通・再生に制限をかけて著作権を管理する DRM(Digital Rights Management) がこれまでにいくつか考案され、注目を集めている。本稿では、そのような DRM 技法の一つとして、画像に対し周波数変換を施した係数の特徴量と正規ユーザ認証のための鍵を用い、原画像の複数の周波数領域にデータを埋め込み劣化させる劣化型データ埋め込み法を取り上げ、この手法において出力される JPEG 画像のファイルサイズを低減する方法について提案し、検討する。

## An Investigation of Filesize Reducing in Degrading Data Embedding Technique

Takuya SAITO<sup>†</sup> Seok KANG<sup>†</sup>

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

**abstract** Recently illegal outflow and copying of digital contents on the Internet has become a serious problem. Digital Rights Management (DRM) has attracted attention as a method to prevent such illegal actions. We proposed a new reversible data embedding technique that degrades the image quality by distorting several frequency regions of the frequency-transformed pattern of an original image using the encoded information on the coefficient characteristics and the key value of legal user's ID. In this paper, we propose and investigate the technique to reduce filesize of JPEG images degraded by former technique.

### 1 はじめに

近年、デジタル回線を利用したインターネットへのブロードバンドアクセスサービスが普及したことにより、映像や音楽などのデジタルコンテンツの配信が可能となり、実際にコンテンツ配信サービスが始まっている。しかし一方ではデジタルコンテンツの特性として、低コストで完全なコピーができ、誤り無しに遠くへ送ることが可能であるという側面から、著作権を有するデジタルコンテンツの不正コピーや不正流出がさかんに行われており、その問題が深刻化している。これらの問題の解決策としてコンテンツの流通・再生に制限をかける DRM 技法が注目されている。DRM 技法には画像にデータを埋め込み、画像の再生に制限を設ける劣化型データ埋め込み法<sup>[1][2]</sup> などがある。劣化型デー

タ埋め込み法は原画像に対して JPEG 符号化と同様に離散コサイン変換し、量子化を行った後に可逆的な変化を付与し、画像を劣化させる。この劣化は、原画像が推測できる程度のものであり、劣化の程度も任意に指定することが可能である。正規認証 Key を持たないユーザは劣化した状態の画像のみを再生することしかできないが、先述の通り原画像がある程度推測できるので劣化したコンテンツをサンプルデータとして楽しむことができる。また、可逆的に劣化させたコンテンツをサンプルデータとして使うことで、本手法におけるコンテンツ復元の特徴から、サンプルデータがそのまま復元されたコンテンツとして使えるため、コンテンツの提供側にとってもユーザ側にとってもデータ通信面で合理的なものといえる。本稿では、劣化型データ埋め込み法に

において出力される JPEG 画像のファイルサイズを低減する方法を提案する。劣化型データ埋め込み法の圧縮率については後に実験で示すが、ブロードバンド時代の今では画像データとしては問題が無いほどのものであるといえる。しかし、この手法を将来に動画へ応用する際、たとえば MPEG 動画などに応用する場合には JPEG と同様のフレーム内符号化を行っている I フレームに適用することが考えられ、その時の動画ファイルサイズの増加は (増加したファイルサイズ) × (動画の GOP 数) となり、およそ 2 時間の動画では、百メガバイト単位で膨らんでしまうことが想定される。したがって現段階でのファイルサイズの低減を行うことが必要であるといえる。

本稿では、希望する程度の劣化を得るために様々な埋め込み位置を設定し、実験データから埋め込み位置とファイルサイズの関係について検討を行う。

## 2 劣化型データ埋め込み法

本章ではまず劣化型データ埋め込み法の流れを説明し、それから詳しいアルゴリズムを記述していく。

これまで説明してきたように、劣化型データ埋め込み法は JPEG のエンコード・デコード手順の中に組み込まれるものである。

まずは、劣化型データ埋め込み法の流れについて記述する。埋め込みについては、

### 1. 埋め込み用データの作成

原画像の DCT 係数列から得られる特徴量 (後述) と、ユーザ認証のための鍵となる文字列 (以降、Key と呼ぶ) からなるデータを埋め込み用データと呼び、これを作成する。

