

# 誤りパターン埋め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  
Researcher : Ippei NAWATE

指導教員：福岡 久雄  
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

### 概要

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。ステガノグラフィ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誤りパターンというビット列に変換し、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し、2 ビット～4 ビット列目を使用することで LSB のみを埋め込みに使用した場合の 1.9～3.0 倍のテキスト情報を第三者から埋め込みが主観的に検知されない範囲で埋め込み可能であることを示す。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。

### Abstract

This paper studies a steganography technique to embed text information into gray scale images. The method of embedding error patterns is one of the steganography methods. In embedding process, each character in text information is converted into an error pattern, a binary string longer in length and lighter in Hamming weight than the binary data of the character, then the LSB plane of the image is exclusive-ORed with the error pattern. We have evaluated the image quality degradation by embedding error patterns and have shown that using the second to fourth LSB planes for embedding enables us to embed 1.9 to 3.0 times more text information with almost perfect imperceptibility for image degradation. This paper also proposes a method of dynamically generating error patterns using the enumerative code proposed by Shalkwijk.

### キーワード

ステガノグラフィ, セキュリティ, 誤りパターン, 数え上げ符号

## 目次

1 はじめに .....	1
2 ステガノグラフィとは .....	1
3 LSB 法 .....	2
4 誤りパターン埋め込み法 .....	2
5 誤りパターンの生成方法 .....	4
5.1 Slalkwijk の数え上げ符号 .....	4
5.2 誤りパターンの動的生成 .....	5
6 SSIM .....	5
7 カバーデータの劣化評価 .....	5
7.1 評価手順 .....	5
7.2 評価結果 .....	6
7.2.1 埋め込み率に対する誤り率 .....	6
7.2.2 埋め込み率に対する画質劣化 .....	6
7.2.3 2 ビット列目以降へ埋め込んだ場合の画質劣化 .....	8
7.2.4 埋め込む文字数を一定にした場合の画質劣化 .....	8
8 おわりに .....	9
付録 A 実験環境 .....	i
付録 B 画像の入手元 .....	i

## 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。安全に通信を行う手段として送信内容の暗号化が挙げられる。暗号化を施すことで、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が、通信行為そのものは第三者に認知されてしまうことから、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。それに対し、ステガノグラフィでは送信内容を別のデータに埋め込み、第三者から通信行為の存在そのものを隠蔽することで、より安全な通信を実現することができる。

しかし、データの埋め込みによって、埋め込みに使用されたデータ（これをカバーデータという）内に誤りビットが発生し、それに伴うカバーデータの劣化が情報の隠蔽性に問題を引き起こしてしまう。この誤りビットが発生する確率を誤り率という。ステガノグラフィではカバーデータの劣化が目立たないようにデータを埋め込む必要があるが、同時に埋め込み可能なデータ容量は制限されてしまうことから、低い誤り率でより多くの情報を効率よく埋め込むことが重要である。この埋め込みの効率を埋め込み率という。

ステガノグラフィの代表的な手法として LSB 法が知られている。LSB 法とは単純にカバーデータの LSB と埋め込みデータのバイナリ表現をそのまま置き換えることによって埋め込みを行なう手法である。この手法は、高い埋め込み率を有するが、一方で、誤りビットが発生しやすくカバーデータが劣化しやすいという欠点を持つ。その欠点を考慮した手法として誤りパターン埋め込み法がある。この手法は埋め込みデータを誤りパターンというビット列に変換し、排他的論理和を用いて埋め込みを行なうことで LSB 法に比べ優れた誤り率を実現している。

文献 [1] では埋込率に対する誤り率という側面から埋め込み手法の性能評価を行っており、誤り率について誤りパターン埋め込み法が LSB 法に比べ優れているという結果を得ている。誤り率はステガノグラフィを評価する上で重要な要素であるが、主観的評価の側面から言えば、必ずしも低い誤り率が高い性能を示すとは限らない。例えば、誤りの生じたビットごとに重みをつけ平均二乗誤差から求められる PSNR という画

質劣化指標があるが、この指標は主観評価と同様の結果が得られない場合が多々あるということが文献 [2] において示されている。つまり、十分低いと考えられる誤り率で埋め込みを行ったとしてもカバーデータによっては大きく劣化が生じてしまうということである。このようなことから、ステガノグラフィを評価する上では誤り率だけではなく、カバーデータの主観的劣化についても測定することが誤り率同様重要であると考えられる。

そこで、本論文ではグレースケール画像にテキスト情報を埋め込むステガノグラフィ技術を対象とし、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。その結果に対し、誤りパターン埋め込み法を主観的観点から評価し、考察を行なう。なお、画質劣化の評価指標として主観評価に近い評価指標である SSIM[2] (Structual Similarity) を用いる。また、誤りパターンテーブルの肥大化問題 [1] 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データ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もある。

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。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、つまり第三者に除去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であり、その情報を外部から確認できることが重要である。そのため埋め込まれる証拠データそのものには大きな価値はない。

