誤りパターン埋込み型ステガノグラフィに関する一考察

A study on steganography based on embedding error patterns

研究者 : 索手　一平　　指導教員 : 福岡　久雄
Researcher : Ippei NAWATE　　Supervisor : Hisao FUKUOKA

松江工業高等専門学校情報工学科
Department of Information Engineering, Matsue College of Technology

Abstract

本論文は，テキスト情報をグレースケール画像に埋め込むステガノグラフィ技術を対象とする．データの埋め込み手 法の一つである誤りパターン埋め込み法は，テキスト情報を冗長かつハミング重みの小さいビット列（これを誤りパターンという）に変換し，画像中の LSB 平面との排他的論理和で LSB 平面を置き換える．本研究では，誤りパターン埋め込み法における画質劣化等について考察する．また，Shalkwijk の数え上げ符号を用いた誤りパターンの動的生成方法を提案する．

Keywords

ステガノグラフィ，セキュリティ，誤りパターン

目次

1　はじめに

近年，ネットワーク通信の増加に伴い，安全に通信を行うための技術の重要性が高まってきている．特に秘密情報の通信において重要なものとなる．安全に通信を行う手段として主なものに送信内容の暗号化がある．暗号化により，送信内容を第三者に読み取られない状態とすることで，第三者による盗聴，なりすましといった妨害行為を防ぐことができる．しかし，通信行為そのものは第三者に認知されていることから，通信経路の遮断などといった妨害行為をうける可能性を排除できない．この点に対し，ステガノグラフィでは送信内容を別のデータに埋込むことで，第三者から秘密情報の存在そのものを隠蔽する．つまり，通信行為そのものを隠蔽し，より安全な通信を実現することができる．

本論文ではステガノグラフィ技術の中で最も埋め込みに利用される画像，特にグレイスケール画像に対し，埋め込みデータをテキスト情報とし，誤りパターン埋め込み法における埋め込み率と誤り率，画質劣化とのトレードオフ関係を実験的に明らかにする．なお，画質劣化の指標には SSIM[1] を用いる．また，誤りパターンテーブル法の肥大化問題 [2] に対する解決策として Shalkwijk の数え上げ符号 [3] を用いた誤りパターンの動的生成手法を提案する．

2　ステガノグラフィとは

ステガノグラフィとは秘密情報を別の媒体に埋め込む技術，研究の総称であり，情報ハイディング技術の一つである．

ステガノグラフィ以外の情報ハイディング技術として電子透かしがあるが，これはデータの著作権や所有権を保護するために，そのデータの中に，ある証拠データを埋め込む技術のことである．電子透かしでは埋め込まれる証拠データの頑健さ，つまり他者に除去されないことが重要である．一般的に埋め込まれる情報は少量であり，外部に見えることが望ましいため証拠データそのものには大きな価値はない．

一方でステガノグラフィは秘密情報を隠蔽するために，秘密情報を別のデータ（これをカバーデータという）内に埋め込む．ステガノグラフィでは電子透かしとは逆に埋め込まれる秘密情報そのものに大きな価値があり，カバーデータに価値はない．また隠蔽性の高さ，つまり他者に秘密情報が埋め込まれているという事実そのものを検知されないことが重要となる．一般的に埋め込まれる情報は少量とは限らず，多くのデータの埋め込みは埋め込み対象となる別のデータを劣化させてしまうため，ステガノグラフィでは以下の 2 点を同時に満たすことが強く望まれる．

1. できるだけ多くの情報の埋め込みが可能である
2. 埋め込みが主観的・客観的に認知されない

また，秘密情報を埋め込まれたカバーデータをステゴデータと呼ぶ．

カバーデータには主に音楽，画像などのメディアデータが用いられる．ほとんどのメディアデータはいくらかの冗長性を有しており，これらを利用した埋め込みが主流である．本論文で取り上げる画像では，画像中の各ピクセルにおける LSB などの下位ビットが冗長性ということになる．これらのビットはそれぞれが持つ情報量が少なく，変化させたとしても視覚的変化がほとんど発生しない．特に LSB は情報量が最も少ないことから，埋め込み利用されることが多い．実際，この特性を利用した埋め込み手法として LSB 法（章 3）が提案されており，ステガノグラフィにおける最も一般的な埋込み手法として知られている．