### 2. データ埋め込み位置の算出

埋め込み用データを用い、輝度値の DCT 係数列のどの位置にデータを埋め込むかを算出する。埋め込み位置はブロック毎に計算する。

### 3. データ埋め込み強度の算出

算出した埋め込み位置に対応する埋め込み強度を算出する。

### 4. 埋め込み

算出した埋め込み位置に、対応する強度を加える。

また、抽出・復元についても埋め込み時とほぼ同

様に、

1. 埋め込み用データの作成
2. データ埋め込み位置の算出
3. データ埋め込み強度の算出
4. 復元

算出した埋め込み位置に、対応する強度を引く。

ここで、劣化型データ埋め込み法の特徴としては、埋め込み用データの作成、データ埋め込み位置・強度の算出については埋め込み時と復元時の両方とも同じ手順を踏むということにある。同じ手順を踏むことで、埋め込みデータ、位置、強度は同じものが得られるので復元時には埋め込みの場合と逆の演算をすることにより劣化する前の状態に戻る。流れ図を Fig.1 に示しておいた。また、流れ図における Embedder の部分については Fig.2 に図示する。

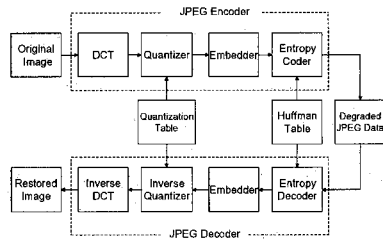


Fig.1 劣化型データ埋め込み法の流れ図

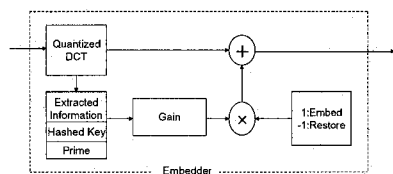


Fig.2 Embedder

## 2.1 埋め込み用データの作成

原画像の輝度値の DCT 係数列から得られる特徴量と、Key を用いて埋め込み用データを作成する。ここでは原画像のサイズが  $N_x \times N_y$  であるものとする。

まず、原画像を  $8 \times 8$  画素のブロックに分割しそれぞれのブロックに対して 2 次元 DCT を施す。次に、量子化パラメータ Quality を用い、JPEG 量子化を行う。量子化された各ブロックをジグザグスキャンし、64 要素の一次元配列へと変換する。このとき、 $n$  番目のブロックの一次元配列を  $F_n(i)$  とする。ここで、 $(i = 0, 1, 2, \dots, 63)$  ( $n = 0, 1, 2, \dots, (N_x \times N_y/64) - 1$ ) である。また 4 ビットのビット列を  $que$  とし、その初期値を 0000 とする。

$F_n(i)$  ( $i = 0, 1, 2, \dots, 63$ ) に対して次の手順を行う。

1.  $que$  を 1 ビットだけ左シフトしたものに  $F_n(i)$  及び  $F_n(i+1)$  が共にゼロ以上または共に負である場合は 0、そうでない場合には 1 を加算する。
2.  $i > 2$  の場合  $que \neq 0000$  である場合は 3. へ進む。そうでなければ  $i$  を 1 増加させ 1. に戻る。 $i = 62$  の段階でも  $que = 0000$  のままであれば 5. に進む。
3. もし  $que$  が直前のブロックで算出された  $que$  と同じならば、直前のブロックの DC 成分を  $que + 1$  を 10 進変換した値で除算し、その余り 4 ビットを 2 進値で表したものを  $que$  とする。そうでない場合は 2. で得たそのままの値を  $que$  とし、4. に進む。
4. 最終的に得られた  $que$  をビット列  $Ft$  の最後尾に付与する。
5.  $n$  を 1 ずつ増加させ、ブロック番号 ( $n = 0, 1, 2, \dots, (N_x \times N_y/64) - 1$ ) に対して 1.~4. を繰り返し計算する。 $n = (N_x \times N_y/64) - 1$  の場合は 6. に進む。
6. 得られた  $Ft$  が 16 バイトの倍数である場合、 $Ft$  の下位 4 ビットを破棄する。その後、 $Ft$  の最後尾に  $Ft$  を付加するという形でこれを繰り返して長さが  $4 \times N_x \times N_y/64$  ビットになるように調節する。