一方、ステガノグラフィでは秘密データを隠蔽するために、カバーデータへの埋め込みを行なう。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データそのものに大きな価値があり、埋め込まれた秘密データの存在そのものを第三者に認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、多量のデータの埋め込みはカバーデータを劣化させるため、第三者によって埋め込みを検知され

る危険性が高まってしまう。そのため、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。

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

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、これを利用した埋め込みが主である。本論文で埋め込み対象とする画像では、画像中の各画素の下位ビットが冗長性であり、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カバーデータの主観的变化は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、この特徴を利用した埋め込み手法として LSB 法が提案されており、ステガノグラフィにおける最も一般的な埋め込み手法として知られている。

### 3 LSB 法

LSB 法とは画像の各画素における 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（以下埋め込み、抽出されるデータはバイナリデータでありバイナリデータであることをいちいち断らない）をそのまま置き換えることで埋め込みを行なう。このテキスト情報が埋め込まれたカバーデータをステゴデータという。

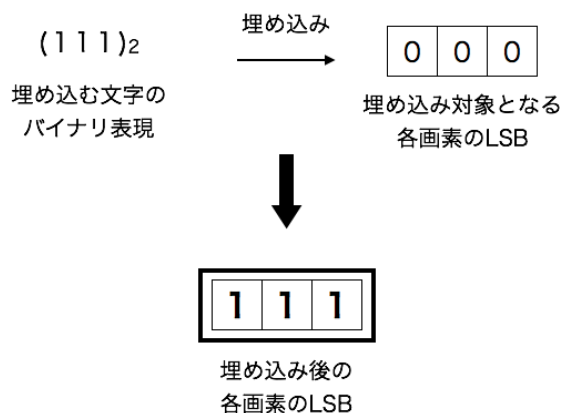


図1 LSB 法でのデータの埋め込み

テキスト情報の抽出時には図 2 のように、ステゴデータにはテキスト情報がそのまま埋め込まれていることから、特別な処理なしに抽出することができる。

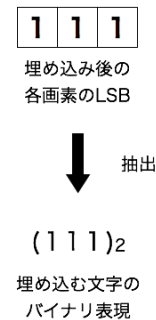


図2 LSB 法でのデータの抽出

単純なアルゴリズムであり実装が容易である反面、埋め込み後に誤りビットが発生しやすく、ステゴデータの誤り率が高くなることから、ステゴデータの劣化によって隠蔽性に問題を引き起こしやすいという欠点がある。この欠点に対し、これまで LSB 法をもとに多数の手法が提案されてきた。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、優れた誤り率を有する手法として知られている。

本論文ではカバーデータのうち、埋め込み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、埋め込み率は  $\frac{m}{n}$ 、誤り率は  $\frac{d}{n}$  として計算される。基本的に埋め込み率が高いほど多くの情報を埋め込むことができ、誤り率が高いほどステゴデータの画質が劣化する傾向にある。

### 4 誤りパターン埋め込み法

LSB 法に比べ優れた誤り率を有する手法として誤りパターン埋め込み法が知られている。

LSB 法では埋め込むごとに  $\frac{1}{2}$  の確率で誤りビットが発生することから、誤り率を  $\Delta$ 、埋め込み率を  $R$  としてこれらの関係は

