

ピクセルブロックによる静止画像データハイディング

1N-11

清水 周一 沼尾 雅之 森本 典繁

日本アイ・ピー・エム株式会社 東京基礎研究所

1 はじめに

デジタル画像をキャリアとして情報を秘密伝送するという先駆的研究 [1] などに続き、近年、データハイディングやマーキングと呼ばれる研究が盛んである。その第一の目的は、デジタルメディアの著作権を保護することであり、電子透かし (Watermarking) や電子指紋 (Fingerprinting) といった機能を実現するための、有望な手法であると期待されている。

画像データは JPEG などの圧縮操作を施して保存されることが多いため、画像を対象としたデータハイディング手法には、DCT 変換など空間周波数を利用した研究 [2, 3] が多い。これらの方法は、圧縮操作後も高精度で情報が抽出できるという利点がある一方で、画質の劣化が大きい、隠すことのできる情報量が小さい、情報抽出のために元画像が必要、などの問題点もある。

本稿では、ビット列として表現できるデジタル情報 (メッセージ) を埋め込むための手法について説明する。上記問題点を解消するために、空間周波数を操作する方法とは異なり、本手法では、画素の特性値を空間的に操作することによりメッセージを埋め込んでいる。

2 画素ブロックの列

本手法では、メッセージはビット情報に分解され、与えられた鍵および分布規則に従って生成された画像データ位置に、順に隠蔽される。ここで、各ビットは、それぞれの位置の指し示す、画像データ上の局所的な部分を微小に更することにより、画質を損なわないように埋め込まれる。

メッセージを取り出すには、埋め込みに使ったのと同じ鍵が必要である。したがって、鍵を秘密にしておけば、メッセージを秘密にできる。また、メッセージの分布系列を生成するための方法を切替えることにより、メッセージを一様に分布させるための方法や、第三者による改ざん操作を防ぐための、公開鍵暗号方式を併用した埋め込み方法 [4] や、すでに埋め込まれたメッセージを壊すことなく、新しいメッセージを追加で埋め込む (追記) ための方法などが可能となる。

3 画素ブロックでの操作

局所的に位置する画素群をブロックの単位として、画素ブロックの列が決定したら、次は、各ブロックに1ビットずつ埋め込む操作を行なう。例えば、ブロックのサイズを 1×1 の1画素とすると、その輝度値が偶数か奇数かでビットを表現することができる。注目している画素が希望のビット値を表していない場合、その画素の LSB (Least Significant Bit) を反転させることで、ビット情報の埋め込みができる。このように、ブロックがビットを表現するためのルールと、そのルールを満たすための画素ブロックの操作とを合わせて、データハイディングの方法が定義できる。

ところで、LSB など輝度値の絶対値を利用する方法は、画像データのフィルタ処理などに非常に弱く、ビット情報は消失しやすい。そこで、各ブロックを二つのサブブロックに分けて、そのペアの特性値の相対的な関係を利用して、ビット情報を表現する手法を以下に2種類説明する。

3.1 一次特性値

自然画像の場合、その隣接する画素は似た輝度値を持つことが多いので、隣接の画素を入れ替えても、画質の劣化にはあまり影響しないことが期待される。そこで、 1×1 のサブブロックを二つ含む 1×2 あるいは 2×1 のブロックの列を決定し、その上でサブブロックの画素の輝度値 (一次特性) を比較して、その大小関係でビット情報を表現する。ビットの埋め込みの際に、比較した結果が目的のビットと一致していない場合には、画素を入れ換えることで、目的のビットを表現することができる。したがって、ビット・ルールは、隣接画素の輝度値の比較で、画素操作は、隣接画素の入れ換えである。ただし、隣接画素の互いに等しい輝度値をもつ場合、ビット情報は表せないため、入れ換え操作は効果がない。このような場合には、入れ換え操作ではなく、一方の輝度値に幾らか足し合わせるなど、画素間に差を付けるための操作が必要となる。

隣接する画素の輝度値に大きな差がある場合、その付近は画像中のエッジ特徴であることが多い。このような画素ペアを入れ換えると、エッジ特徴を壊してしまうため、視覚的に劣化が大きいように感じる。そこで、エッ

ジ特徴を破壊しないような閾値判定が必要となるが、これは、例えば、差が32以下のときに限って入れ換える、というような単純な方法でも、よい結果を示す。

この方法では、2画素で1ビットを表現できるので、大量のメッセージを埋め込むことができる。例えば、768x512サイズの濃淡画像なら、最大24Kバイトのメッセージ内容を隠すことができる。ただし、前述のように画質の劣化を避けるためには、すべてのペアは使うことはできない。

