

情報ハイディング画像の高画質化 A Highly Imperceptible Information Hiding System

佐々木隆幸

弘前大学大学院理工学研究科

長瀬智行

弘前大学大学院理工学研究科

概要：この発表は、1枚の画像の中に1枚の情報画像を埋め込める一重の情報ハイディング画像において、高画質化を追求した制作と再生の方法を提案するものである。

この提案の要点は3つある。1つは直交関数系として擬似乱数系列を用いたこと。2つ目は埋め込む情報画像の画素値を限られた範囲の整数値に量子化したこと、そして3つ目はその量子化したビットプレーンを移し替えたことである。

ここでは、これらの要点をイラストや画像で例示しながら、一重の情報ハイディング画像の制作と再生について発表する。

This presentation proposes a method of constructing a watermarking image and recovering original image where the difference between these images is highly imperceptible. There are three main points of this proposal. One is the use of pseudorandom number sequences as orthogonal function systems. The second is that the pixel values of the information image to be embedded have been quantized to a limited range of integer values, and the third is that the correlation coefficient of the information hiding image can be improved by implementing a bit-plane transposition.

キーワード：情報ハイディング画像、擬似乱数、直交関数系、量子化、ビットプレーン移し替え

1. はじめに

デジタル技術の進歩により、さまざまなイラストや写真画像などを容易に伝達[1]できるようになってきた。また、Wi-Fi環境が整備されつつあり、誰でもが通信ネットに簡単にアクセスできるようになってきた。その反面、ネット上に流通する画像を困難なく盗聴あるいは改ざんすることができるようになり、社会的トラブルを生んでいる。

このような背景に対処する方法がいろいろと提案されている。その一分野に情報ハイディング技術がある。情報ハイディング技術で採用されている直交関数系の多くは、三角関数やハール関数などの超越関数に属する関数[2][3][4]が大半で、代数関数に属する直交多項式で構成される直交関数系が採用されるのは、あまり見当たらない[5][6][7][8]。さらに擬似乱数を採用した情報ハイディング技術に関しては、ほぼ皆無である。そこで、擬似乱数系列を採用した文献[9]に改良を加え、元の画像に近い高画質な画像を得ることができたので、それについて発表する。

2. 擬似乱数系列の正規直交化

情報ハイディング画像の制作と再生に採用した直交関数系は擬似乱数系列を加工したものである。その加工方法を、文献[9]を参考に述べる。

なお、以降で用いる用語を整理しておく。秘匿に伝達する画像を情報画像、その画像を直交関数系で変換し量子化した画像をホログラム、ホログラムを埋め込む土台となる画像をカギ画像、ホログラムをカギ画像に埋め込んだ画像を情報ハイディング画像、情報ハイディング画像から再生

した画像を再生画像と呼ぶことにする。

また、画像の横×縦の画素数を $N \times N$ とする。記号 Z_{ij} は、画像画面の最左下位置を1行1列とする i 行 j 列における画素値を表すものとする。

2.1 直交化

- ①個数が N^2 個の擬似乱数からなる系列を一つ用意する。乱数の値が $0 \sim 1$ の小数であるときは、その値を 0.5 だけ下げ、正と負の数が混在する擬似乱数系列につくり変える。
- ②擬似乱数系列を区切り N 行 $\times N$ 列の行列に配置替える。ここで、各行の横並びの擬似乱数を、点 $j = 1, 2, \dots, N$ において定義された関数値とみなす。すると関数は N 個できる。第 i 行($i = 1, 2, \dots, N$)における関数を $\psi_i(j)$ と書き表す。
- ③次に、第1行の関数 $\psi_1(j)$ の擬似乱数をすべて値 1 に置き換える。
- ④その関数 $\psi_1(j)$ を基に、第2行以降の関数 $\psi_i(j)$ を順に直交化する。その方法はシュミットの直交化手続きを用いて行う。直交化した関数 $\phi_i(j)$ は次式のとおおり。

$$\begin{cases} \phi_1(j) = \psi_1(j) \\ \phi_i(j) = \psi_i(j) - \sum_{k=1}^{i-1} \frac{(\phi_k(j), \psi_i(j))}{(\phi_k(j), \phi_k(j))} \phi_k(j) \end{cases} \quad (1)$$

