The 29th Symposium on Information Theory
and its Applications (SITA2006)
Hakodate, Hokkaido, Japan, Nov. 28 - Dec. 1, 2006

SPIHTアルゴリズムにおけるデータ埋め込み法
A Data Hiding Method for SPIHT

作田 豊* 細谷 剛* 平澤 茂一*
Yutaka SAKUTA Gou HOSOYA Shigeichi HIRASAWA

Abstract— We propose a new technique for embedding binary data in coded image by the SPIHT algorithm. The purpose of our algorithm is to suppress degradation in the image quality when we embed hidden data in the image. Since the structure of the output bit stream by using the SPIHT algorithm is maintained, coded data can use the same decoder. Moreover, it is not necessary to specify the position of the embedded data. We show by simulation result that degradation in the image quality with proposed method is quite small.

Keywords— SPIHT,DWT,Data Hiding,Embedding,PSNR

1 はじめに

画像に付加された情報を用いて検索，画質の監視や誤り訂正などを行う技術がある．それに伴いデジタル画像にその画像と独立なデータを視覚的に認知しにくい形式で挿入するデータ埋め込みの研究が多方面で行われている．これらの目的は電子透かしとは異なり，情報埋め込み者が符号化することを前提に非可逆圧縮のもとでも情報を完全に抽出することである．[3] や [5] では，JPEG2000 符号化画像に対して，画像への寄与度の低いレイヤに情報を埋め込んだり複数画素に対して法演算を用いた埋め込みを行うことで画質の劣化を抑えつつ情報の埋め込みを行うことを可能とした．しかし，前者の手法では高い画質を得る為には下位レイヤの破棄が必要であること，後者では量子化後の DWT 係数に操作をするために復号の際に劣化が拡大してしまうなど問題があった．本研究では JPEG2000 と同様にエンベデッド性を持つ SPIHT(Set Partitioning in Hierarchical Trees) アルゴリズムを用いて，これらの問題を解決しつつ画質劣化が少ない情報を埋め込む手法を提案し，シュミレーションによって従来の手法と同量の情報を埋め込んだ際に画質劣化が小さいことを PSNR により示す．

2 SPIHT 符号化

まず準備として，SPIHT 画像符号化 [2] の概要について説明する．更に，具体的なアルゴリズムとその特徴について述べる．

2.1 SPIHT 符号化の特徴

SPIHT の符号化処理の構成を図 1 に示す．SPIHT ではまず，入力画像をウェーブレット変換によりサブバンド分解する．次にサブバンド分解されたウェーブレット係数に対してゼロツリー構成を定義する．その後に Sorting Pass と Refinement Pass と Quantization-Step Update とを繰り返しながらビットストリームを出力していく．ゼロツリーは EZW(Embedded Zerotree Wavelet)[1] によって新たに導入された概念であり，同一周波数方向，同一空間にある係数を木構造として構成したものである．(図 2 参照)

一般にゼロツリーの根となる要素がある値に対して有意 (ある閾値より大きな値) でない場合，ゼロツリーに含まれる要素も有意でないことが多いため，ゼロツリー全体を符号化対象とすることで複数の画素の情報を一度に送信することが可能となっている．

また SPIHT は逐次近似量子化可能なアルゴリズムとなっており，出力されたビットストリームを任意の場所で復号することも可能である．

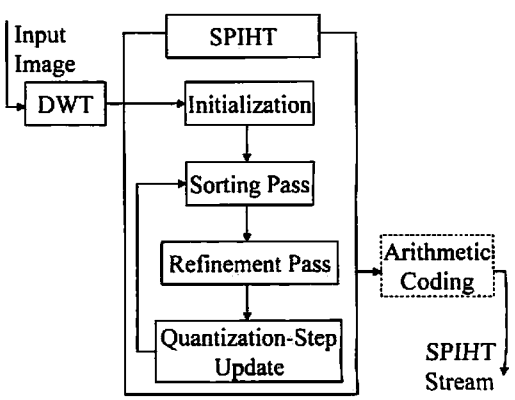

図 1: SPIHT 符号化処理の構成

2.2 定義

ゼロツリーによる関係を以下のように記述する．

- H ：最低解像度の座標集合
- $O(i,j)$：座標 (i,j) を根としたときのゼロツリーの 1 階層下の座標集合
- $D(i,j)$：座標 (i,j) を根としたときのゼロツリーの下の全階層の座標集合
- $L(i,j)$：$D(i,j)$ から $O(i,j)$ を除いた集合