3　LSB 法

LSB 法とは画像の各ピクセルにおける LSB のみを埋め込み対象とすることで，埋め込みによる画像の劣化を抑えた手法である．この手法は，図 1 に示すように，カバーデータ中の LSB に対しテキスト情報のバイナリ表現をそのまま置き換えることで埋め込みを行なう．

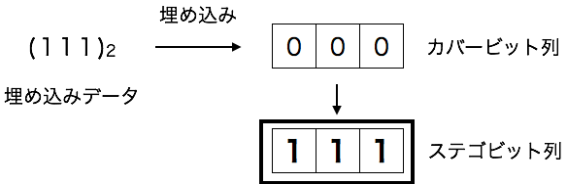

図 1　LSB 法での埋め込み

また，埋め込み時に LSB をテキスト情報のバイナリ表現でそのまま置換することから，図 2 のようにス

テゴデータの LSB のからテキスト情報を直接抽出することができる．

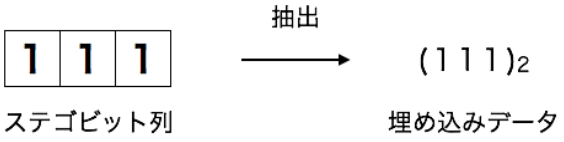

図 2　LSB 法での抽出

単純なアルゴリズムであり実装が容易である反面，埋め込み後のビット誤りが起きやすく，図 1 のようにすべてのビットが反転してしまう場合もある．このため，ステゴデータの画質が劣化しやすく，ステゴデータの隠蔽性に問題を引き起こす可能性がある．

このような問題から，これまで LSB 法におけるビット誤りを改善した手法が多数提案されてきた．それらの中でも本論文で扱う誤りパターン埋め込み法は，ビット誤りを大きく改善した手法として知られている．これについて事象で述べる．

本論文ではカバーデータのうち，埋め込みに対象となるビット列をカバービット列，ステゴデータ中のデータが埋め込まれたビット列をステゴビット列と呼ぶ．また，カバービット列とテキスト情報のバイナリ表現のビット長をそれぞれ n，m，ステゴビット列中の誤りビットの数を d として，$\frac{m}{n}$ を埋め込み率，$\frac{d}{n}$ を誤り率とする．

埋め込み率，誤り率は埋め込み評価指標として利用される．埋め込み率はいかに効率よくテキスト情報を埋め込めているかを表しており，高いほどに多くの情報を埋め込むことができる．また，誤り率は埋め込みによるビット誤りの発生頻度を表しており，一般的に誤り率が高いほど画質が劣化する傾向にある．LSB 法では埋め込み率は常に 100 %であるが，一方でビット誤りが起きやすいことから誤り率が高い．

4　誤りパターン埋め込み法

LSB 法をもとに，より低い誤り率での埋め込みを実現した手法として誤りパターン埋め込み法が知られている．この手法では次章にて述べる変換方式を用いてテキスト情報のバイナリ表現を，より長くハミング重みの小さいビット列である誤りパターンに変換する．そして，カバービット列をそのまま置き換えるのではなく，カバービット列と誤りパターンとの排他的論理和でカバービット列を置き換える（図 3）．

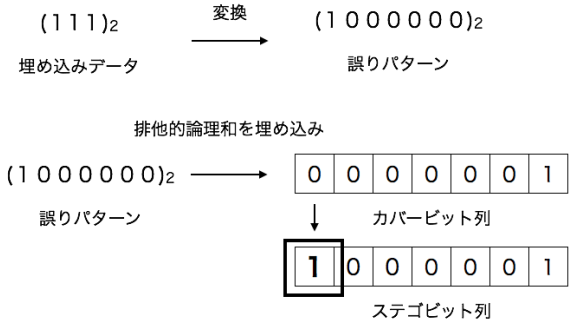

図 3　誤りパターン埋め込み法での埋め込み

また，カバービット列とステゴビット列との排他的論理和をとることで誤りパターンを抽出することができ，その誤りパターンをテキスト情報へと逆変換することでテキスト情報の抽出を完了する（図 4）．

