東海大学紀要情報通信学部
vol.3,No1,2013,pp.8-14

論文

ハミング符号に基づくカラー画像の認証と復元に関する研究

福江　勇希[*1]，熱田 清明[*2]

Study on Authentication and Restoration of Color Image based on the Hamming Code

by

Yuki FUKUE[*1] and Kiyoaki ATSUTA[*2]

(Received on March 29, 2013　&　Accepted on July 25, 2013)

Abstract

Image authentication method using Hamming code for gray scale image which can recover tempered region has been proposed. However, this method sometimes misses estimating the most significant bit of tempered pixels, so the recovered image is not so good quality. In this paper, we extend this method to color images. Furthermore, we embed the most significant bit and parity bits of two color components at three color components of the color image. According to our experimental results, our new proposed method can effectively eliminate tempered region, and can recover two color components of the tempered region pixels using the most significant bit and parity bits which are embedded and estimate remaining color component by using around pixel's data.

Keywords: *Hamming Code, Image Authentication, Image Restoration*
キーワード:ハミング符号、画像認証、画像復元

1. はじめに

近年の情報技術の発展により、電子書籍などの新しい媒体があらわれ、デジタル情報のやり取りは増え続けている。それと同時に、インターネット上に公開された動画や静止画像は、不特定多数のユーザーに二次利用され、改ざんされることも多くなっている。また、改ざんされたコンテンツはさらに改ざんされ、広く出回ることで、元々のコンテンツと照らし合わせて改ざんの有無を確認することは難しくなっている。この問題を解決する方法に、電子透かしを用いた方法が提案されている[1)]。しかしながら、改ざんの有無、改ざんされた箇所までは検出できるが、元の画像に戻すことまではできない。この問題に関して、グレースケース画像に対して、ハミング符号を用いて、画像データのパリティ情報を埋め込み、改ざん領域の検出・復元が可能となる方法が提案されている[2)]。しかしこの方法にも、復元時の誤推定などの問題点がある。そこで本研究の目的はこの手法をカラー画像に拡張し、R，G、B３成分の２成分についてパリティビット、および最上位ビットを埋め込み、残りの１成分は周囲の推定済みの画素より推定することにより、より精度の高い復元を行う方法を提案することである。

*1 工学研究科情報理工学専攻 修士課程
School of Engineering, Course of Information Science and Engineering, Master's Program

*2 情報通信学部情報メディア学科 教授
School of Information and Telecommunication Engineering, Department of Information Media Technology, Professor

2. 理論

2.1 ハミング符号

ハミング符号とは自己訂正符号の１つである。ハミング符号は２ビットまでのビット誤りを検出でき、１ビットのビット誤りを修正することができる[3)]。（７，４)ハミング符号では画像の１画素のデータである８ビットの上位４ビットをデータビット(D_1, D_2, D_3, D_4)とし、下位３ビットにパリティビット(P_1, P_2, P_3)を埋め込むものとする。したがって下位から４ビット目は使用しない。その関係を Fig.1 に示す。また、パリティビットは式(1)によって定められる。ここで⊕は１ビットの排他的論理和を示す。

$$\begin{aligned} P_1 &= D_1 \oplus D_2 \oplus D_4 \\ P_2 &= D_1 \oplus D_3 \oplus D_4 \\ P_3 &= D_2 \oplus D_3 \oplus D_4 \end{aligned} \quad (1)$$

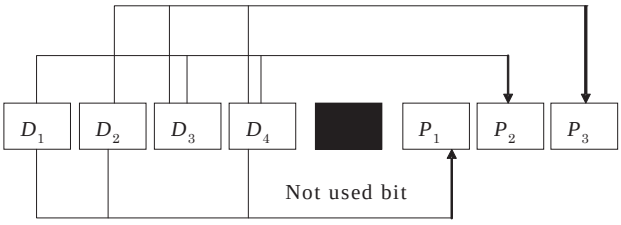


Fig.1 Date bits and parity bits in pixel data.