ただし、 $(\phi_i(j), \phi_k(j)) = \sum_{j=1}^N \phi_i(j) \cdot \phi_k(j)$ とする。

2.2 正規化

直交関数系 $\{\phi_i(j)\}$ の各関数 $\phi_i(j)$ を次式で正規化する。正規化した関数を $\varphi_i(j)$ とする。

$$\varphi_i(j) = \frac{\phi_i(j)}{\sqrt{(\phi_i(j), \phi_i(j))}} \quad (2)$$

$\varphi_i(j)$ は次の関係を満たす。

$$(\varphi_i(j), \varphi_k(j)) = \delta_{ik} \quad (\delta_{ik} \text{はクロネッカーのデルタ}) \quad (3)$$

以上の処理で、情報ハイディング画像に採用できる正規直交関数系 $\{\varphi_i(j)\}$ がつけられる。

3. 電子透かし画像の制作と再生の方法

3.1 制作方法

情報ハイディング画像を制作する過程を図1に示す。

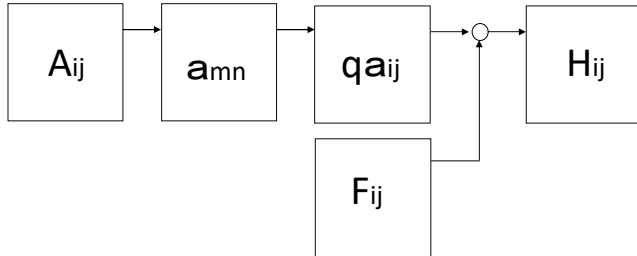


図1 情報ハイディング画像の制作過程

①情報画像を1枚用意する。それを A_{ij} とする。さらに、カギ画像を1枚用意し、それを F_{ij} とする。

②式(4)を用いて、 A_{ij} を正規直交関数系で展開したときの展開係数 a_{mn} を算出する。

$$a_{mn} = \sum_{j=1}^N (\sum_{i=1}^N A_{ij} \varphi_m(i)) \varphi_n(j) \quad (4)$$

カラー画像のときは、赤色、緑色、青色の3つの展開係数を算出する。

③展開係数 a_{mn} を量子化する。量子化する理由を述べる。展開係数 a_{mn} の値は広範囲であるのに対して、BMP画像の画素値は限られた正の整数値0~255しかとれない。したがって、展開係数を画像として記録するためには整数値に量子化する必要がある。展開係数 a_{mn} を D_a ビットの画素空間 $\{0, 1, \dots, 2^{D_a} - 1\}$ の画素値に量子化する。量子化する画素値をその範囲に制限した理由は、ホログラムをカギ画像に埋め込むとき、定数倍することなく、その画素値のまま埋め込めるようにしたためである。なお、以降では、量子化された展開係数を量子化展開係数と呼ぶことにする。

展開係数を量子化する方法を述べる。展開係数の範囲を下に示した(i), (ii), (iii)の3つの区間に分ける。値 u, v は a_{mn} の範囲に基づいて定める。目安としては $u \approx \log_{10}(255^2 N)$, $v \approx \log_{10}(1/\sqrt{N})$ である。

展開係数 a_{mn} と量子化展開係数 qa_{mn} の関係をそれぞれの区間ごとに(5), (6), (7)とする。それをグラフに表したのが図2である。ただし、図2は量子化展開係数の画素値が8個の場合で例示する。