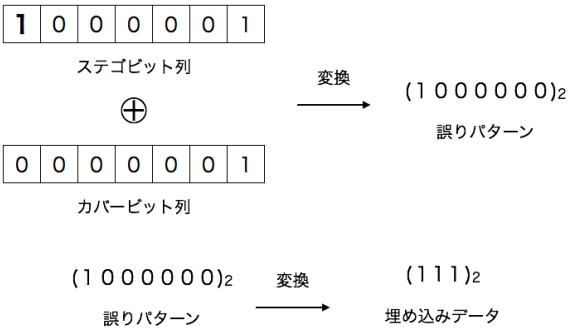

図 4　誤りパターン埋め込み法での抽出

埋め込みに排他的論理和を用いることから，誤りパターンにおける「1」の対応するビットのみがカバービット列中で置き換わるため，ビット誤りを大幅に抑えることができる．一方でより長いビット列を使用することから結果としてに埋め込み率は減少する．

この手法ではステゴビット列中で発生するビット誤りは誤りパターンのハミング重みと同じであり，生成される誤りパターンは埋め込まれるテキスト情報によることから，誤り率はテキスト情報の内容に依存する．そのため同じテキスト情報を埋め込みデータとした場合，カバーデータによらず誤り率は等しい．また，テキスト情報の抽出時にカバービット列とステゴビット列の比較を用いることから，この手法は埋め込みデータの抽出にカバーデータを必要とする手法である．

5　誤りパターンの生成方法

一般的な誤りパターンの生成方法として各文字のバイナリ表現と誤りパターンとの対応を示した誤りパターンテーブルを利用した方法がある．誤りパターンテーブルの例を表 1 に示す．この方法では埋め込み時，抽出時にテーブルを参照することで相互の変換を行う．単純な方式であり実装が容易である一方で，テキスト情報の各文字のバイナリ表現を m ビットとしたとき，最大で 2^m 個の誤りパターンを対応付ける必要があることから，テーブルが膨大となりやすく，強いメモリ制約下での実装が困難であるという問題点がある．

表 1　誤りパターンテーブル

埋め込みデータ	誤りパターン
000	0000000
001	0000001
010	0000010
011	0000100
100	0001000
101	0010000
110	0100000
111	1000000

この問題の解決策として文献 [2] では，コスト付き符号化を用いた誤りパターンテーブルの効率的実装方法が提案されている．

本論文では誤りパターンテーブルを用いない生成方法として，Shalkwijk の数え上げ符号 [3] を用いた誤りパターンの動的生成手法を提案する．

5.1　Slalkwijk の数え上げ符号

Shalkwijk の数え上げ符号とは長さ n，ハミング重み k の 2 進数列 x の集合に対し 10 進数 $i(x)$ を一意に割り当てることができる符号化手法である．なお i(x) は

$$0 \leq i(x) \leq {}_nC_k$$

を満足する．

またパスカルの三角形を用いて x，$i(x)$ 間の相互変換が可能である．この方法ではパスカルの三角形における $n+1$ 段目，左から $k+1$ 番目の値を起点として変換を行う．一例として $x = 010100$ としたときの符号

1
1　1
1　2　1
1　3　3　1
1　4　6　4　1
1　5　10　10　5　1
1　6　15　20　15　6　1
1　7　21　35　35　21　7　1

図 5　パスカルの三角形

化について考える．この x は $n = 6$，$k = 2$ であるからまず頂点の 1 から数えて 7 段目，左から 3 番目の値である 15 に注目する．ここで x の MSB から 1 ビットずつ値を取り出していき，0 であれば右斜め上に，1 ならば左斜め上の値に注目点を移動する．また取り出した値が 1 であったとき注目点を移動する前の値から見て右斜め上の値を記憶し加算していく．今回の場合，MSB が 0 であるから右斜め上の 10 に移動する．そして次のビットは 1 であるから右斜め上の 6 を記憶し，左斜め上の 4 に移動する．これを繰り返し，頂点に達した時点で終了となる．$x = 010100$ とした場合，最終的に加算していった値として 8 が得られ，これが x に割り当てられる $i(x)$ となる．