次に、Key の MD5 ハッシュ値をビット列に変換する。得られたビット列は  $Ft$  と同様に長さが

$4 \times N_x \times N_y/64$  ビットになるように調節する。これを  $Key_{MD5}(i)$ , ( $i = 0, 1, 2, \dots, 4 \times (N_x \times N_y/64) - 1$ ) と表すとする。その後、 $Ft$  と  $Key_{MD5}$  の排他的論理和を計算したものが埋め込みデータ列であり、これを  $wm(i)$ , ( $i = 0, 1, 2, \dots, 4 \times (N_x \times N_y/64) - 1$ ) とする。この埋め込みデータ列から、目的となる配列  $Wm(n)$ , ( $n = 0, 1, 2, \dots, (N_x \times N_y/64) - 1$ ) を計算する。 $Wm(n)$  の求め方は以下の通りである。

$$Wm(n) = wm(4n) \times 2^3 + wm(4n+1) \times 2^2 + wm(4n+2) \times 2^1 + wm(4n+3) \times 2^0$$

ここで、 $Wm(n)$  は後に説明するデータ埋め込み位置の算出において除数として使われる。したがって、 $Wm(n) = 0$  だと都合が悪くなる。

そこで  $Wm(n) = 0$  である場合は  $i = 1$  とし、以下の手順を踏んで  $Wm(n)$  の値を置き換える。

・  $n = 0$  の場合

$$Wm(n) = Key_{MD5}(0) \times 2^3 + Key_{MD5}(1) \times 2^2 + Key_{MD5}(2) \times 2^1 + Key_{MD5}(3) \times 2^0$$

・  $n \neq 0$  の場合

$$Wm(n) = Wm(n-i)$$

$$i = i + 1$$

## 2.2 データ埋め込み位置の算出

先述の  $Wm(n)$  を用い、原画像の DCT 係数列  $F_n(i)$  にデータを埋め込む位置を算出する。劣化型データ埋め込み法は画像を可逆的に劣化させることが目的であるため、劣化後の  $F'_n(i)$  は劣化前に比べそれなりに大きく変化していなければならない。しかし、あるブロックにおいて増減される周波数が単一である場合、劣化後の DCT 係数列に不自然な特徴が出てしまう可能性が高くなる。よって、全てのブロックにおいて複数の周波数領域を増減していく必要がある。

$n$  番目のブロックの埋め込み位置群を  $P_n(i)$  ( $i = 0, 1, 2, \dots, 23$ ) とし、埋め込み位置を算出するためのもう一つの鍵を素数  $Prime$  とする。以下の手順で埋め込みデータ  $Wm(n)$  より  $P_n(i)$  を算出する。

1. 除数  $m$  に素数  $Prime$  を  $Wm(n)$  で割った余りを格納し、 $j$  に 24 を格納する。これはブロックの中で 24 カ所に埋め込みを行うということである。

$$m = Prime \bmod Wm(n), j = 24$$

2. 次式を用いて位置  $p$  を算出する。

$$p = \begin{cases} m + j & (m < j) \\ m & (m \geq j) \end{cases}$$

3.  $P_n(i)$  に  $p$  を格納する.  
 $P_n(24 - j) = p$
4. 除数  $m$  に  $Prime$  を  $p$  で割った余りを格納する.  
 $m = Prime \bmod p$
5.  $j$  に  $j - 1$  を格納し,  $j > 0$  の場合 2. に戻る.

### 2.3 データ埋め込み強度の算出

目標とする PSNR をもつ透かしを生成する手法としては画質保証型電子透かし法 [3] などがある. この手法において利用されている, 埋め込みによるブロックの DCT 領域の誤差エネルギーと輝度値の誤差エネルギーはエネルギー保存則が成立し, 等価となる性質を本手法でも利用する. まず, 一つのブロックについて考える. ここで, 目標とする PSNR(以降は  $TPsnr$  と呼ぶ) からどの程度の誤差エネルギー (DCT 領域, 輝度値ともに  $e^2$  とする) を出せば良いか算出する方法について述べる. PSNR を求める式は以下の通りである.