以下にアルゴリズムの実行に用いる判別関数 $S_k(\Gamma)$ とリストを定義する．Γ を画像全体の集合とし $c_{i,j}$ は座標 (i,j) における DWT 係数である．

* 〒 169-8555 東京都新宿区大久保 3-4-1 早稲田大学理工学部経営システム工学科 School of Science and Engineering,Waseda University 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan. E-mail: sakuta@hirasa.mgmt.waseda.ac.jp

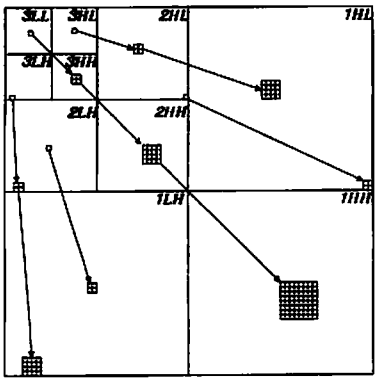

図 2: DWT 変換係数におけるゼロツリー構成

[定義 1：判別関数 $S_n(\Gamma)$]

$$S_k(\Gamma) = \begin{cases} 1 & \max_{(i,j)\in\Gamma}|c_{i,j}| \geq 2^n, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

[定義 2：リスト]

- LIP：有意でない画素値を持つ座標リスト
- LSP：有意である画素値を持つ座標リスト
- LIS：有意でない画素値を持つ座標集合リスト
  (タイプ A とタイプ B の 2 種類を区別する)

2.3 SPIHT アルゴリズム

S1. [Initialization]

1. $n = \lfloor \log_2(\max_{(i,j)}|c_{i,j}|) \rfloor$ を閾値とする.
   (ただし $\lfloor x \rfloor$ は x を超えない最大の整数)
2. LIP：H において子集合を持たない画素値.
3. LSP=$\emptyset$
4. LIS：タイプ A として H において子集合を持つ画素値の集合.

S2. [Sorting Pass]

- LIP の要素それぞれに対して
  1. $S_k(i,j)$ を出力
  2. $S_k(i,j) = 1$ ならば (i,j) を LSP に移動し符号の情報を 0,1 で出力
- LIS の要素それぞれに対して
  1. タイプ A であれば
     (a) $S_k(D(i,j))$ を出力
     (b) $S_k(D(i,j)) = 1$ を出力した場合
     - $(s,t) \in O(i,j)$ に対して $S_k(s,t)$ を出力
       * 1 を出力したならば (s,t) を LSP に追加し符号の情報を出力
       * 0 を出力したならば (s,t) を LIP に追加
     - $L(i,j) \neq \emptyset$ ならば (i,j) をタイプ B に変更して LIS の最後尾に追加
       $L(i,j) = \emptyset$ ならば (i,j) を LIS から消去
  2. タイプ B であれば
     (a) $S_k(L(i,j))$ を出力.
     (b) $S_k(L(i,j)) = 1$ を出力した場合
       i. $(s,t) \in O(i,j)$ を LIS の末尾にタイプ A として追加
       ii. (i,j) を LIS から除去

S3. [Refinement Pass]
現在の n での走査で追加された要素以外の LSP における (i,j) に対して n 番目のビットを送信.

S4. [Quantization Step]
$n = n - 1$ として閾値を更新して
S2. [Sorting Pass] に戻る. □

アルゴリズムからもわかるように，SPIHT は画像に対する寄与度の高いビットの情報を優先的に 2 値のシンボルとして出力するアルゴリズムである.