$$\Delta = \frac{R}{2}$$

で表される。

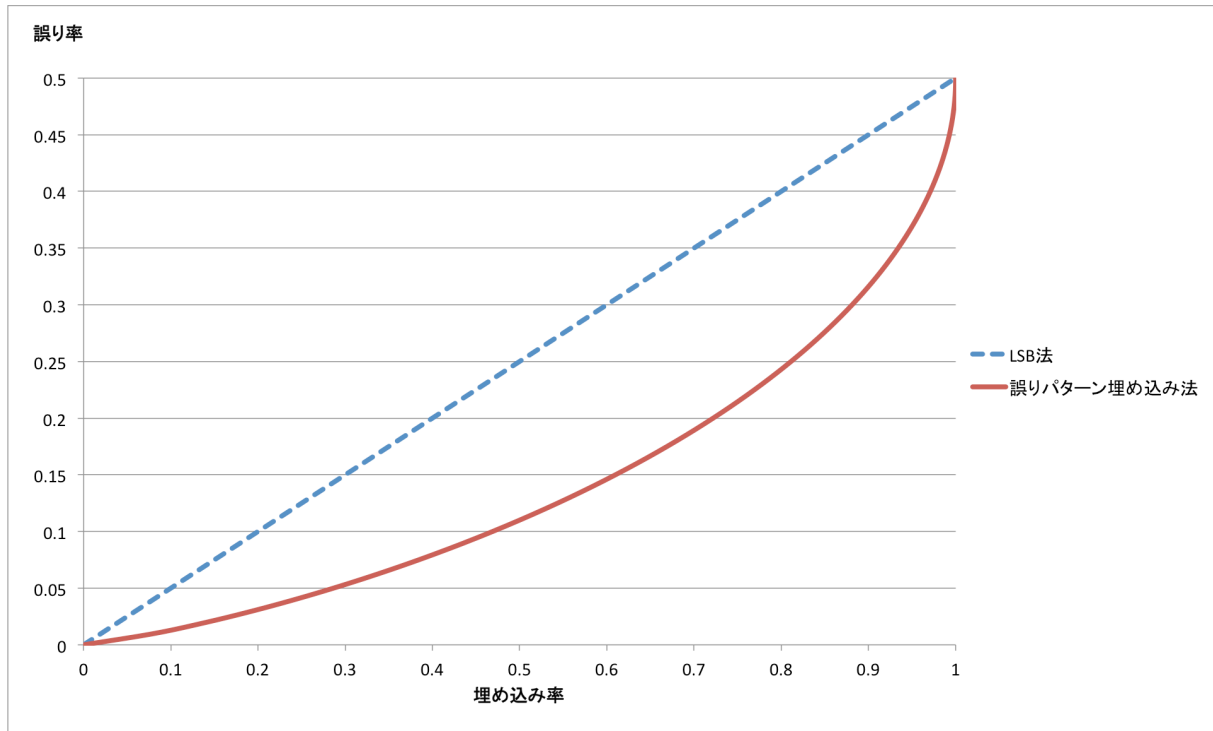


図3 LSB法と誤りパターン埋め込み法の性能比較

一方、誤りパターン埋め込み法の埋め込み率は次式のエントロピー関数  $H(\Delta)$  で表される。

$$H(\Delta) \triangleq H(\Delta, 1 - \Delta) \\ = -\Delta \log_2 \Delta - (1 - \Delta) \log_2 (1 - \Delta)$$

図3にLSB法と誤りパターン埋め込み法の埋め込み率と誤り率の関係を表したグラフを示す。図3より、端点を除き、明らかにLSB法に比べ誤りパターン埋め込み法は誤り率について優れていることがわかる。

この手法では図4のように、テキスト情報を、冗長でかつハミング重みの小さいビット列である誤りパターンへと変換する。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置き換えを行なう。

テキスト情報の抽出時には図5のように、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埋め込まれた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変換することでテキスト情報を得る。このような抽出方法となることからステゴデータの受信者はカバーデータを所持している必要がある。

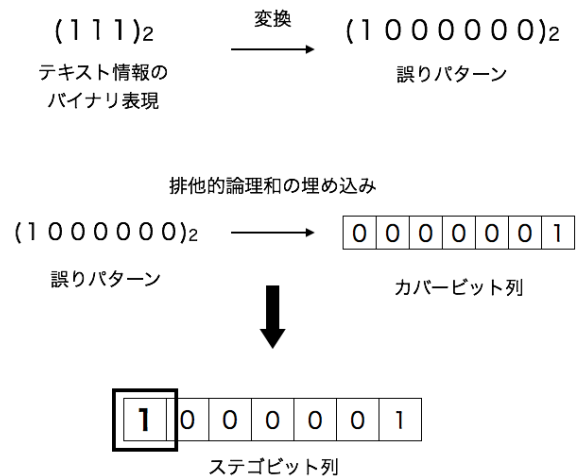


図4 誤りパターン埋め込み法でのデータの埋め込み

この手法では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、誤りビットの発生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埋め込み率は減少する。また、テキスト情報から生成される誤りパターンのハミング重みとステゴビット列中で発生する誤りビット

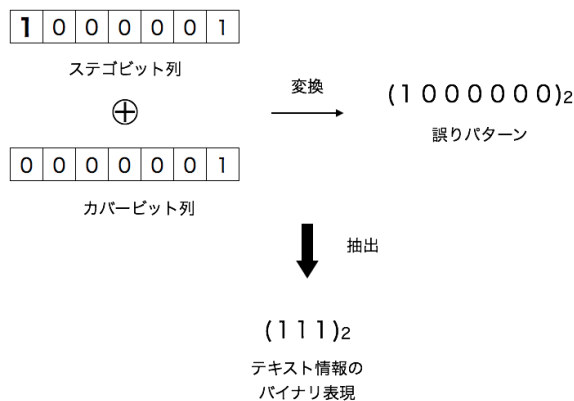


図5 誤りパターン埋め込み法でのデータの抽出

の数が等しくなるため、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くなる。

## 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文字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する方法がある。誤りパターンテーブルの例を表1に示す。

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

この方法ではテキスト情報の埋め込み時、抽出時にテーブルを参照することで誤りパターンの相互変換を行う。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になるという問題がある。

本論文では、この問題の解決策として 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。

### 5.1 Shalkwijk 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。

図6 パスカルの三角形

また、図6に示すような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処理を行う。一例として  $x = 010100$  としたときの符号化について考える。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。今回の場合、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  $i(x) < n$  であれば決定されるビットは 0 となり、注目点を右斜め上の値移動する。

$i(x) \geq n$ であれば決定されるビットは1となり、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

提案する手法ではテキスト情報の各文字を 2 進数列  $x$  として与えることで、10 進数  $i(x)$  へと変換する。この変換は 5.1 で述べたパスカルの三角形を用いて行なう。このとき、 $x$  が一意である一方で、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、本手法では  $x$  の長さ  $n$ 、ハミング重み  $k$  を用いて次式で求められる  $\text{offset}_k$  を  $i(x)$  に加算することで一意の誤りパターンとする。

$$\text{offset}_k = \begin{cases} 0 & (k = 0) \\ \sum_{i=0}^{k-1} n C_i & (k \geq 1) \end{cases}$$