2.2 トーラス自己同型写像

ハミング符号によって作成した３ビットのパリティデータを元の画素の下位３ビットに保存した場合、

改ざんが行われた際に、改ざんが行われた画素のデータビットとパリティビットの両方が失われるため、認証および修正が困難になる。この問題を解決するために、上位４ビットのデータ情報と３ビットのパリティ情報を異なる座標の画素に配置する。また、改ざん領域を復元することを考えると、データ情報の画素とパリティ情報を埋め込む座標の画素が１対１の関係になければならない。そのために、トーラス自己同型写像を用いる[4]。

トーラス自己同型写像はサイズ$N\times N$の画像に対して、ある画素(x_i, y_i)を他の画素(x'_i, y'_i)に１対１に対応させることができる。kを秘密鍵となる整数とし式(2)で定義される。ここでmodは剰余を示す。

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ k & k+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \bmod N \qquad (2)$$

具体的に各(x'_i, y'_i)を求める式は式(3)となる。

$$\begin{aligned} y'_i &= (x_i + y_i) \bmod N \\ x'_i &= (k \times (x_i + y_i) + y_i) \bmod N \end{aligned} \qquad (3)$$

2.3　パリティビットの回転

2.2 節で述べたトーラス自己同型写像で用いる秘密鍵kはNの値によって制限され、k=0,1,2,3,4,..,N-1までしか存在しない。したがって、このN個の秘密鍵を試すことで、kの値を特定することができる。文献 2)では、この問題を解決するためにパリティビットの順番を回転させて暗号化している。

ビット回転は式(4)で定義される。k_2を二番目の秘密鍵とする。そして$R_1, R_2, \ldots, R_{N\times N}$のように、$N\times N$個の乱数系列を、$k_2$を初期値とする疑似乱数により作成する。次に$i$番目の画素のパリティ３ビットの順序$J$=1,2,3 を次式に従って$J'$に変換する。また、$M$はパリティビットのビット数すなわち３となる。

$$J' = (J + R_i) \bmod M \qquad (4)$$

2.4　排他的論理和による暗号化

2.3 節のパリティビットの回転による暗号化では、パリティビット３ビットが000または111である場合、ビット回転を行っても変化がないため、このビットパターンをもつ画素のみを用いて秘密鍵kの推定が可能であることがわかった。よって本手法では従来のビット回転ではなく、式(５)のように乱数系列との排他的論理和を用いてパリティビットを暗号化する手法を用いる。

$$P' = \left(P \wedge \left(R_i \bmod 2^M\right)\right) \qquad (5)$$

ここで、Pはパリティビット列を示し、^はビット毎の排他的論理和を表し、Mはパリティビットのビット数すなわち３となる。これによりパリティビットは暗号化され、秘密鍵kの推定に対する耐性が向上する。また、再度同じ乱数系列との排他的論理和を求めることで復号することができる。

3. 埋め込み・認証・復元方法

3.1　埋め込み手順

パリティビットを埋め込む手順は、ハミング符号によるパリティビットの作成、トーラス自己同型写像による埋め込み位置の算出、パリティビットの排他的論理和による暗号化および埋め込みの順に行う。具体的には。各画素に対して以下のように埋め込みを行う。

① ハミング符号を用いてパリティビットを作成する。
② 鍵k_2を初期値とする疑似乱数系列で生成した３ビットとパリティビットの排他的論理和を求める。
③ 画素の位置(x_i, y_i)と秘密鍵kを用いてトーラス自己同型写像により埋め込み画素(x'_i, y'_i)を求める。
④ 求めた埋め込み画素の下位３ビットを②で求めた３ビットに置き換える。

3.2　画像の認証

画像の認証すなわち雑音による濃度値の変化や改ざんが行われていない証明は以下のように行う。

各画素に対して、暗号鍵kを基にトーラス自己同形写像により、パリティビットを埋め込んだ画素位置を求めて、下位３ビットを取り出し、暗号鍵k_2を基に同じ疑似乱数を発生させ排他的論理和により、復号化する。画素の上位４ビットからハミング符号によりパリティビット３ビットを求める。これらの３ビットが全画素において一致する場合、改ざん等が行われていない画像であることが認証される。