3 従来手法

SPIHT と同様にエンベデッド性を持った圧縮方式として JPEG2000[4] がある．ここでは，金らの提案する「法演算を用いる JPEG2000 符号化画像への情報埋込法」[5] を述べる．この手法では DWT 後の係数に対して量子化を行いその量子化係数に対して量子化係数を直接整数の範囲で操作することでデータの埋め込みを行う．複数ビットに対して法演算を利用しながら更に量子化誤差を小さくするように埋め込むことで複数画素に対する整数値の埋め込みの際の画質劣化を抑制することに成功している.

3.1 SPIHT への適用

上述したように従来手法の操作は DWT 係数上で行われるため同様の議論が SPIHT でも可能である．従来手法を SPIHT に適用した時の構成を図 3 に示す.

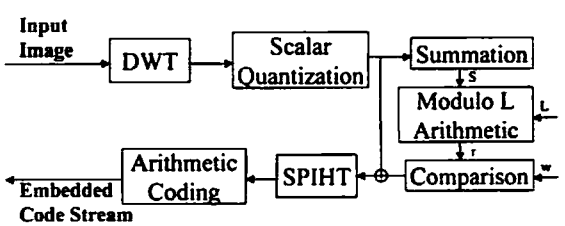

図 3: 従来手法の SPIHT アルゴリズムへの適用

3.2 埋め込みアルゴリズム

従来手法では画像を任意の数の画素値集合に分割して，それぞれの集合に以下に示す操作を行って整数値データの埋め込みを行う．埋め込みアルゴリズム中において，L は埋め込む整数値情報のダイナミックレンジ，w は実際に埋め込む値である．DWT 後の量子化係数に対して以下の手順を行う．次に，図 3 の右側のブロックで実行しているアルゴリズムを示す.

[埋め込みアルゴリズム]

K1. [Summation]

画素値集合内の量子化係数の合計を S とする.

K2. [Modulo L Arithmetic]

$$r = S \bmod L, \quad r \geq 0 \tag{2}$$

K3. [Comparison]

$$g_1 = |w - r| \tag{3}$$

$$g_2 = L - g_1 \tag{4}$$

$$d = \min(g_1, g_2) \tag{5}$$

$$\hat{S} = \begin{cases} S-d, & r > w \quad \text{and} \quad g_1 < g_2 \\ S+d, & r > w \quad \text{and} \quad g_1 \geq g_2 \\ S+d, & r < w \quad \text{and} \quad g_1 < g_2 \\ S-d, & r < w \quad \text{and} \quad g_1 \geq g_2 \\ S, & r = w \end{cases} \tag{6}$$

□

以上のようにして計算された $\hat{S}$ と量子化係数の合計が等しくなるように, DWT 係数を書き換えることで情報の埋め込みを行う. なお, 抽出は $\hat{S}$ を L で割った余りを計算すれば, それが w となる.

4 提案手法

本節では従来手法における問題点を述べ, 提案手法の着眼点, 埋め込みの流れについて説明していく.

4.1 従来手法の問題点

法演算, 量子化誤差の補正により画質の劣化を抑えた従来手法であったが, エンベデッド性を持った符号化方式に組み込む場合

- 任意点に埋め込みが可能であるが, 特定の場所に埋め込む場合にはその場所を記録する工夫が必要になる.
- 量子化後の係数に埋め込みを行うため, 画素値を 1 変化させただけでも復号した際の量子化係数の誤差が大きくなってしまう.

などの問題点が考えられる.

4.2 提案手法の着眼点

SPIHT アルゴリズムでは画像への寄与度が大きいビットほどビットストリームの前方に出力される傾向がある. 逆に後半のビットほどデータが埋め込まれた際の画像への影響は小さいと考えてよい. つまり符号化の終了直前で更新された画素ほど画質への寄与度は低い. そこで, 符号化ビットストリームの後半で確定した画素値に対して優先的に埋め込みを行う. [3] の文献においても同様に JPEG2000 の最下位レイヤを用い, 不要となるビットは復号前に破棄することで効果的な埋め込みを実現しているが, 本研究では単独で従来の復号器のみを用いた場合でも画質劣化に対して優れた耐性を持つ埋め込み方法を提案する.