ところで、JPEGなどの圧縮操作に対しては、2画素の関係は高周波成分となることが多いので破壊されやすい。したがって、ブロックのサイズを4x4や8x8などに大きくする必要があるが、そうすると画質の劣化が大きくなり、データハイディング本来の目的を果たせない。このように、一次特性値の場合は、大量のメッセージを埋め込むことができる一方で、圧縮操作に弱いという問題がある。

3.2 二次特性値

複数画素の4x4や8x8サイズのサブブロックでは、輝度値は、平均値や分散値といった二次特性値に分解することができる。前節で述べたように、一次特性値をサブブロック間で交換すると画質は大幅に劣化するが、二次特性値、特に分散値は交換しても、画質にはあまり影響は出ないことが期待される。

二つのサブブロックA、Bの分散値 σ_A, σ_B が計算されたとき、サブブロック間で分散値を入れ換えるには、以下のように画素の操作を施せば良い。次式は、サブブロックAの分散値を σ_B に取り替えるための処理を表す。

$$x_i \leftarrow m_A + (x_i - m_A) \frac{\sigma_B}{\sigma_A}$$

ここで、 x_i はサブブロックAの画素輝度値、 m_A はサブブロックAでの画素輝度値の平均である。

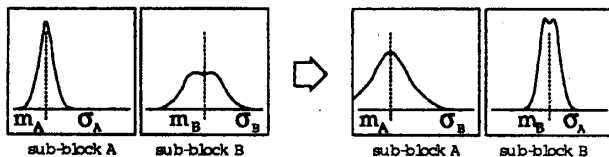


図1: Secondary attribute interchange

上式によれば、図1に示すように、サブブロックの平均値は変化せずに、分散値のみ変更され、正確に入れ換えられる。なお、輝度の分布の形も変化しないので、二つの山に分かれるような分布をするサブブロックでも、それぞれの山が平均値からわずかに離れるか近づくかの変更が行なわれるため、サブブロックの中の内容が大きく変化することはないと期待される。

なお、二つの分散値が同一の値を取る場合には、一次

特性値のときと同じように、二つに幾らかの差を付けてやれば、ビット情報を表現することができる。また、二つの分散値が大きくかけ離れている場合には、交換操作を行わないというような閾値処理で、画質の劣化を抑えることができる。

4 実験および評価

損失のあるJPEG圧縮について、耐性実験を行なった。図2に結果の一例を示す。対象は、768x512サイズのRGB 24-bit画像である。グラフ横軸は、オリジナル画像に対する、JPEG圧縮後のファイルサイズを表し、縦軸は、圧縮後も、ビット情報が正しく保存された比率を表す。グラフ中、“primary”は一次特性値によるハイディング、“deviation”は二次特性の分散値を利用したハイディングでの結果を指す。また、図中“4x4”などはサブブロックのサイズを表す。

グラフによれば、5%にまで圧縮しても、DCTのウィンドウサイズと同じ8x8のサブブロックなら、9割程保存されるので、一割以上を救うエラー訂正符号を使って、メッセージを符号化しておけば、メッセージを正確に取り出すことができる。

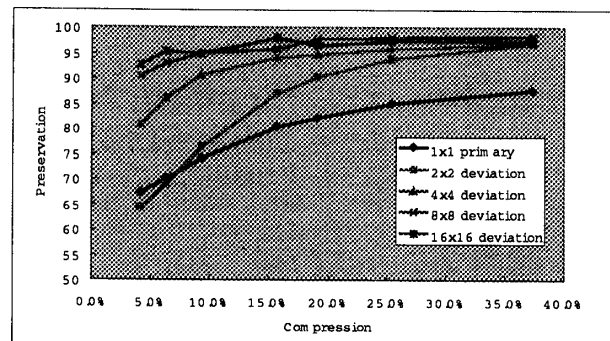


図2: Compression and preservation

参考文献

- [1] K. Tanaka, Y. Nakamura, K. Matsui: "Embedding secret information into a dithered multi-level image", In Proc. 1990 IEEE Military Communications Conference, pp.216-220, 1990
- [2] Ingemar J. Cox, et al.: "Secure Spread Spectrum Watermarking for Multimedia", NEC Research Institute, Technical Report 95-10, 1995
- [3] Jian Zhao, et al.: "Embedding robust labels into images for copyright protection", In Proc. of the Int. Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies, Vienna, August 1995
- [4] 沼尾他: "データハイディングによるデジタル署名技術", In Proc. of IPSJ 96 annual conference, 1996