$$(i) -10^u < a_{mn} \leq -10^{-v} \text{のとき} \quad (5)$$

$$qa_{mn} = \frac{2^{D_a-1}-1}{u+v} (\log_{10}(-a_{mn}) + v) + 1$$

$$(ii) -10^{-v} < a_{mn} < 10^{-v} \text{のとき} \quad (6)$$

$$qa_{mn} = 0$$

$$(iii) 10^{-v} \leq a_{mn} \leq 10^u \text{のとき} \quad (7)$$

$$qa_{mn} = \frac{2^{D_a-1}-1}{u+v} (\log_{10}(+a_{mn}) + v) + 2^{D_a-1}$$

④ビットプレーンを移し替える。理由は情報ハイディング画像の画質を高めるためである。LSBに相当するビットプレーンをビットプレーン0、MSBに相当するビットプレーンをビットプレーン7とする。ビットプレーン $(D_a - 1)$ をビットプレーン0に置き換え、ビットプレーン0はその1つ上のビットプレーン1へ、ビットプレーン1はビットプレーン2へと順に1つ上へ移し替え、そしてビットプレーン $(D_a - 2)$ をビットプレーン $(D_a - 1)$ に移し替える。これでビットプレーンの移し替えは完了する。

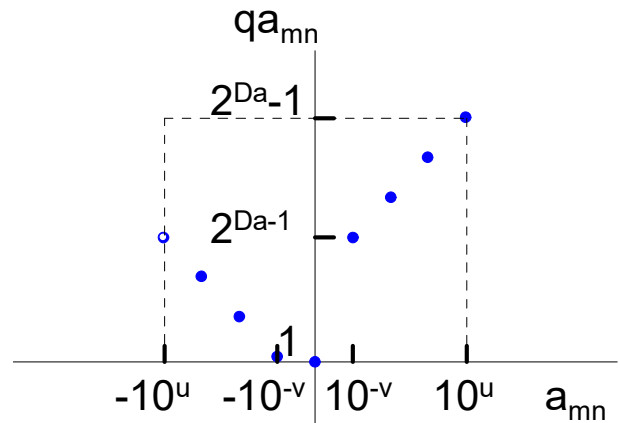


図2 a_{mn} に関する式(5),(6),(7)のグラフ

⑤移し替えたビットプレーンを合成したのがホログラム G_{ij} である。

⑥最後に、このホログラムをカギ画像 F_{ij} に式(8)にしたがって埋め込む。情報ハイディング画像 H_{ij} の制作完了である。

$$H_{ij} = G_{ij} + \frac{256-2^{D_a}}{255} F_{ij} \quad (8)$$

3.2 再生方法

情報ハイディング画像を再生する過程を図3に示す。再生方法は原理的には制作方法の逆過程である。

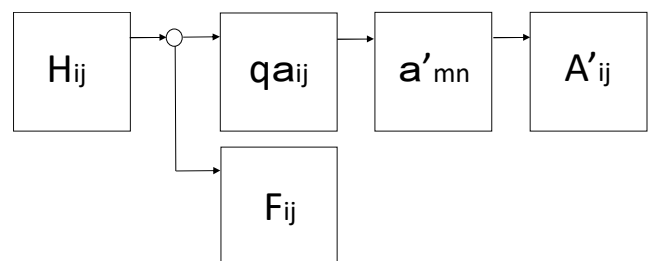


図3 情報ハイディング画像の再生過程

①情報ハイディング画像からカギ画像を差し引く。ホログラムが残る。

②ビットプレーンを元の位置に戻すと、量子化展開係数 qa_{ij} を算出できる。

③式(5),(6),(7)の逆演算を行う。その結果を a'_{mn} とする。 a'_{mn} は a_{mn} そのものではなく、量子化による誤差が含まれたものである。

⑤最後に、 a'_{mn} を式(9)に適用し、再生画像 A'_{ij} を再生する。

$$A'_{ij} = \sum_{n=1}^N (\sum_{m=1}^N a'_{mn} \varphi_m(i)) \varphi_n(j) \quad (9)$$

4. 制作と再生の実験

4.1 擬似乱数の乱雑さと正規直交化

擬似乱数系列は、ソフトウェア「Mathematica」(Wolfram Research 社) が発生する擬似乱数系列である。この擬似乱数系列から正規直交関数系を構築する。それを、文献[9]を参考に示す。

その結果が図 4 である。最前列が直交関数 $\varphi_1(j)$ であり、右上奥へ順に進み、最終列が $\varphi_N(j)$ ($j = 1, 2, \dots, N$)である。