3.3　画像の復元

画像の認証に失敗した場合、すなわちデータビットとパリティビットの整合性が取れない画素が存在する場合、その原因には雑音がある環境と改ざんの２つが考えられる。雑音がある環境とはデータがインターネット上などで伝送される時、雑音の影響を受けてデータビットにエラーが発生することである。改ざんとは故意に画像の改ざんが行われ、特定の領域のデータが書き換えられ、失われてしまうことである。雑音によるものか改ざんによるものかの判定は、雑音による不整合な画素の分布で簡単に判定することができる。雑音による場合はランダムで、画像全体にまばらに分布するが、改ざんによる場合は、ある部分にまとまって発生する。

3.3.1　雑音がある場合

雑音がある場合についてはハミング符号によって復元が可能である。

Fig.2 は、データビット（D_1～D_4）とパリティビット（P_1～P_3）の関係を示したものである。この３つのサークル内での１ビットの数は常に偶数になるようにパリティビットが定えられている。Fig.3 にはビットエラーが１つある。サークル A に属する値 0,1,1,1

の 1 のビット数は 3 となりサークル A に属する値にエラーが存在することがわかる。同様にサークル B、C の値も計算をおこなうと、1 のビット数が奇数になるのはサークル A とサークル B なのでこの 2 つにのみ属する D_1 がエラーであることがわかり値を反転することによって、復元が可能となる。

もしパリティビットにエラーが生じた場合は、そのパリティビットを含むサークル 1 つだけが奇数になる。この場合、画質的には修正の必要はないが、再度の配信を行う場合などを考え、誤ったパリティビットを修正する。

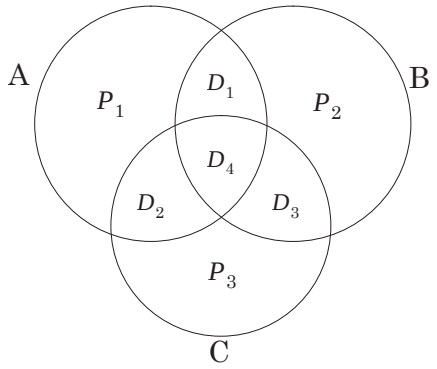


Fig. 2 Relation between data bits and parity bits.