次に $i(x)$ から x への復号化について考える．復号化では注目点から見て右斜め上の値と $i(x)$ とを比較し，MSB から順に各ビットの値が決定される．比較対象となる値を n として $i(x) < n$ であればビットは 0 として注目点を右斜め上の値移動する．$i(x) \geq n$ であればビットは 1 として左斜め上の値に移動し，$i(x) = i(x) - n$ として i(x) の値を減らしていく．今回の場合，起点である 15 から見て右斜め上の値は 10 であり，これよりも 8 は小さいことから MSB は 0 であることが決まり，そのまま右斜め上の 10 に移動する．次に，10 の位置から見ると右斜め上の値は 6 であり 8 のほうが大きいことから左斜め上の 4 に移動し，次のビットは 1 とする．このとき，値 8 から 6 を引いた値である 2 を記憶する．これを繰り返していくことによって最終的に 010100 が得られる．これが $n = 6, k = 2$ とした場合に 8 に対応する x ということになる．最終的に $n = 6, k = 2$ の 2 進数列 x の集合は 10 進数 $i(x)$ が表 2 のように割り当てられる．

表 2　$n = 6, k = 2$ の場合の対応付け

x	$i(x)$	x	$i(x)$
000011	0	010100	8
000101	1	011000	9
000110	2	100001	10
001001	3	100010	11
001010	4	100100	12
001100	5	101000	13
010001	6	110000	14
010010	7		

5.2　誤りパターンの動的生成

動的生成のアルゴリズム中ではテキスト情報の各文字のバイナリ表現を 2 進数列 x として与えることで，パスカルの三角形を用いて 10 進数 $i(x)$ へと変換する．このとき，x が一意のであるの一方で，異なる x に対し同じ $i(x)$ が割り振られる場合がある．例えば，表 2 で x = 000011 に割り当てられる $i(x)$ は 0 であるが，x = 000000 に対しても同様に 0 が割り当てられる．そこで x の長さ n，ハミング重み k を元に式 (1) で求められる offset_k を i(x) に加算することで一意の誤りパターンとする．

$$\begin{aligned} \mathrm{offset}_k &= \sum_{i=0}^{k-1} {}_nC_i \\ \mathrm{offset}_0 &= 0 \end{aligned} \tag{1}$$

この方法を用いることで誤りパターンを動的に生成し，強いメモリ制約下での誤りパターン変換をを可能にすることができる．

6　実験

6.1　実験手順

実験手順の概要を図 6 に示す．以下に示す手順で実験を行った．実験には 256×256px の 8 ビットグレイスケール Bitmap である SIDBA 画像 30 枚を使用し，埋め込みに使用するテキスト情報は当確率で発生する 8bit コードの列とした．テキスト情報の生成には 623 次元に分布する乱数発生器である SFMT を用いた．

(1) テキスト情報の各コードを章 5.2 にて述べた手法で動的に誤りパターンへと変換し，変換した誤りパターンと画像の LSB 平面との排他的論理和を画像へと埋め込む．
(2) 埋め込み前後の画像を比較し，誤り率，SSIM を算出する．
(3) 誤りパターン長を 8bit から 256bit まで変化させ（1），（2）を繰り返す．
(4) 画像を入れ替えて（3）を繰り返す．