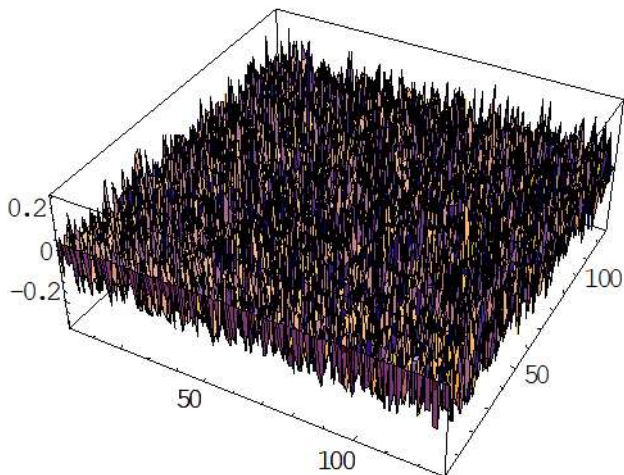


図 4 擬似乱数系列を加工した正規直交関数系

4.2 情報ハイディング画像の制作と再生

画素値の量子化における u, v, D_a をそれぞれ5, -1, 6とする。

(1) 制作実験

図 1 の制作過程に制作途中の画像を挿入したのが図 5 である。

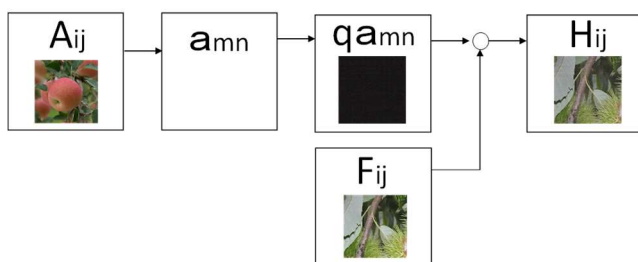


図 5 情報ハイディング画像の制作過程(画像挿入)

(1) 再生実験

図 3 の再生過程に再生途中の画像を挿入したのが図 6 である。

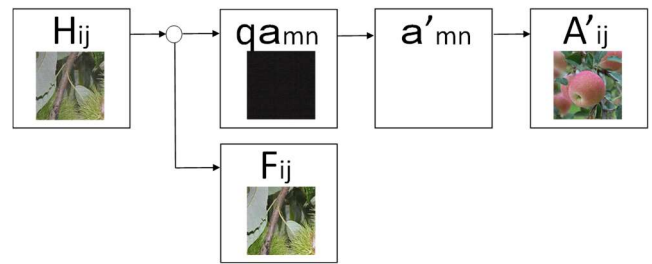


図 6 情報ハイディング画像の再生過程(画像挿入)

(2) 相関係数の測定

図 7 のカギ画像 F_{ij} と制作した図 8 の情報ハイディング画像 H_{ij} との相関係数を測定する。その結果を表 1 に示す。



図 7 カギ画像 F_{ij} 図 8 情報ハイディング画像 H_{ij}

表 1 情報ハイディング画像とカギ画像の相関係数

	赤色	緑色	青色
相関係数	0.991	0.992	0.993

また、図 9 の情報画像 A_{ij} と図 10 の再生画像 A'_{ij} の相関係数を測定する。その結果を表 2 に示す。



図 9 情報画像 A_{ij} 図 10 再生画像 A'_{ij}

表 2 情報画像と再生画像の相関係数

	赤色	緑色	青色
相関係数	0.991	0.991	0.991

5. 高画質化の要点

この情報ハイディング画像の高画質化には 3 つの要点がある。採用した直交関数系が擬似乱数系列を加工したものであること、情報画像の画素値を限られた範囲に量子化したこと、ホログラムがビットプレーンを移し替えたものであることの 3 つである。これらの要点について述べる。

5.1 擬似乱数系列

直交関数系が擬似乱数系列を加工によるものであるため、情報画像から得た量子化展開係数の分布が局所的ではなく、ビットプレーン全体に乱雑に散在する傾向にある。このことが情報ハイディング画像の画質向上に役立つ。