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可能とする。

## 6 SSIM

SSIM は人間の視覚特性が考慮された画質評価の指標であり、主観評価との相関が高いことで知られている。SSIM の値は、比較される画像を  $x, y$  とし、それぞれの各画素とその周辺画素について画素値の平均  $\mu_y, \mu_x$  分散  $\sigma_x, \sigma_y$  共分散  $\sigma_{xy}$  を用いて以下の式で算出される。

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

ここで  $C_1, C_2$  は式の分母を 0 としないために加算される定数であり、任意のパラメータ  $K_1, K_2$  によって決定される。これらは画像  $x, y$  の 1 画素あたりの階調を  $L$  として次式で表される。

$$\begin{aligned} C_1 &= (K_1 L)^2 \\ C_2 &= (K_2 L)^2 \end{aligned}$$

代表的な画質劣化指標として PSNR があるが、これに比べ SSIM は主観評価との相関が高く、より正確に画質を評価できることから本論文では SSIM を用いることとした。

実際に SSIM を測定する際には、パラメータ  $K_1, K_2$ 、計算プログラム上で使用されるガウシアンフィルタの偏差値、サイズには SSIM の考案者である ZhouWang から自身が公開している MATLAB の SSIM 計算プログラム [4] とまったく同じ値を用いることとし、 $K_1 = 0.1$ 、 $K_2 = 0.3$ 、偏差値 = 1.5、フィルタサイズ = 11 とした。

## 7 カバーデータの劣化評価

### 7.1 評価手順

評価実験の手順の概要を図 8 に示す。以下に示す手順で実験を行い、この結果について考察を行った。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等確率で発生する 8 ビットコードの列とした。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。

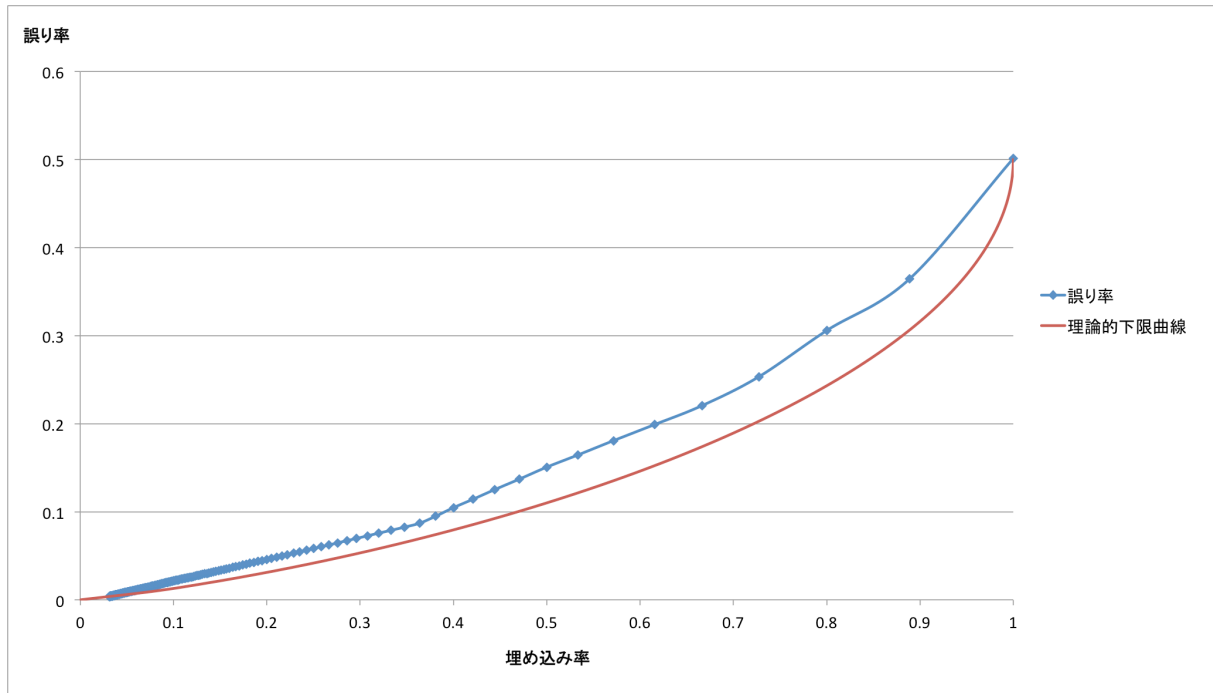


図7 埋め込み率に対する誤り率の変化

- (1) テキスト情報の各コードを 5.2 章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 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 8 ビットから 255 ビットまで変化させ (1)、(2) を繰り返す。
- (4) 画像を入れ替えて (3) を繰り返す。

