

Be-code: ロゴ画像への情報埋め込み

Be-code: Information Hiding for a Logo Image

山田 一葉[†] 山下 泰奈[†] 丸山 哲平[†] 豊浦 正広[†] 茅 暁陽[†] 高津 聡[‡] 菅原 千尋[‡]
Kazuha YAMADA[†] Yasuna YAMASHITA[†] Teppei MARUYAMA[†] Masahiro TOYOURA[†]
Xiaoyang MAO[†] Satoshi TAKATSU[‡] and Chihiro SUGAWARA[‡]

[†] 山梨大学大学院医学工学総合教育部

[‡] 株式会社 クレスコ

[†] Interdisciplinary Graduate School of Medical and Engineering, University of Yamanashi

[‡] CRESCO Co., Ltd.

1. はじめに

カメラで URL などの情報を取得するために、二次元コードが広く一般に利用されている。従来の二次元コードは、QR コードに代表されるように画像自体に意味を持たないものが多く、コードを商品に貼り付けることでデザイン性を損ねることがあった。

本研究では、ロゴなどの二値画像を生かした二次元コードである Be-code(Binary-emblem code) を提案する。Be-code は、ロゴのエッジ部分に画素レベルの変更を加えることで、ロゴの見た目を保ったまま情報を埋め込むことができる。情報読み込みはカメラの観測画像から行うことができ、原画像の参照も必要としない。

二次元コードには QR コードの他にも、マトリクス上のグリッドの色に情報を埋めるカラーコード [1] が提案されている。カラーコードも画像自体には意味を持たず、デザイン性に乏しい。写真画像の輝度値変化に情報を埋め込む FPCODE [2] では、単純な画像に対しては適用ができず、写真を貼りつけることができるような場面のみに利用が限られる。ロゴ画像に対して情報埋め込みができるものには、三宅らの手法 [3] と Pramila らの手法 [4] がある。三宅らの手法 [3] は図形の輪郭線に情報を埋め込む。ロゴ画像自体のデザイン性を生かした情報埋め込みを実現したもの、ロゴを囲む輪郭線が必要となり、利用先によってはなおデザイン性を確保できない恐れがある。また、青色を用いて信号を埋め込むために、カラーで印刷される必要があるという問題もある。一方、Pramila らの手法 [4] は二値画像への情報埋め込みに Wu らの手法 [5] を利用する点において本手法と同じであるが、ロゴ画像全体を囲う実線が必要である。また、画像サイズは既知である必要があり、この制約がコードをデザインするときの妨げになることがある。さらに、コードの付近に文字列や絵が印刷された場合、正しくコードを読み取ることができない。

2. Be-code の概要

Be-code は図 1 に示す各要素で構成する。

エンブレムエリア (emblem area) には、ロゴ画像に情報埋め込みを行ったものを配置する。ロゴ画像への情報埋め込みの手法は、3.1 および 3.4 で述べる。観測画像からの情報埋め込みを行うときには、ロゴ画像のデザイ

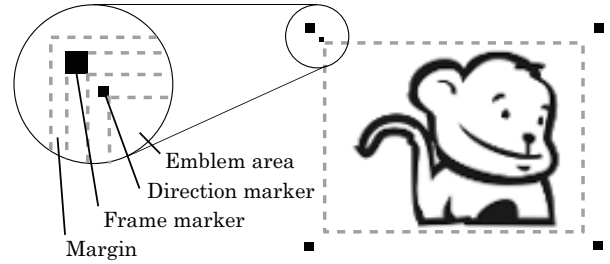


図 1 Be-code の構造

ン性を確保することと、観測環境に依らずに頑健に読み取れることの 2 つを満足する必要がある。

フレームマーカ (frame marker) は、観測画像から元のロゴ画像の座標系に変換するためのマーカである。それぞれのマーカは縦横 m 画素で黒色の矩形領域であり、エンブレムエリアの外側の四隅に配置される。フレームマーカの内側には、縦横 1 画素で黒色の方向検出マーカ (direction marker) を左上の隅にのみ配置し、画像が回転しても正しい向きで画像復元できるようにしている。フレームマーカと方向検出マーカの間には、1 画素分の白色のスペースを取る。フレームマーカを用いて原画像の座標系に変換する方法については、3.2, 3.3 で説明する。また、Be-code の周囲に文字や絵があってもフレームマーカが検出できるように、フレームマーカの外側には白色で 1 画素以上のマージン (margin) を設ける。

3. Be-code の実現

3.1. 可変得点を用いた情報埋め込み

ロゴ画像への情報埋め込みは画素値を変化させることで行うが、原画像のデザイン性確保のためには、人間が見て目立たないように画素値を変化させなければならない。これを実現するために、画像上の各微小領域のなめらかさや連結性を考慮して情報埋め込みを行う Wu らの手法を利用する [5]。Wu らの手法では、二値画像を 3×3 のブロックに分け、なめらかさと連結数に基づいて可変得点 (flippability score) を計算する。

なめらかさは、画素 (i, j) に対して水平 $N_h(i, j)$ 、垂直 $N_v(i, j)$ および対角 $N_{d_1}(i, j)$, $N_{d_2}(i, j)$ に対して計算