5.2 量子化

情報画像を直交関数系で展開した直後の展開係数の値は正負の広範囲に及ぶ。これを画像として記録できる正の整数値に変換しなければならない。その変換が量子化である。その量子化を自由自在に制御できることが情報ハイディング画像の高画質化に役立つ。

5.3 ビットプレーン移し替え

ビットプレーンの移し替えがある場合とない場合を比較してみる。ビットプレーンの移し替えがない場合の情報ハイディング画像を図 11 に示す。これと図 7 のカギ画像との相関係数を表 3 に示す。

一方、ビットプレーンの移し替えがある場合の情報ハイディング画像はすでに示したように図 8 で、その画像と図 7 との相関係数は表 1 のとおりである。



図 11 情報ハイディング画像(移し替えなし)

表 3 図 7 のカギ画像と図 11 の再生画像の相関係数

	赤色	緑色	青色
相関係数	0.930	0.930	0.939

上の比較から、ビットプレーンを移し替えた情報ハイディング画像が移し替えのない画像よりも優れていることがわかる。その理由を次に述べる。

表 4 はビットプレーンの移し替えがない場合とある場合をビットプレーン 0 から 7 までを併記したもの（ただし青色の場合のみ）である。移し替え有り列のビットプレーン 0 は、移し替え無し列のビットプレーン 5 である。そして、移し替え有り列のビットプレーン 1~5 は移し替え無し列のビットプレーン 0~4 が 1 つずつ上にシフトしたものである。このようなビットプレーンの移し替えによって、情報ハイディング画像の相関係数が向上し、高画質になる。

表 4 ビットプレーン移し替えの有・無

	移し替え無し	移し替え有り
bit plane 7		
bit plane 6		

bit plane 5		
bit plane 4		
bit plane 3		
bit plane 2		
bit plane 1		
bit plane 0		

6. おわりに

この情報ハイディング画像が、伝達において秘匿性・安全性に優れている点を 1 点述べる。たとえば、情報ハイディング画像が盗聴者に不正に盗聴されたとしても、盗聴者は秘匿な情報画像を再生すること困難である。なぜならば、情報ハイディング画像が擬似乱数系列を加工した直交関数系でつくられたものであるため、関数に定型性がなく、盗聴者はその直交関数系を知ることが困難であるからである。この情報ハイディング画像が多いに役立つことを期待する。

参考文献

- [1] U. Mustafa, U. Guzin, V. V. Nabiyev, Medical image security and EPR hiding using Shamir's secret sharing scheme, Journal of Systems and Software, Vol.84, No.3, pp.341-353, 2011.
- [2] 大西淳二, 小野東: "電子透かしを用いた印刷の改ざん検知方法の検討", 電子情報通信学会論文誌 D, Vol. j90-D, No.6, pp.1484-1494, 2007
- [3] 木野将人, 和田成夫: "ビットデータを埋込み可能なウェブレット画像透かし法", 電子情報通信学会論文誌 A, Vol. J86-A, No.2, pp.160-167, 2003
- [4] 栗林稔, 田中初一: "DCT 係数間の加法特性に基づく電子透かし", 電子情報通信学会論文誌 A, Vol. J85-A No.3 pp.322-333, 2002
- [5] 佐々木隆幸, 2 枚の電子透かし情報画像を埋め込んだ電子透かしの制作と復元, 特許庁, 特願 2016-217619, 2016
- [6] S. Alyammahi, F. Taher, H. Al-Ahmad and T. McEloughlin, "A New Multiple Watermarking Scheme for Copyright Protection and Image

- Authentication,” 59th IEEE Inter. Midwest Symposium on Circuits and Systems, 2016
- [7] 佐々木隆幸,川守田聡,直交関数系でつくる電子透かし, 職業能力開発報文誌, Vol.30, No.1, pp.1-12, 2018
- [8] Takayuki Sasaki and Tomoyuki Nagase, “Constructing Digital Watermark Based on Orthogonal Functions,” 5th IEEE Inter. Conference on Cyber Security and Cloud Computing (CSCloud), pp.140-143, 2018.
- [9] 佐々木隆幸,長瀬智行,直交多項式でつくる二重電子透かし, 情報処理学会研究報告, Vol.2018-DCC-19, No.5, 2018