$$PSNR = 10 \times \log_{10} \left( \frac{MAX_Y}{MSE} \right)^2$$

$$MSE = \frac{1}{m \times n} e^2$$

ここで,  $MAX_Y$  というのは輝度値の最高値で本稿では 255 を取る.  $MSE$  というのは平均二乗誤差であり, ここでの  $m, n$  はそれぞれ画像の縦横のサイズ, つまりブロックサイズにあたるので, 両方 8 となる. この 2 式から,  $e^2$  に関する式を求め, さらに PSNR に  $TPsnr$  を代入した式により, 求めるべき  $e^2$  が求まる.  $MAX_Y$  に 255,  $m, n$  それぞれに 8 を代入すると

$$e^2 = 8^2 \times 255^2 \times 10^{-(TPsnr/10)}$$

となる. これは一つのブロックについての結論であるが, [3] では画像全体の結論にも拡張していることから, 提案手法においても有効な考え方であるといえる. ちなみに, 本稿では扱わないが, 様々な画像を劣化させ画像の PSNR を測定した結果, PSNR の値は  $TPsnr$  に近いものが得られたことをことわっておく.

次に, 埋め込み強度の計算の手順を説明する.

1.  $n = 0, k = 0$  とする. また, 全ての  $n, i (i = 0, 1, 2, \dots, 23)$  について  $gain_n(i) = 0$  とする.

$$2. \sum_{i=0}^k (gain_n(i) \times QTable(P_n(i)))^2$$

- が先述の  $e^2$  を超えたら 3. に進む. 超えなければ  $gain_n(k)$  に 1 を加え, その上で  $k$  に 1 を加える. もし  $k$  が 24 以上である場合は  $k$  を 0 にし, そうでない場合には  $k$  はそのまま 2. を繰り返す.
3.  $n$  が  $N_x \times N_y / 64$  の場合, 得られた  $gain_n(i)$  が今回埋め込み強度となる. そうでない場合には  $n$  に 1 を加え,  $k$  を 0 に戻して 2. に戻る.

### 2.4 埋め込み

これまでに計算した位置情報, 強度情報を元に DCT 領域に値の変更を全ての  $n, i$  について下記の要領で行う.

- $F_n(P_n(i)) \geq 0$  の場合  
 $F'_n(P_n(i)) = F_n(P_n(i)) + gain_n(i)$
- $F_n(P_n(i)) < 0$  の場合  
 $F'_n(P_n(i)) = F_n(P_n(i)) - gain_n(i)$

### 2.5 復元について

ここから先は劣化させたデータを復元する手順について説明する. 復元時の手順は途中までは埋め込み時の手順と同様である. まず, 埋め込み時と同様の手順で劣化画像の DCT 係数列およびユーザの入力する Key より埋め込み用データ  $Wm'$  を作成する. 次に, 埋め込み時と同様の手順で  $Wm'$  と  $Prime$  から埋め込み位置列  $P'_n(i) (i = 0, 1, \dots, 23)$  を算出する. さらに, 埋め込み時と同様の手順で埋め込み強度  $gain'_n(i)$  を算出する. その後全ての  $n, i$  について次式を適用し, 劣化の除去を行う.

- $F'_n(P'_n(i)) \geq 0$  の場合  
 $F''_n(P'_n(i)) = F'_n(P'_n(i)) - gain'_n(i)$
- $F'_n(P'_n(i)) < 0$  の場合  
 $F''_n(P'_n(i)) = F'_n(P'_n(i)) + gain'_n(i)$

## 3 提案手法

ここでは, ファイルサイズの低減をはかるため, 埋め込み位置に制限をかける方法を提案する. ファイルサイズを考慮した JPEG 画像へのデータ埋め込み法としては [4] などがある. しかし, 本手法は埋め込み位置の特定を難しくするために画像の特徴

量およびユーザ認証 Key より埋め込み位置を決めて行くため、非ゼロ係数をゼロに近づけるように埋め込む、またはブロックにおける EOB 以降の周波数領域には埋め込まない、非ゼロ係数の領域のみに埋め込むなどの工夫をすることは埋め込み位置の推定を難しくするという劣化型データ埋め込み法における埋め込み位置算出法の目的にかなったものではないといえる。従って本稿では高周波成分に埋め込むように計算されたものを低・中周波成分の部分へ置き換えるという方法をとった。具体的には従来手法の埋め込み位置決定アルゴリズムを次に説明するものとする。ここで、 $p_{MAX}$  は埋め込む位置の最大値で、 $p_{MIN}$  は  $p_{MAX}$  を超える埋め込み位置が現れた時に、それを置き換える位置の最小値である。

