

ハフマン符号を用いた二値画像データへの情報秘匿の提案

日大生産工(院) 大井 政典 日大生産工 伊藤 浩

1 まえがき

電子透かし等の情報秘匿技術は、デジタルコンテンツにその存在がわからないようにデータを埋め込むことでデジタルコンテンツに、特定の情報を記録するための技術である。電子透かしの使用形態として著作権・使用条件の表示等がある [1]。

本研究では、圧縮された画像データに容量を増やす事無く情報を埋め込む方法を提案する。JPEG は現在も広く静止画像の圧縮に使われており、JPEG には可変長符号としてハフマン符号が使われている。そこで二値画像をハフマン符号で符号化したデータに情報を埋め込み、提案方式の基本原理を確認した。

2 情報秘匿の方法

2.1 原理

Fig.1 は本提案による情報の伝達方法を示す。まず、二値画像はハフマン符号で符号化され、符号データになる。次に送信者は鍵を用いて符号データにデータの一部を反転することによって情報を埋め込み、これを受信者に送る。受信者は、鍵を使って復号を行い画像と情報を取り出す。

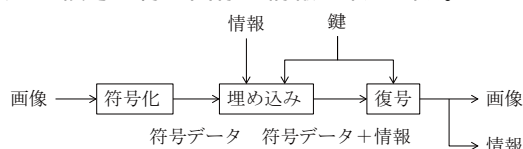


Fig. 1: 情報の伝達方法

情報が伝達される理由は次の通りである。送信者は情報によって異なる場所を反転する。受信者はその場所のデータを反転して元に戻す。これを復号すれば正しい画像と情報が得られる。もし、間違った場所を反転すれば正しい画像や情報は得られない。また、ハフマン符号の復号の同期がずれることがある。このことを利用して、受信者は原画を知らなくても画像の規則性などからどこが反転されたを知ることができるので情報が伝達できる。

Fig.2 は復号時の同期のずれを説明している。シンボル $a \sim d$ に図に示す符号が割り当てられ

ている。この時、元データ列: $adacd$ を符号化すると "01110110111" となる。ここで、3 ~ 6 桁目を反転させると "01001010111" となり、これを復号すると $ababbd$ のように 6 シンボルが復号される。このようにもともと送信したシンボル長と復号したシンボル長が異なる事を同期がずれると言う。

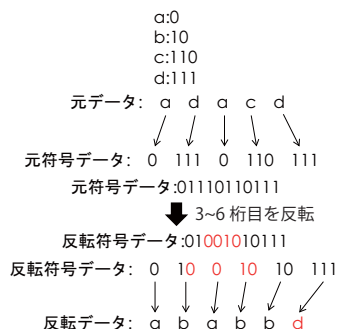


Fig. 2: 例:同期のずれ

2.2 実現方法

簡単な為、1bit の情報を埋め込む場合を説明する。二値画像を 2×2 のブロックに分割し、各ブロックの 16 通りのパターンに対してハフマン符号を割り当てる。この符号で符号化した画像データから $2n$ 桁のビット列を取り出す (n は整数)。どこを取り出すかはパスワードによって決める。秘匿情報 (0/1) が 0 の場合は $2n$ 桁の内、前の n 桁を反転 (前反転)。1 の場合は後の n 桁を反転する (後反転)。これを元の場所に埋め込み相手に送る。以下は $n = 4$ の場合の例である。情報によって前反転の時は元の符号データの中の 0101 を 1010 に反転させ、後反転時は 1010 を 0101 に反転させている。

例 元の符号データ: 0101 1010...

前反転 (0 の場合): 1010 1010...

後反転 (1 の場合): 0101 0101...