4.3 提案手法の方向性

SPIHT で出力されるビットストリームのうち

- 符号を表すビット
- リストの操作に関連するビット

は埋め込みに適さない. なぜならば前者は +,− が変化した場合画像に対する劣化が非常に大きくなってしまい, 後者は符号化の流れが変化してしまうと想定した画素に対する復号を正常に行えなくなるためである.

以上を考慮してデータの埋め込みには Refinement Pass で出力されるビットを中心とする. 提案手法では更に Sorting Pass 内でのデータの埋め込みで有意, 非有意の変化があっても符号化の流れが変化するのはその次の閾値での Sorting Pass からであることを利用して埋め込みが可能なビットを拡張する. また, 埋め込みによる画質の劣化を考え SPIHT において出力されるビットストリームの最下位 2 ビットまでを埋め込みの候補とする.

4.4 提案手法の流れ

SPIHT アルゴリズムにおいて以下の 3 つの箇所に埋め込みを行う.

1. LIP リストにおける $S_n(i,j)$ の出力
2. LIS リストにおけるタイプ A での $S_n(s,t)$ の出力
3. Refinement Pass における出力

なお埋め込みは DWT 係数を書き換えることで行い, 符号化や復号化の時の操作順序が変化しないように留意する. そのため Sorting Pass における埋め込みは符号化を打ち切る時の n の場合のみとなる. 更に複数の画素に 1 ビットを埋め込むことが可能なときには, 0, 1 の 2 値となるが従来手法 [5] と同様に法演算を用いながら複数画素に対してビット切捨てによる画質劣化が最も小さくなるように埋め込みを行う. 提案アルゴリズムの構成を図 4 に示す.

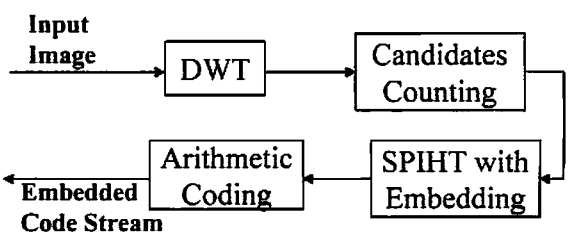

図 4: 提案手法による埋め込みの構成

[提案アルゴリズム]

P1. [Candidates Counting]

通常の符号化と同様の走査のみを行い埋め込みの候補となる画素の数 C をカウントし, 埋め込む情報の合計ビット数を H として以下を計算する.

$$G = \lfloor H/C \rfloor \tag{7}$$

このとき提案手法では G 画素毎に埋め込みを行う．また $G=0$ の時は埋め込み不可能とする．

P2. [SPIHT with Embedding]

1. 通常の符号化と平行し Candidates Counting で行ったのと同様に埋め込みの候補となる画素のカウントを行う．
2. カウントが $C-H\times G$ となった点から埋め込みを行う．
3. すべてのデータを埋め込み終えたところで符号化を終了する． □

埋め込み後も C の値は変化しないので，抽出は埋め込みビット数 H を受信側に知らせておくことで可能である．

5 数値実験

グレースケール標準画像SIDBA（Standard Image DataBAse）[6] の 512 × 512 画像に対して数値実験を行う．DWT において 5/3 フィルタ [7] を用い，変換回数は 5 回とした．また出力系列には算術符号化を適用している．画質に対する評価については N を総画素数として，以下の PSNR を用いる．

$$PSNR = 10\log_{10}(\frac{255^2}{MSE})[dB] \quad (8)$$

ただし

$$MSE = \frac{1}{N}\sum_i\sum_j(c_{i,j}-\hat{c}_{i,j})^2 \quad (9)$$

5.1 実験結果

図 5 に LENA 画像に対して 8,192bit を埋め込むことを考えた時，1 ブロック 4 × 4 画素として従来手法を実行した場合と提案手法を用いた時の bit-rate と PSNR の関係を示す．提案手法では bit-rate が低い（全体の符号量に対して埋め込みデータの比率が大きくなる）と従来手法に比べ特に優れた画質を得られた．また，他の画像に対しても同様の傾向を持つ結果が得られることを確認した．