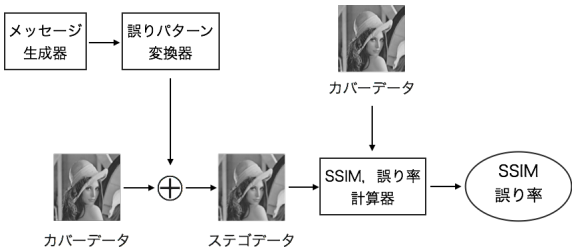

図 6　実験概要

6.2　実験結果

6.2.1　誤り率と埋め込み率

誤り率の計測結果と文献 [2] において示されている誤りパターン埋め込み法における誤り率の理論的下限曲線を図 7 に示す．なお，誤りパターン埋め込み法における誤り率は埋め込まれるテキスト情報にのみ依存するため，同じテキスト情報が埋め込まれたすべての画像について計測される誤り率は等しい．

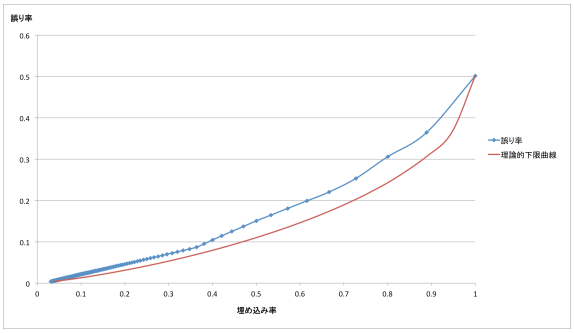

図 7　埋込率に対する誤り率の変化

図 7 より，実際に計測した誤り率は理論的下限曲線と同様，埋め込み率が増加するとともに誤り率も増加し，理論的下限曲線に比べ曲率の低い曲線を描いた．

6.2.2　SSIM と誤り率

30 枚の SIDBA 画像のうち，SSIM の最も高い画像 10 枚と最も低い 10 枚の埋込率に対する SSIM の変化を図 3 に示す．

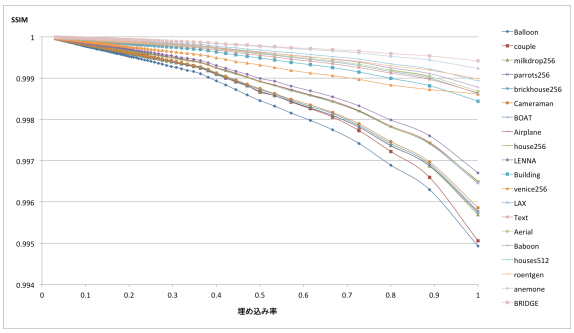

図 8　埋込率に対する SSIM の変化（画像ごと）

図 8 よりすべての埋め込み率について SSIM が 0.99 を上回っていることがわかる．一般的に SSIM は 0.98 以上で比較された画像間の見分けがつかないと言われていることから，どのような埋め込み率に対しても画質が大きく劣化することはないと考えられる．このことから，埋め込みを LSB 平面に限定した場合，埋め込み率，誤り率によって画質が大きく劣化することはなく，画質劣化の面でステガノグラフィ技術の隠蔽性に大きく影響を与えることはないということがわかる．同時に図 8 より，画像ごとに SSIM の変化の様子が異なることがわかる．これについて，実験に使用した画像を主観的に観察したところ，図 9 のように画像全体の複雑度が高い画像ほど SSIM の変化が小さく，一方で図 10 のように複雑度の低い大きな領域を含む画像ほど SSIM の変化が大きくなるという傾向が見られた．ここで複雑度とは画像全体における画素間の明暗の変化の多さを表しており，複雑度が高いほどにノイズ画像のようになる．これは，埋め込まれるデータ内の 0,1 の発生はランダムであり，埋め込みによる画像の劣化がノイズ状に発生することから，図 9 のように複雑度が高い画像であれば画像中にノイズが溶け込んみ，目立たなくなってしまうためであると考えられる．

6.2.3　2 ビットプレーン以降の埋め込みでの実験結果

章 6.2.2 の結果から，隠蔽性に影響を及ぼさない範囲で 2 ビットプレーン以降への埋め込みも可能なのではないかと考え，2〜4 ビットプレーンまでの埋め込み

図 9　BRIDGE.bmp

図 10　Balloon.bmp

を行った．実験手順はこれまでと同様の手順で行い，実験対象となるビットプレーン以下のビットプレーンについては埋め込み率 100 %で埋め込みを行った．実験結果をそれぞれ図 11〜13 に示す．