次に情報の検出方法を説明する。まず、前 n 桁を反転して復号し、復号画像の隣接するライン間で白と黒に反転している画素数を数える。この値を n_0 とする。また、後 n 桁を反転させて、同じ処

理をし、 n_1 を求める。 $n_0 < n_1$ ならば 0、 $n_0 > n_1$ なら 1 を復号する。画像データには規則性があるので、正しい位置を反転させていれば規則性が元に戻るので個数が減る。逆に間違った位置を反転させると規則性がさらに崩れるので個数が増える。 $n_1 = n_2$ の場合は、同期のずれを確認し、ずれのない方を復号する。これらを利用して情報 (0/1) を判定している。

伝達する情報が k bit の場合は、符号データ中からランダムに $2nk$ 桁を取り出し、その中の $2n$ 桁毎に情報ビットによって前 n 桁・後 n 桁を反転させる。例えば $k = 2$ の場合は、00 は前反転、前反転、01 は前反転、後反転、10 は後反転、前反転、11 は後反転、後反転させ元の場所に埋め込むことで情報を埋め込む。受信者は 2^k 通り復号し反転画素を数えて最小値を与える情報に復号する。また、最小値を与えるものが複数ある時は、同期ずれのないものを復号する。

3 実験

埋め込む情報量を $k = 2$ とし、二値画像を使って符号化、埋め込み、復号を行い、正しく情報を取り出せるのか確認する実験を行った。Fig.3 は設計したハフマン符号パターンである。このハフマン符号は適当な画像を用いて 2×2 のパターンの出現頻度を調べて作成したものである。このハフマン符号を用いて画像を符号データに変換し、符号データ中からランダムに 16 桁 ($n = 8$) を取り出し埋め込む。次に、情報が埋め込まれた符号データをコピーして、同じものを四つ作成し、パスワードに従ってそれぞれを反転、復号して、得た画像の白黒交互になっている箇所を数えて最小値を取る画像から 00/01/10/11 の判定を行う。また、同期のずれの検出を行い、前節の方法で情報を復号する。以上のことをパスワードを変えて 100 回繰り返し行った。

Table.1 は画像の規則性による判定結果である。ここで一部誤りは、2bit の内 1bit が誤って復号される事を示す。全て誤りは 2bit 全て誤って復号される事である。判定不能は $n_{ij}(i = 0, 1)(j = 0, 1)$ の中に複数の最小値がある場合である。Table.1 から分かる事は、画像の規則性だけを用いた場合、95%は正しい復号に成功したという事である。ここで、Table.1 で判定不能になった 4 つの場合について同期のずれを確認した所、全ての場合におい

て誤った復号をすれば同期がずれることが分かった。よって、同期のずれを併用すれば 99%の割合で正しい復号ができたことになる。

Fig.4 は実験に使用した二値画像、Fig.5 は、復号した画像の例であり、Fig.4 の赤枠に対応する部分を拡大して表示してある。左の画像は反転を正しく戻したものの、右の画像は正しくないものである。途中から同期がずれて画像が不連続になっているのが分かる。この画像から n_{ij} を計算すると、 $n_{00} = 21098$ 、 $n_{01} = 21104$ 、 $n_{10} = 21104$ 、 $n_{11} = 21106$ となり、正しく 00 が復号された。

0	11001	111001	1110110
10	11011	111010	11101110
1111	110101	1101000	111011110
11000	111000	1101001	111011111

Fig. 3: 割り当てたハフマン符号

Table 1: 復号結果 (規則性だけによる)

誤りなし	95
判定不能	4
一部誤り	0
全て誤り	1

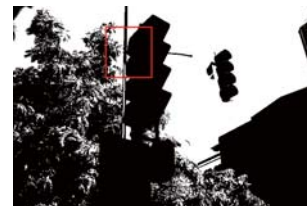


Fig. 4: 元画像 (640 × 720)



Fig. 5: 復号した画像 (拡大)

4 まとめ

ハフマン符号を用いた二値画像データへの情報秘匿の提案をした。実験により、99%情報を正しく取り出す事ができた。今後は、埋め込み対象をグレースケール画像やカラー画像に拡大し、最終的には JPEG で符号化されたデータに情報を埋め込めるようにしたい。

参考文献

- [1] 小松, 田中, 電子透かし技術ディジタルコンテンツのセキュリティ, 東京電機大学出版局 (2004)