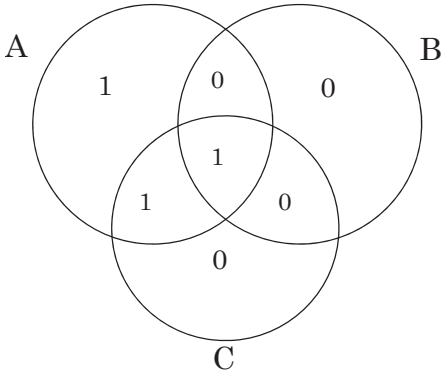


Fig. 3 An error occurs in D_1

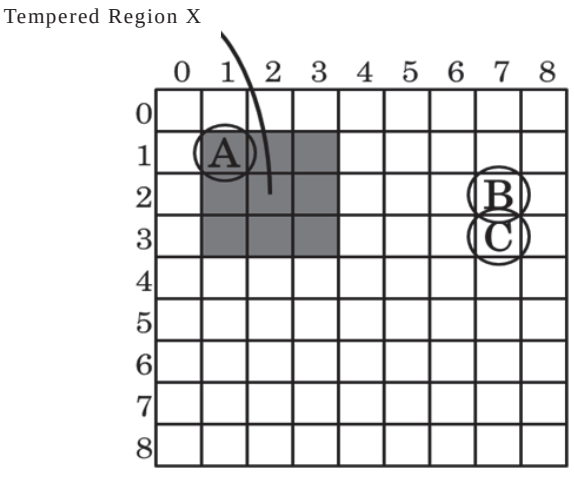


Fig.4 An example for tampered regions

3.3.2 改ざんが行われた画像の場合

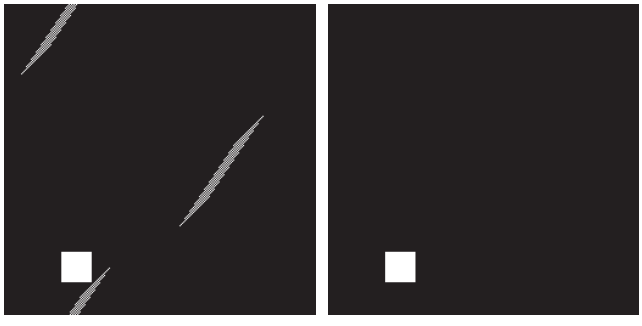
改ざんが行われた画像の場合、改ざんは一定の領域に行われ、集中してエラーが生じることになる。この場合、最初に改ざんされた領域を特定する必要がある。Fig.3 におけるグレーの領域 X を改ざんされた領域とし、画素 A がこの領域に属すと仮定する。このとき、A のパリティビットは B に埋め込まれており、C のパリティビットは A に埋め込まれる。

画素 A は改ざんされているので、そのデータビットから算出されたパリティビットは画素 B に埋め込まれたものとは異なる。したがって、このとき、A または B のどちらかが改ざんされたと判断できる。パリティビットを画素 A に埋め込んでいる画素 C についても、同様のことが言える。すなわち、C または A にどちらかが改ざんされたと判定される。このようにして A,B,C が改ざん候補となる。

このプロセスをすべての画素について行い、すべての改ざん候補画素を検出する。Fig.4 では領域 X に改ざんが行われているので、この領域のすべての画素が改ざん候補となっている。その他の改ざん候補画素は、まとまった領域を作らず、分散して存在する。分散した領域を除去するために、改ざん候補の画素に対して形態学的画像処理すなわち収縮処理と膨張処理を行う。

3.2.4 形態学的画像処理による改ざん領域の特定

改ざん領域の特定のために形態学的画像処理を用いる。ここでの形態学的画像処理とは 2 値画像に対する収縮処理と膨張処理のことである。収縮処理の後に膨張処理を行うことにより、孤立点や突起を取り除くことができる [5]。各処理を 1 回ずつ行うと、2 画素分の画素領域は取り除くことができ、それより大きい画素領域は残る。Fig.5(a)に改ざん画素の候補を示し、図 Fig.5(b)に収縮・膨張処理により求めた改ざん領域を示す。



(a) Candidate of tempered region　(b) Estimated tempered region

Fig.5 Estimation of tampered region

3.2.5 ハミング符号による復元

3.2.4 項の手続きにより、改ざんされた画素が特定できる。次の手続きはこの改ざんされた画素を復元する。

Table 1 にデータビットとそれに対応するパリティビットの対応表を示す。この Table 1 を見ると、デー

タビットの最上位ビットが０または１によって２つのグループに分ける事ができる。そして最上位ビットが０、または１である事が判明すると、パリティビットからデータビットを推定することができることが表から読み取れる。

データビットの最上位ビットを判定する方法は以下の通りである。

画像の場合、もし最上位ビットが１であれば、その輝度値が１２８以上であることになり、最上位ビットが０であれば、輝度値が１２８よりも低いことになる。したがって、周囲画素から画素の大まかな輝度値を推定できれば、その値が１２８以上であれば最上位ビットを１とし、１２８未満であれば０として、データビットの復元が行える。

Table 1 The Hamming code word.

Data bits		Parity bits
MSB	Remain 3bits	
0	000	000
0	001	111
0	010	011
0	011	100
0	100	101
0	101	010
0	110	110
0	111	001
1	000	110
1	001	001
1	010	101
1	011	010
1	100	011
1	101	100
1	110	000
1	111	111

3.2.6　非線形推定

改ざん領域の左上から非線形推定 [6]を行って対応画素の最上位ビットを求める。Fig.4 において画素 A が改ざんされている。この場合、改ざんされていない左、左上、上の画素から非線形推定を用いて推定する。この非線形推定は次式で定義される。すなわち、画素値 d をその近傍の 3 つの画素値 a,b,c で推定する。Fig. 6 に画素位置と a,b,c,d の対応関係を示す。

$$d=\begin{cases}\min(a,b) & if\ c\geq\max(a,b).\\ \max(a,b) & if\ c\leq\min(a,b).\\ a+b-c & \text{otherwise.}\end{cases} \quad (6)$$

a	b
c	d

Fig.6 Arrangement of pixels for equation (6).