図 11 より埋め込み率 0.9 前後で SSIM が 0.98 を下回り始めていることがわかる．このことから，LSB への埋め込みを 100 %として 2 ビットプレーンまで埋め込みを行うことで，ほとんどの画像に対し，隠蔽性に影響を及ぼさない範囲で，約 1.9 倍のテキスト情報の埋め込みが実現できるといえる．

図 12 より 3 ビットプレーンまで埋め込み率 100

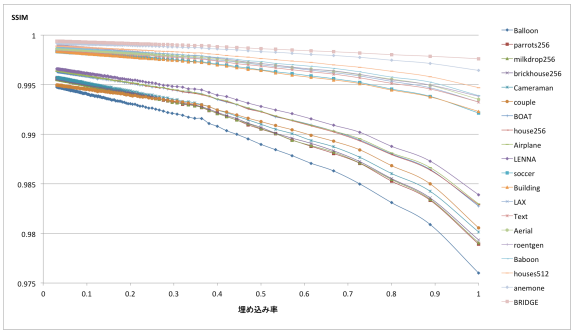

図 11　埋込率に対する SSIM の変化（2 ビットプレーン使用）

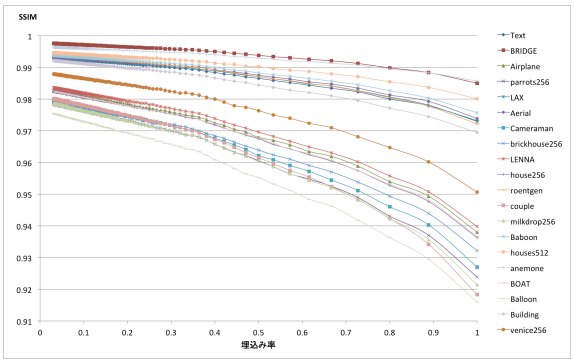

図 12　埋込率に対する SSIM の変化（3 ビットプレーン使用）

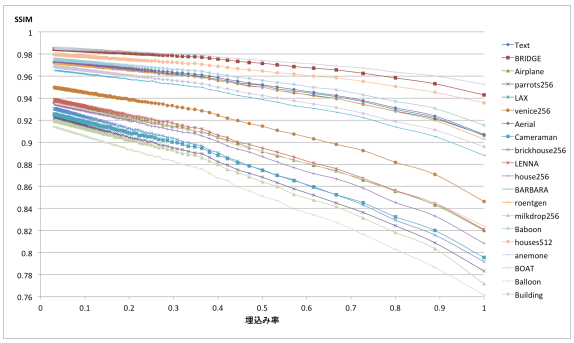

図 13　埋込率に対する SSIM の変化（4 ビットプレーン使用）

%での埋め込みを行ったとしても SSIM が 0.98 を上回る画像があることがわかる．また，図 13 においても最大で埋め込み率 0.2 前後までテキスト情報を埋め込むことができることがわかる．

ここで，図 9，10 のステゴデータをそれぞれ図 14，15 に示す．ステゴデータは LSB から 3 ビットプレーンまで埋め込み率 100 %で埋め込みを行ったものである．図 15 はもとの画像に比べ視覚的に大きく劣化が生じており，一方図 14 は劣化が小さく元の画像からほとんど変化していない．これらのことから適切な画像を選択することでより LSB のみへの埋め込みに比べ 3 倍以上のテキスト情報の埋め込みが可能になるといえる．