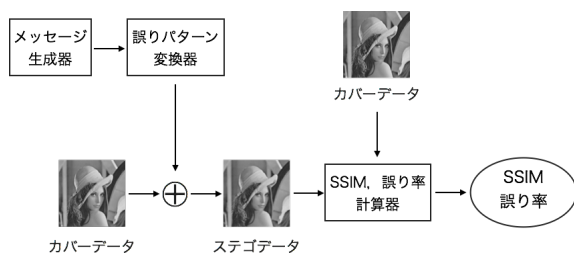


図8 実験概要

## 7.2 評価結果

### 7.2.1 埋め込み率に対する誤り率

誤り率の計測結果と文献 [1] 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カバーデータによらず等しいことから、すべてのカバーデータに対し同じ曲線を描く。

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。

### 7.2.2 埋め込み率に対する画質劣化

30 枚の SIDBA 画像のうち、SSIM の最も高い画像 10 枚と最も低い画像 10 枚の埋め込み率に対する SSIM の変化を図 9 に示す。

図 9 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 以上で埋め込み前後の画像の区別がつかないと言われていていることから、LSB のみをカバービット列とした場合、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、視覚的な隠蔽性に大きく影響を与える



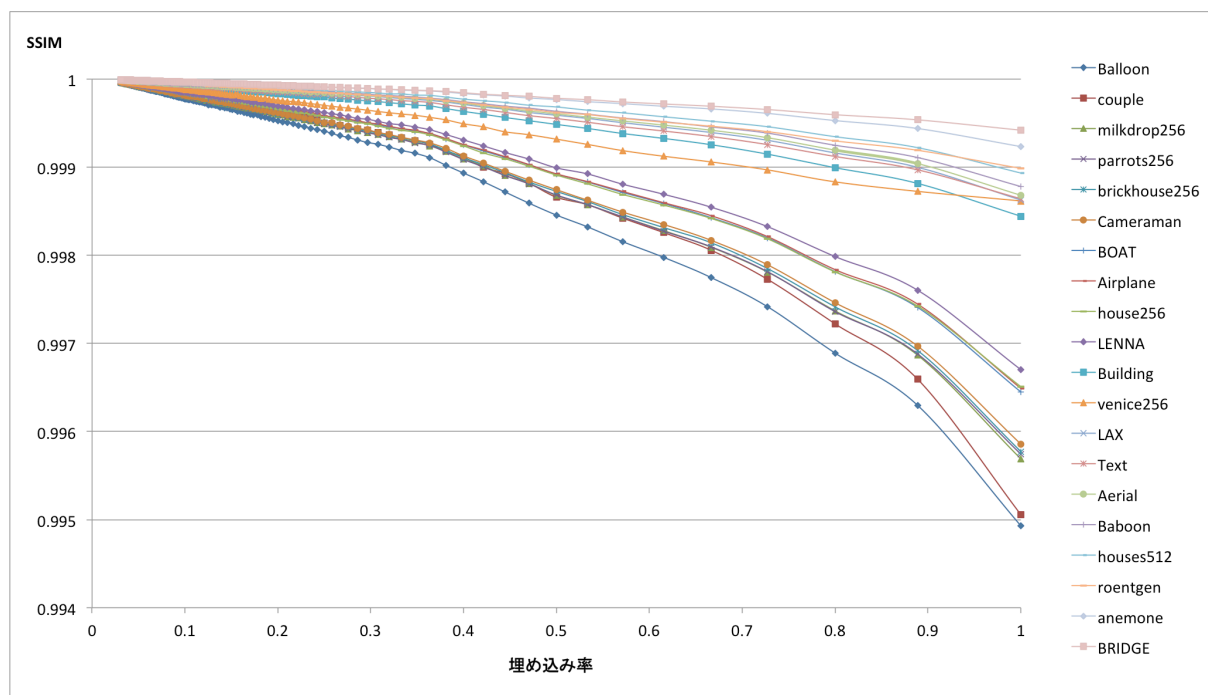


図9 埋め込み率に対する SSIM の変化 (画像ごと)



図10 BRIDGE.bmp



図11 Balloon.bmp

ことはないということがわかる。同時に図9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10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11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に近くなる。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10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み、目立たなくなってしまうためであると考えられる。