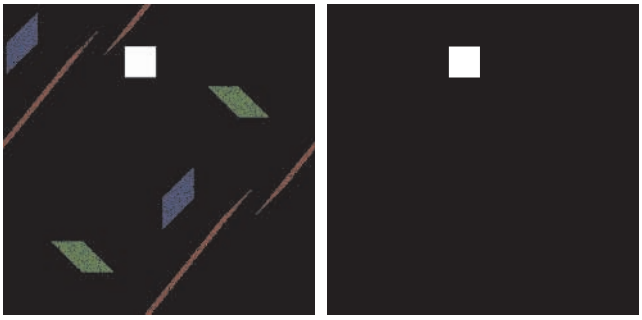
画素Ａの右隣の画素は画素Ａの推定値と左上、上の画素から推定することができる。これを順次繰り返して改ざん領域内の画素の値を推定していく。しかしながら、改ざん領域内にエッジが存在した場合、非線形推定では、最上位ビットの推定に失敗することがある。

3.3　カラー画像の復元方法

この方法を各成分に単純に適用することによって、カラー画像にも拡張可能であるが、最上位ビットの推定を誤ると復元画像の画質が著しく劣化するという欠点がそのまま残されてしまう。そこで、カラー画像が３成分から構成されることを利用して、改善方法を以下に提案する。

3.3.1　カラー画像の改ざん領域特定

カラー画像の場合には、Fig.7 に示すようにそれぞれの成分でトーラス自己同形写像に異なる鍵 k_R,k_G,k_B を用いることで、ＲＧＢ各成分の改ざん領域候補座標を照合し、簡単に改ざん領域候補が重複している座標を改ざん領域として特定することが可能であり、より信頼性の高い改ざん領域の検出ができる。



(a) Candidate of tempered region.　(b) Estimated tempered region.

Fig.7 Determine the tempered reagion.

3.3.2　カラー画像に対する改ざん領域の復元

カラー画像はＲＧＢの３成分で構成されている。ＲＧＢの３成分の下位３bit に復元用のデータを保存することが出来る。例えばＲ成分の下位２ビットにＧ成分とＢ成分の最上位ビットを保存し、Ｇ成分の下位３ビットにＧ成分のパリティビットの乱数との排他的論理和により暗号化したデータを保存し、Ｂ成分の下位３ビットには、Ｂ成分のパリティビットの暗号化したデータを保存する。このように下位３ビットに２成分の最上位ビットとパリティビットの情報を保存することにより、Ｇ成分とＢ成分の２つの成分を完全に復元できる。すなわち上位４ビット

のデータが正確に復元できる。同様にして、R成分とG成分、R成分とB成分が完全に復元できる画素を設定することができる。RG、GB、RBを完全に復元できる画素をFig.8のように配置する。行ごとにRG成分を復元できる画素、GB成分を復元できる画素、BR成分を復元できる画素の順にずらしながら繰り返し配置されている。このように画素を配置することによって、ある画素において復元できない成分は、その画素の左、左上、上、右、右下、下の6画素では復元されているので、それらから推定して復元することができる。例えばFig.8の中央画素の場合、R成分とG成分は復元でき、B成分が復元できないが、左、左上、上、右、右下、下の6画素はB成分が復元されている。したがって、これらの値を用いて推定することができる。したがって、3.2.6項で述べた非線形推定による最上位ビットの推定を行わないので、エッジ部分の誤推定がなくなり、エッジ領域の改ざんも復元できる。

GB	RB	RG
RB	RG	GB
RG	GB	RB

Fig.8 The pattern of embedded two color components.