図 14　BRIDGE.bmp のステゴデータ

図 15　Balloon.bmp のステゴデータ

6.2.4　メッセージ長が同じ場合の劣化について

埋め込み範囲を広げて同じメッセージ量を埋め込む場合，より長い誤りパターンを用いることができることから，ステゴデータの誤り率を下げることができる．

例えば，LSB のみへ埋め込み率 100 %，つまり誤りパターン長を 8 ビットとして埋め込みを行った場合，256×256px の画像であれば 8192 文字の埋め込みが可能である．これを 2 ビットプレーンにまで埋め込み範囲を拡張して埋め込みを行った場合，単純に埋め込み範囲が倍になることから誤りパターン長を 2 倍の 16 ビットとして埋め込みを行うことができ，誤り率を半減させることができる．

これについて，LSB のみへ埋め込みを行った場合，2 ビットプレーンまで埋め込んだ場合について，どのような違いが見られるかを検証するため，今までと同様の実験手順で比較を行った．

30 枚の SIDBA 画像のうち BRIDGE.bmp（図 9）を使用した場合の実験結果を図 16 に示す．この結果から埋め込む文字数にかかわらず，2 ビットプレーンまで埋め込みを行った場合の SSIM が LSB のみの場合を下回っていることがわかる．このことから埋め込み範囲を拡張することによって画質劣化を抑えることはできないといえる．

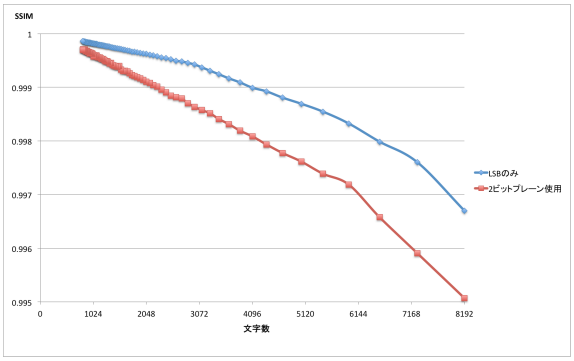

図 16　埋込率に対する誤り率の変化

7　実験環境

- OS : Mac OS X 8，9，Windows 7，Ubuntu 13.10
- IDE : Intellij IDEA 13.0

8　画像の入手元

1. 神奈川大学
   http://www.ess.ic.kanagawa-it.ac.jp/app_images_j.html
2. 京都大学
   http://vision.kuee.kyoto-u.ac.jp/IUE/IMAGE_DATABASE/STD_IMAGES/index.html
3. Ashi-LAB
   http://asssy.sakura.ne.jp/idba.html
4. Rensselaer Polytechnic Institute http://www.cipr.rpi.edu/resource/stills/index.html

9　おわりに

本研究では誤りパターン埋め込み法における埋め込み率と誤り率，画質劣化とのトレードオフ関係を実験的に明らかにした．これによって，埋め込みを LSB 平面に限定した場合，埋め込み率，誤り率によって画質が大きく劣化することはなく，ステガノグラフィ技術の隠蔽性に大きく影響を与えることはないということがわかった．さらに，この結果から埋め込み範囲を広げての SSIM の計測を行い，2 ビットプレーンを使用することでほとんどの画像に対し 1.9 倍近くのテキスト情報の埋め込みが可能であること，画像を適切に選択することで 3 倍以上ものテキスト情報の埋め込みが可能であることを示した．

また，Shalkwijk の数え上げ符号を用いた誤りパターンへの動的な変換方法を提案することで，強いメモリ制約下における誤りパターンテーブルの実装問題に対する解決策を示した．しかし，本実験で使用した環境は決してメモリ制約の強い環境ではないため，今後実際に強いメモリ制約下での実験を行う必要がある．

参考文献

[1] Z.Wang, A.C.Bovik, H.R.Sheikh, and E.P.Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, Vol. 13, No. 4, pp. 600–612, April 2004.

[2] 合田翔, 渡辺峻, 松本和幸, 吉田稔, 北研二．コスト付き符号化を用いたステガノグラフィ．信学技法 IT, Vol. 113, No. 153, pp. 5–9, 7 2013.

[3] J.P.M.Shalkwijk. An algorithm for source coding. *IEEE TRANSACTIONS ON INFORMATION THEORY*, Vol. IT-18, No. 3, pp. 395–399, May 1972.