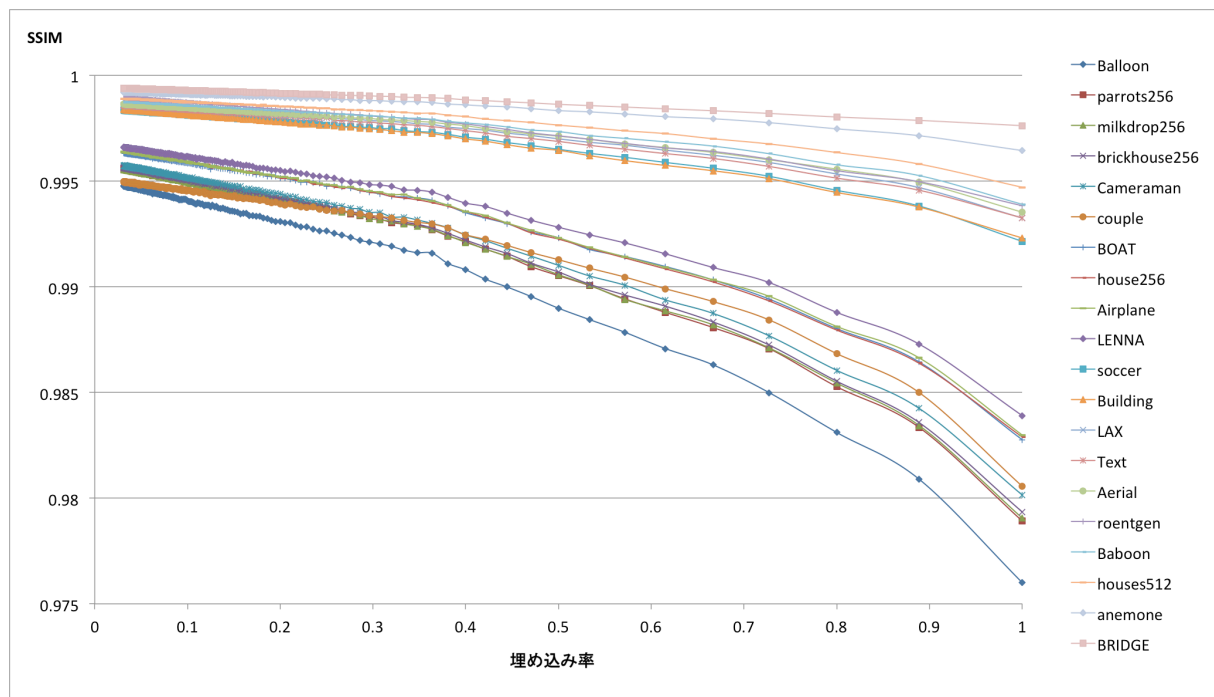


図 12 2 ビット列目を加えた場合

### 7.2.3 2 ビット列目以降へ埋め込んだ場合の画質劣化

章 7.2.2 より LSB のみをカバービット列とした場合にステゴデータの劣化はほとんど起こらず、視覚的な隠蔽性には大きく影響を与えないということがわかった。このことから視覚的な隠蔽性に影響を与えない範囲で 2 ビット列目以降への埋め込みも可能であると仮定し、さらに 2~4 ビット列目までをカバービット列として加え、これまでと同様の手順で実験を行った。このとき、カバービット列中の最上位ビット以外のビット列については埋め込み率 100 %での埋め込みを行った。実験結果をそれぞれ図 12~14 に示す。

図 12 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 と 2 ビット列目をカバービット列とすることで、ほとんどの画像に対し、視覚的な隠蔽性に大きく影響を与えない範囲で、約 1.9 倍のテキスト情報の埋め込みが可能であることがわかる。

図 13 より 3 ビット列目まで埋め込み率 100 %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 14 においても埋め込み率 0.2 前後まで SSIM が 0.98 を上回っていることがわかる。

ここで、図 10, 11 のステゴデータをそれぞれ図 15, 16 に示す。ステゴデータは LSB から 3 ビット列目まで埋め込み率 100 %で埋め込みを行ったものである。図 16 は複雑度が低く、もとの画像に比べ大きく画質劣化が生じている。一方で図 15 は複雑度が高く、元の画像からほとんど画質劣化が生じていない。これらのことから複雑度の高い画像を使用することでカバービット列を LSB のみとした場合に比べ、視覚的な隠蔽性に影響を与えない範囲で 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。

### 7.2.4 埋め込む文字数を一定にした場合の画質劣化

埋め込まれるテキスト情報の文字数を固定して誤りパターン長を長くした場合、埋め込みに必要となるステゴビット列は長くなるが、ステゴデータの誤り率を下げるができる。

例えば、カバービット列を LSB のみとして埋め込み率 100 %、すなわち誤りパターン長を 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 文字のテキスト情報の埋め込みが可能である。これに対し、誤りパターン長を 16 ビットとして 8192 文字のテキスト情報を埋め込んだ場合、LSB に加え 2 ビット列目をカバービット列とする必要があるが、誤