3.3.3 提案手法におけるカラー画像復元の流れ

この手法を用いた画像に改ざんが行われた場合に対するカラー画像の復元の復元は以下のように行う。

① Fig.8のパターンとR，G，B成分用のトーラス自己同型写像の鍵 k_R,k_G,k_B を用いて、暗号化されたパリティビットと最上ビットを埋め込んだ画素を求める
② 排他的論理和用の k_2 によって、生成された疑似乱数系列を用いて、パリティビットを排他的論理和により復号化する。
③ データビットとパリティビットが対応しない画素をチェックし、2成分で対応しない画素を改ざんされた領域とする。
④ 改ざん領域の画素に対して、パリティビットと埋め込まれていた最上位ビットにより、2成分のデータビットを求める。
⑤ 改ざん領域の全画素について復元できる2成分を求め、残りの1成分を周囲6画素から推定する。ここでは上下左右の4画素の成分の単純平均を用いて推定する。

4. 実験

4.1 実験画像

実験で扱う画像は256×256[pixels]の各色256階調のカラー画像Parrotsとする。Fig.9に実験に用いた画像を示す。

Fig.9 An original image(Parrots).

4.2 従来手法によるパリティビットの埋め込み

従来手法をカラー画像に行う場合、R、G、B成分それぞれでに鍵 k_R,k_G,k_B を用いてパリティビットを埋め込む。Fig.10にパリティビットを埋め込んだ画像を示す。

Fig.10 The authenticated image by original method.

4.3 画像復元

4.3.1 改ざん画像

復元の実験を行うため、Parrots画像に改ざんを行った。Fig.11に改ざん画像を示す。今回、改ざんは指定された領域をネガポジ反転することで作成した。

Fig.11 The tampered image.

4.3.2 改ざん領域の特定

カラー画像のRGBの成分毎にハミング符号の一致しない領域を求め、2成分以上が一致しない領域を改ざん領域とした。Fig.12に改ざん領域の特定結果を示す。

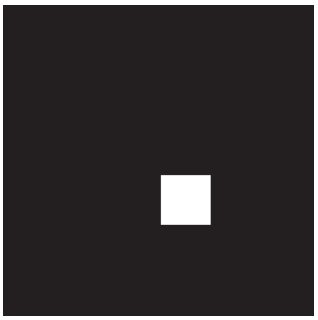


Fig.12　Estimated tampered region.

4.3.3　従来手法による復元

従来の手法をそのまま３成分に適用して、改ざん領域を非線形予測とパリティビットによって復元する。復元した画像を Fig.13 に示す。非線形推定は左上から行うので、改ざん領域において左上と右下の濃淡値の差が激しいエッジでは値の推定がうまくできない。そのため、赤の成分において、エッジ部分で最上位ビットの推定を誤り、正しく復元ができていないため、本来黄色になる部分が緑色のままになってしまったことがわかる。

Fig.13 The image corrected by original method.

4.4　提案手法

２成分のパリティビットとその最上位ビットを埋め込む提案手法による埋め込みと復元を行った。埋め込み画像を Fig.14 に、提案手法により復元した画像を Fig.15 に示す。この提案手法によって、エッジ部分での誤推定がなくなり、黄色の領域が正しく復元されていることがわかる。

Fig.14 The authenticated image by our proposed method.

Fig.15 The image corrected by our proposed method.

4.5 PSNR による画質の評価

4.5.1　PSNR（Peak　Single to Noise Ratio）

PSNR とは客観的な画像評価の指標の一つである。画像評価の基準値であり、値が大きくなればなるほど原画像に近いことを表す。まったく同じ画像であれば∞(無限大)となる。PSNR は下記の式で表される。

$$PSNR = 10\log\frac{(\text{階調数}-1)^2}{\frac{1}{N*N}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1}(src(y,x)-dst(y,x))^2} \quad (7)$$

単位はデシベル[dB]となり、ここで *x,y* はピクセルの位置、*src* は原画像　*dst* は評価対象の画像、*N* は画像の大きさを表す。

4.5.2　PSNR による画像評価

原画像に対する従来手法による埋め込み画像、改ざん画像、復元画像の PSNR を Table 2 に示す。R 成分の最上位ビットの誤推定により、R 成分の PSNR がほとんど改善されていないことがわかる。

Table 2 PSNR of images for original method.