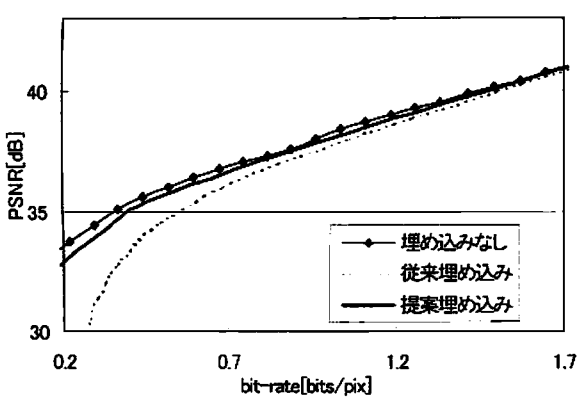

図 5: LENA 画像（8,192bit 埋め込み）の画質比較

6 考察

6.1 提案手法の長所

(1.) 従来手法では，量子化誤差の補償で画像の欠損を抑えたが，1bit の情報の書き換えに対して量子化係数を乗算した分の変化があった．
提案手法では，逐次近似量子化の性質を利用することで，1bit の情報書き換えに対する元画像との誤差の広がりを抑えられた．

(2.) 従来手法と同様埋め込みには DWT 係数を操作するので汎用の復号器が利用可能である．

(3.) 埋め込み位置を特定しなくても，優先的に画像への寄与度が低い bit への情報埋め込みが行われる．

6.2 提案手法の課題

(1.) 埋め込みビットの位置を決定する為に今回のように画素の走査を 2 パスにするか，メモリの消費を大きくする必要がある．

(2.) 特に複数 bit に対して埋め込みを行う場合に最適となる埋め込み方法を探索する際に大きな計算量を必要としてしまう．

(3.) 埋め込み可能画素数に制限がある．

(4.) 複数画素に対して埋め込みが可能になる場合とそうでない場合に画質劣化の抑制に差が生じて bit-rate と PSNR のグラフが完全に滑らかではない．

7 まとめと今後の課題

本研究では SPIHT アルゴリズムの逐次近似量子化の性質を利用することで，画像劣化に対して従来よりも高い性能を持った埋め込み方式が提案した．
今後は複数ビットに対する情報埋め込みが出来ない場合の画質向上や計算量削減などについて改善の必要がある．

8 謝辞

本研究の成果の一部は早稲田大学特定課題研究助成 2005B-189 による．

文献

[1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," IEEE Trans. Signal Processing, vol. 41, pp. 3445-3462,Dec. 1993.

[2] J. M. Shapiro, "An embedded hierarchical image coder using zerotrees of wavelet coefficients," Proc. IEEE Data Compression Conf., Snowbird,UT, 1993, pp. 214-223.

[3] 安藤 勝俊, 小林 弘幸, 貴家 仁志, "レイヤ構造を利用した JPEG2000 符号化画像へのバイナリーデータ埋込み" 電子情報通信学会論文誌,Vol.J85-D- ,No.10,pp.1522-1530,2002

[4] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," IEEE Signal Processing,vol. 18, pp. 36-58, 2001.

[5] 金 弘林, 藤吉 正明, 関 裕介, 貴家 仁志, "法演算を用いる JPEG 2000 符号化画像への情報埋込法" 電子情報通信学会論文誌,Vol.J89-A,No.3,pp.234-242,2006

[6] The USC-SIPI Image Database : http://sipi.usc.edu/database/

[7] ISO/IEC,"The JPEG2000 image coding system Part1-Core coding system," ISO/IEC15444-1,December .