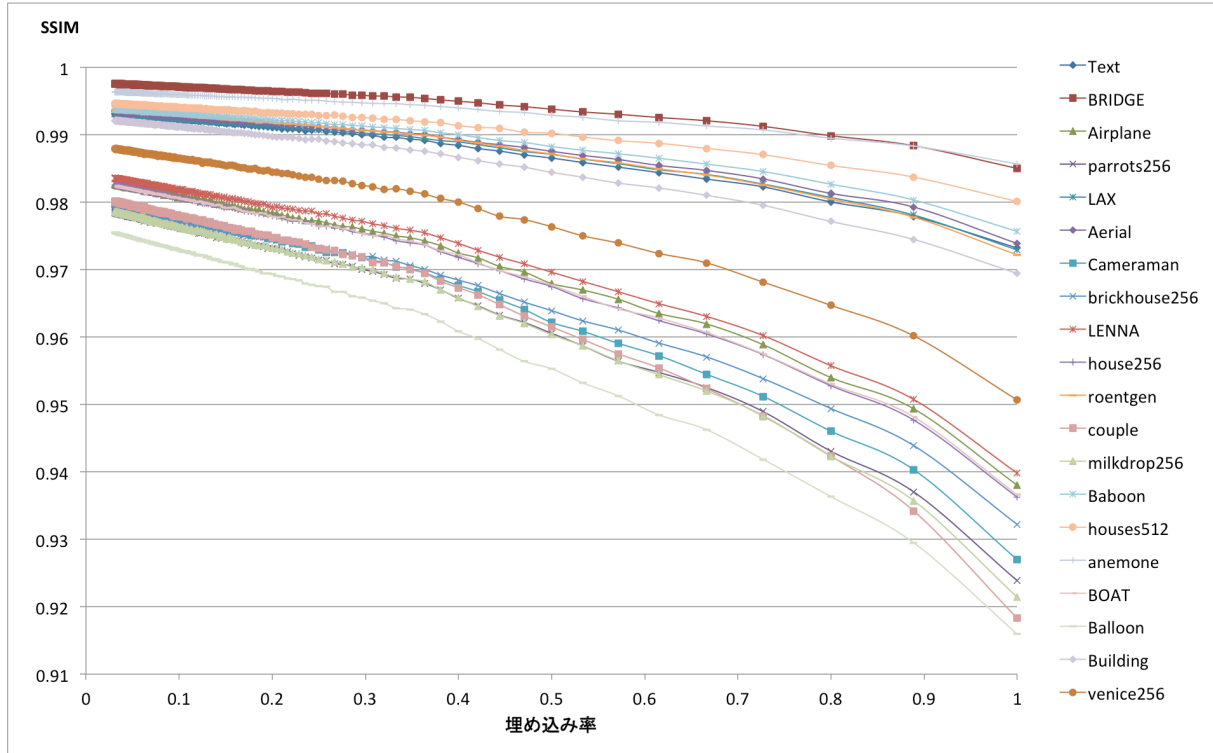


図 13 3 ビット列目を加えた場合

りパターン長が倍となることから誤り率を半減させることができる。

しかし、一般的に誤り率を下げることで画質劣化を抑えることができるが、2 ビット列目は LSB に比べ、各ビットが倍の情報量を持つことから誤りビットによるステゴデータの主観的变化が大きくなる。

これについて、埋め込む文字数を固定し、カバービット列を LSB のみとした場合、LSB と 2 ビット列目とした場合について、後者の場合により画質劣化を抑えることができると仮定し、これまでと同様の実験手順でステゴデータの画質劣化の比較を行った。

実験で使用した 30 枚の SIDBA 画像のうち BRIDGE.bmp (図 10) を使用した場合の実験結果を図 17 に示す。図 17 より、埋め込む文字数にかかわらず、カバービット列を LSB と 2 ビット列目とした場合の SSIM が、LSB のみとした場合の SSIM を下回っていることがわかる。この結果から、誤りパターン長を長くしたとしても、より重みのあるビット使用した場合、画質劣化を抑えることはできないということがわかる。また、他の画像についても同様の結果が得られている。

## 8 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、LSB のみをカバービット列とした場合、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い、ステゴデータの視覚的な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。この結果を受け、さらに 2 ビット～4 ビット列目をカバービットえて実験を行い、視覚的な隠蔽性に大きく影響を与えない範囲で、LSB のみをカバービット列とした場合に比べ 1.9～3.0 倍のテキスト情報を埋め込むことができることを示した。

また、Shalkwijk の数え上げ符号を用いた誤りパターンへの動的生成手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、本論文の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動作比較実験を行う必要があると思われる。