	The authenticated image	The tampered image	The image corrected by original method
R	37.83[dB]	18.71[dB]	20.80[dB]
G	37.92[dB]	21.31[dB]	33.50[dB]
B	37.82[dB]	18.04[dB]	30.00[dB]

原画像に対する提案手法による埋め込み画像、改ざん画像、復元画像の PSNR を Table 3 に示す。埋め込み画像の画質もパリティビット３ビット２成分と２つの最上位ビットと１ビット少ない埋め込みとなったため、１[dB]ほど向上していることがわかる。また、復元画像も１成分は周囲からの推定であるが、２成分は正確に復元できるため、大幅に PSNR 向上していることがわかる。

Table 3 PSNR of images for our proposed method.

	The authenticated image	The tampered image	The image corrected by proposed method
R	39.01[dB]	18.72[dB]	38.63[dB]
G	39.12[dB]	21.32[dB]	38.80[dB]
B	39.01[dB]	18.04[dB]	38.52[dB]

4.6 未決定色成分の推定方法の比較

以上の評価は上下左右の４画素の単純平均により未決定の色成分の値を推定しているが、隣接画素中の６画素を用いることもできる。また前述の３画素からの非線形推定を用いることもできる。そこで、これらの推定方法による復元を行い、画質の違いを Table 4 に示す。３画素しか用いない非線形推定は急激な値の変化に対応できないこともあり、一番画質が低い。６画素の単純平均は、４画素の単純平均に比較して若干の低下がみられる。これらの画質や計算量の点からから、最近傍の４画素を用いる方法で十分であるといえる。

Table 4 Compare of image quality by estimation method.

	Non-Liner estimation	Estimation form 4 pixels	Estimation form 6 pixels
R	37.98[dB]	38.63[dB]	38.49[dB]
G	38.52[dB]	38.80[dB]	38.69[dB]
B	38.18[dB]	38.52[dB]	38.42[dB]

5. 結論

本研究では、白黒画像に対する方法をカラー画像に拡張し、復元精度を向上させることを目的とした。

従来のビット回転では自己同形写像の鍵 k の推定が容易であるという問題点は、排他的論理和を用いることで解消され、秘密鍵の推定に対する耐性が向上した。

目的であるカラー画像における復元精度の向上については、３成分の内２成分の最上位ビットとパリティビットの情報を３成分の下位ビットに埋め込むことにより、２成分を非線形推定に頼らず正確に復元し、２成分の組み合わせ、配置を工夫することにより未決定の成分を上下左右の４画素から線形推定する方法を提案して、その有効性を実験的に示した。また、未決定の色成分の推定には上下左右の４画素の単純平均を用いることで十分な効果が得られることを実験的に示した。

今後の展望としては非可逆圧縮等の各種画像処理に対する耐性の検証が今後の課題として残されている。

謝辞

本研究は JSPS 科研費 23500130 の助成を受けたものである。

参考文献

1) 須藤正之、三井靖博、“改ざん検出可能な電子透かし”、沖電気研究開発 2000 年 7 月第 183 号 Vol.67No.2,2000.
2) Chi-Shiang Chana, and Chin-Chen Chang “An efficient image authentication method based on Hamming code” Pattern Recognition 40(2007), 681-690, 2008/01/29.
3) R.W. Hamming,”Error detecting and error correcting codes”,Bell Syst. Tech. J., 26 (2) (1950), pp. 147–160.
4) G. Voyatzis, I. Pitas, “Chaotic mixing of digital images and applications to watermarking”, Proceedings of European Conference on Multimedia Applications, vol. 2, 1996, pp. 687–695.
5) 田村秀行,“コンピュータ画像処理”オーム社出版局.
6) Marcelo J. Weinberger and Gadiel Seroussi, Guillermo Sapiro “The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS”, http://www.hpl.hp.com/loco/HPL-98-193R1.pdf.