1. 除数  $m$  に素数 Prime を  $Wm(n)$  で割った余りを格納し、 $j$  に 24 を格納する。これはブロックの中で 24カ所に埋め込みを行うということである。

$$m = Prime \bmod Wm(n), j = 24, p' = p_{MIN}$$

2. 次式を用いて位置  $p$  を算出する。

$$p = \begin{cases} m + j & (m < j) \\ m & (m \geq j) \end{cases}$$

3. もし  $p$  が  $p_{MAX}$  を超えていたら、 $p = p'$  とし、 $p'$  に 1 を加える。この時、 $p'$  が  $p_{MAX}$  ならば、 $p' = p_{MIN}$  とする。
4.  $P_n(j)$  に  $p$  を格納する。

$$P_n(24 - j) = p$$

5. 除数  $m$  に Prime を  $p$  で割った余りを格納する。

$$m = Prime \bmod p$$

6.  $j$  に  $j - 1$  を格納し、 $j > 0$  の場合 2. に戻る。

上記に置き換える。しかし、高周波領域に割り当てられた埋め込み位置をどの周波数領域へ置き換えるか ( $p_{MIN}$  と  $p_{MAX}$  の設定) が劣化型データ埋め込み法にとって一番良いのかということについては、サンプルデータにするための劣化ということも考慮すると PSNR 意外での効果としての画像の劣化の仕方とでも言うべき部分なども考慮する必要があるが、それについては主観による部分が大きいので今回はファイルサイズの低減率などにより結論を出すものとする。

## 4 実験

SIDBA(Standard Image DataBase) のカラー画像 3 種類 (Lenna, Parrots, Sailboat) について提案手法および従来手法により劣化した JPEG 画像におけるファイルサイズについて目標 PSNR を 10~18, JPEG の Quality を 50, 75, 100 の 3 種類に設定してファイルサイズの圧縮率, 増加分を測定した。

まず、ファイルサイズの低減率を調べるため、JPEG 出力した画像 Lenna ファイルサイズの圧縮率を求め、表にまとめた。Table 1 は、Quality を 50 に、Table 2 は Quality を 75 に、Table 3 は Quality を 100 に設定し、それぞれの Quality について目標 PSNR を 10 から 18 の 9 種類に、3 種類の埋め込み位置 ( $p_{MIN}, p_{MAX}$ ) = (1, 10), (11, 20), (21, 30) に設定し、劣化させた時の圧縮率を表にしたものである。ここで圧縮率は単純に出力 JPEG 画像のファイルサイズを原画像 (BMP 形式) のファイルサイズで割ったものに 100 をかけ、それを 100 から引いたものとする。単位はパーセントである。なお、画像 Lenna を劣化させずに出力した場合の圧縮率は Quality 50 で 97 パーセント、Quality 75 で 95 パーセント、Quality 100 では 71 パーセントであった。

TPsnr	従来	1-10	11-20	21-30
10	88	93	91	90
11	89	93	91	90
12	89	93	92	90
13	89	94	92	90
14	90	94	92	91
15	90	94	92	91
16	91	94	93	92
17	91	94	93	92
18	92	94	93	92

Table1 ファイルサイズ圧縮率 (Quality = 50)

表 (Table 1, 2, 3) の結果から、劣化型データ埋め込み法において、ファイルサイズの圧縮率の低減に関しては JPEG の Quality には関係なく、埋め込み位置によってほぼ決まると言うことが言える。埋め込み位置が低周波成分に近いほどファイルサイズが抑えられるが、どの程度限定したら良いかと言うことに関しては安全面なども考慮に入れ均衡点を探

Tpsnr	従来	1-10	11-20	21-30
10	85	91	88	87
11	85	91	89	87
12	86	91	89	87
13	86	91	89	87
14	86	91	90	88
15	87	92	90	88
16	87	92	90	88
17	88	92	90	89
18	88	92	91	89