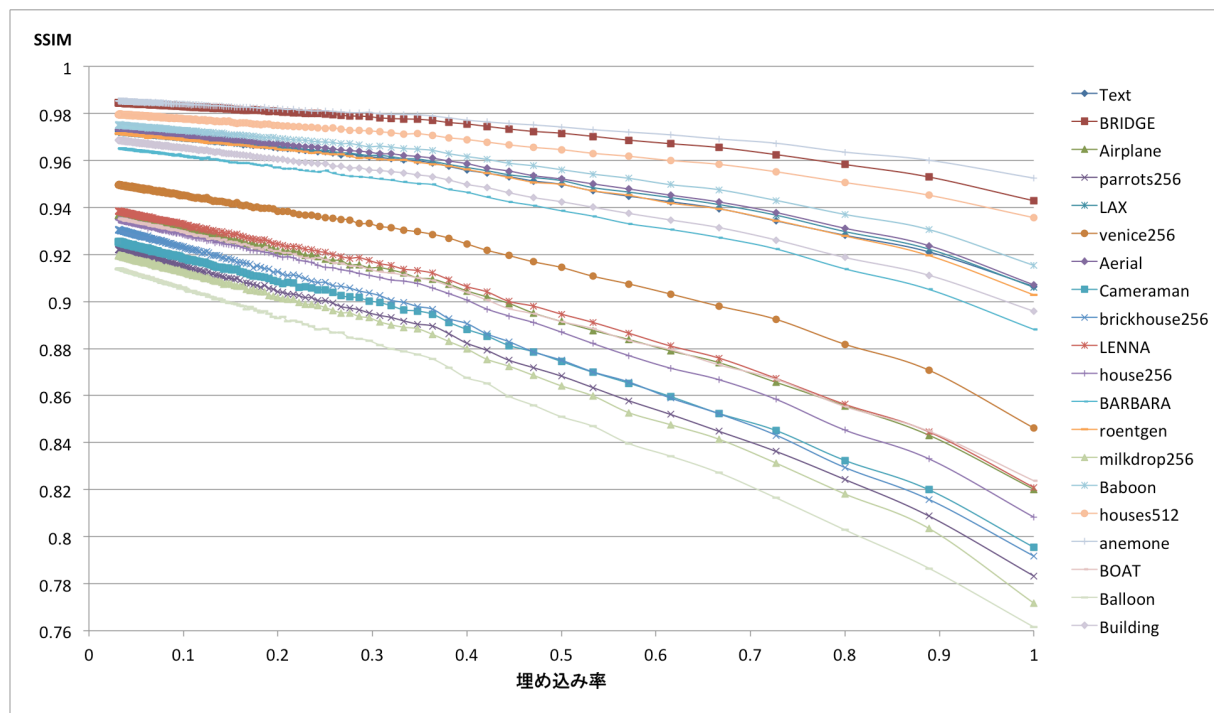


図 14 4 ビット列目を加えた場合

## 参考文献

- [1] 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. 信学技報 IT, Vol. 113, No. 153, pp. 5–9, 7 2013.
- [2] Z.Wang, A.C.Bovik, H.R.Sheikh, 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.
- [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.
- [4] Z.Wang. ssim\_index.m. [https://ece.uwaterloo.ca/~z70wang/research/ssim/ssim\\_index.m](https://ece.uwaterloo.ca/~z70wang/research/ssim/ssim_index.m).



図 15 BRIDGE.bmp のステゴデータ



図 16 Balloon.bmp のステゴデータ

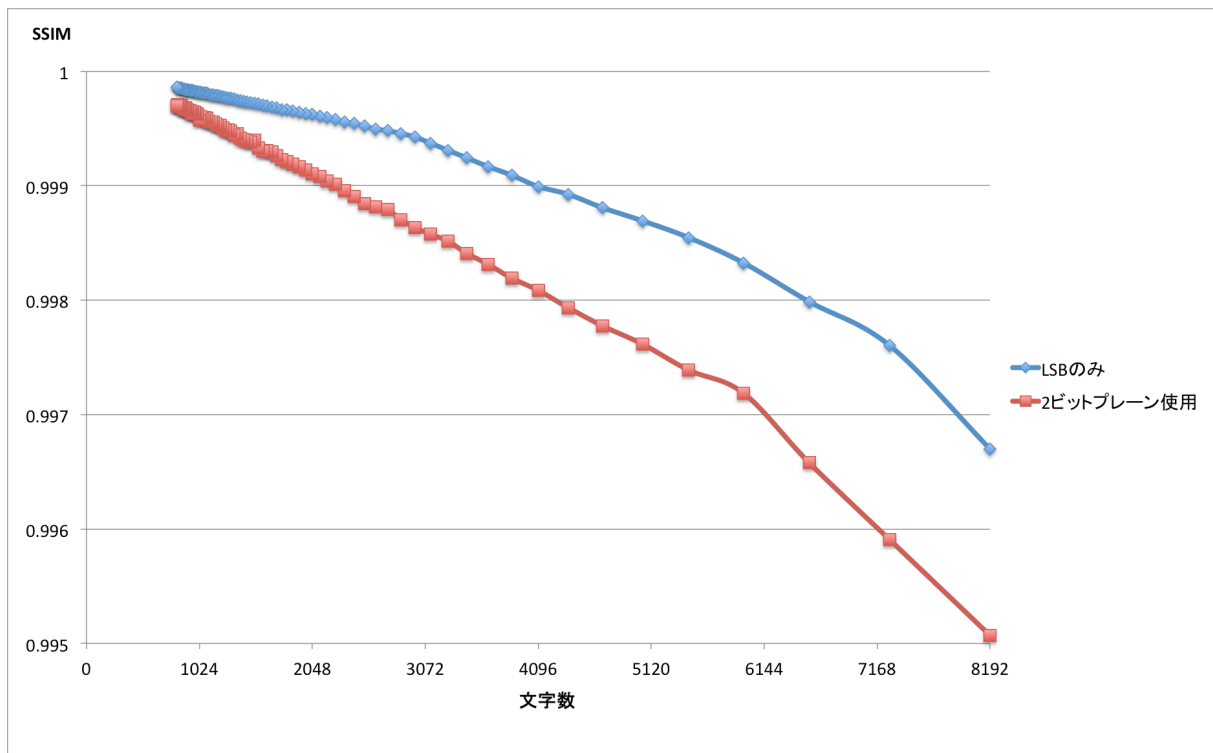


図 17 埋め込み率に対する誤り率の変化

## 付録 A 実験環境

- OS : Mac OS X 8, 9
- RAM : 8GB 1600 MHz DDR3
- IDE : IntelliJ IDEA 13.0

## 付録 B 画像の入手元

1. 神奈川大学  
[http://www.ess.ic.kanagawa-it.ac.jp/app\\_images\\_j.html](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](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>