Table2 ファイルサイズ圧縮率 (Quality = 75)

TPsnr	従来	1-10	11-20	21-30
10	53	59	57	54
11	56	61	60	57
12	57	62	60	58
13	57	62	60	58
14	58	62	61	59
15	58	63	61	59
16	58	63	61	59
17	59	65	63	61
18	60	65	63	61

Table3 ファイルサイズ圧縮率 (Quality = 100)

る必要がある。また、他の2種類の画像に対してもほぼ同じような圧縮率の低減がみられた。

次に、画像 Lenna に関して、具体的に低減されたファイルサイズを知るため、様々な条件下で出力した JPEG ファイルの増加したサイズをグラフまとめた。Fig.1 は、Quality を 50 に、Fig.2 は Quality を 75 に、Fig.3 は Quality を 100 に設定し、それぞれの Quality について目標 PSNR を 10~18 の9種類、 $(p_{MIN}, p_{MAX}) = (1, 10), (11, 20), (21, 30)$  に設定し、劣化させた時のファイルサイズをグラフにしたものである。

増加したサイズは、劣化させた画像のファイルサイズから劣化させない画像のファイルサイズを単純に引いたものである。

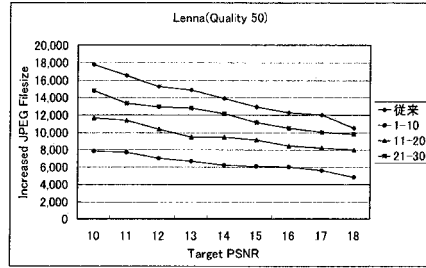


Fig.3 ファイルサイズ増加分 (Quality = 50)

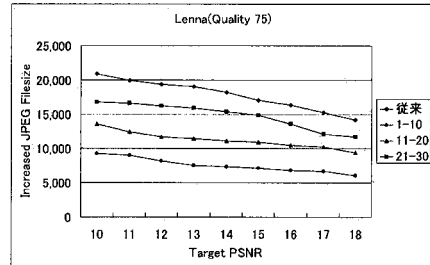


Fig.4 ファイルサイズ増加分 (Quality = 75)

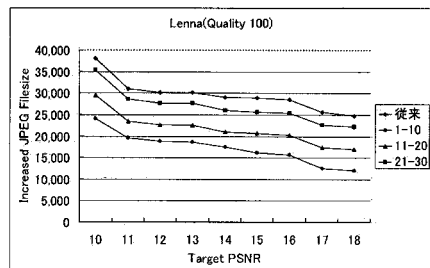


Fig.5 ファイルサイズ増加分 (Quality = 100)

グラフ (Fig.3, 4, 5) の結果からでも, ファイルサイズの増加が低減されたということがわかる. ファイルサイズ増加の低減率は埋め込み位置を置き換える箇所によって異なるが, 今回提案した手法では従来手法のおよそ 20 パーセントから 50 パーセントの低減が見込めると言うことができる.

## 5 むすび

本稿では, 劣化型データ埋め込み法においてファイルサイズが増大する問題点について, 改良する方法を提案・実験し, 検討を行った. 今後は動画ファイルへの応用や効率的な埋め込み位置決定アルゴリズムの考案などが課題として挙げられるだろう.

## 参考文献

- [1] 松本 康佑, 青木 直史, 青木 由直 "ロスレス型電子透かしを用いたユーザー認証と動画への応用", 信学技法, Vol.102, No.632, pp.123-128, Feb. 2003.
- [2] 加藤 幸親, 姜 錫, 青木 由直 "可逆的データ埋め込み技術を用いた DRM 技法", 信学技報, Vol.104, No.647, pp.95-100, Feb. 2005.
- [3] 橘 高志, 藤吉 正明, 貴家 仁志 "放送監視のための非参照型画像電子透かし法", 電子情報通信学会論文誌, D-II, Vol. J86-D-II, No.2, pp.233-241, Feb. 2003.
- [4] 関 祐介, 小林 弘幸, 藤吉 正明, 貴家 仁志 "埋め込み位置の特定を必要としない JPEG 画像へのデータ埋め込み法" 電子情報通信学会論文誌, D-II, Vol. J88-D-II, No.10, pp.2037-2045, Oct. 2005.