される。

$$\begin{aligned}
N_h(i, j) &= \sum_{k=-1}^1 \sum_{l=-1}^0 \mathcal{I}(\{p_{i+k, j+l} \neq p_{i+k, j+l+1}\}) \\
N_v(i, j) &= \sum_{k=-1}^1 \sum_{l=-1}^0 \mathcal{I}(\{p_{i+l, j+k} \neq p_{i+l+1, j+k}\}) \\
N_{d_1}(i, j) &= \sum_{k, l \in \{-1, 0\}} \mathcal{I}(\{p_{i+k, j+l} \neq p_{i+k+1, j+l+1}\}) \\
N_{d_2}(i, j) &= \sum_{\substack{k \in \{0, 1\} \\ l \in \{-1, 0\}}} \mathcal{I}(\{p_{i+k, j+l} \neq p_{i+k-1, j+l+1}\})
\end{aligned}$$

ただし、 $p_{i,j}$ は画素 (i, j) の値であり、 $\mathcal{I}(\cdot)$ は括弧内が真のとき 1、偽のときに 0 となる関数である。

連結数は、ブロック内の白黒の連結領域の数によって調べる。連結領域であるかどうかは、白黒ともに 4 連結で調べる。

可変得点の計算は以下の手順で行う。

1. ブロック内の画素の値がすべて白、もしくは、すべて黒ならば、可変得点は 0 とし、終了する。
2. $N_h = 0$ または $N_v = 0$ であれば、可変得点は 0 とし、終了する。それ以外のときには $s = 0.5$ とする。
3. $N_{d_1} = 0$ または $N_{d_2} = 0$ のとき、 $s = s - 0.125$ とする。 $N_{d_1} \neq 0$ かつ $N_{d_2} \neq 0$ のとき、 $n_p = \max(0, 3 - \min_{p \in \{h, v, d_1, d_2\}} N_p)$ を求め、 $s = s - 0.125n_p$ とする。
4. 中心画素の値を反転するとき、 N_h が変化しなければ $s = s + 0.125$ とし、変化すれば $s = s - 0.125$ とする。同様に、 N_v が変化しなければ $s = s + 0.125$ とし、変化すれば $s = s - 0.125$ とする。
5. 中心画素の値を反転するとき、連結数が変化すれば、 $s = s - 0.125$ とする。
6. s を可変得点とし、終了する。

図 2 は可変得点が 0.5 以上となるブロックの例である。可変得点が 0.5 以上のブロックでは中央の画素の値を変更しても、やはり可変得点が 0.5 以上であることが保証される。

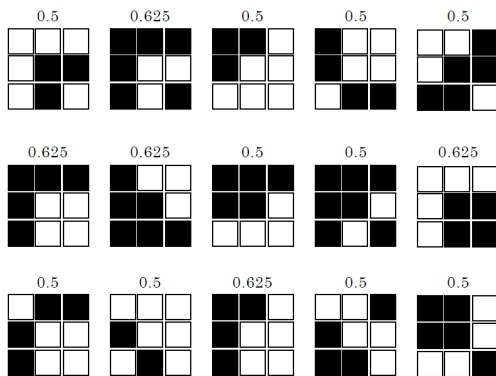


図 2 可変得点が 0.5 以上となるブロックの例

ロゴ画像から可変得点が 0.5 以上のブロックを検出し、0 を埋め込むときには中央画素に黒を、1 を埋め込むと

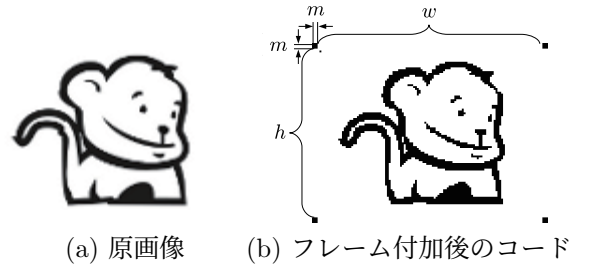
きには中央画素に白を埋め込む。情報を読み取る时候にも、画像上で可変得点が 0.5 以上のブロックを見つけ、中央画素が 0 か 1 かを見れば、情報を読み取ることができる。この手法では、情報読み込みのときに原画像を参照する必要がない。

3.2. 原画像復元のためのフレーム付加

カメラで観測する画像で情報読み込みを実現するためには、まず観測画像を原画像の座標系に変換する必要がある。本研究では、図 1 に示したように、画像の周囲に小さなマーカを付加することで原画像の座標系への復元を行う。

原画像には、3.1 で述べた可変得点によって情報埋め込みが行われる。このとき、原画像の向きを表現するために、画像の左上に方向検出マーカを付加する。方向検出マーカとその外側の白色のスペースによって、画像の四辺は 2 画素分ずつ外に膨らむ。このときの画像のサイズを $w \times h$ 画素とする。

さらに、この画像の四隅に縦横 m 画素の黒色矩形領域から構成されるフレームマーカを付加する。図 3 に例を示す。図 3(a) の原画像から構成される Be-code は、図 3(b) に示すようになる。



(a) 原画像 (b) フレーム付加後のコード
図 3 フレームの付加による Be-code 生成

原画像復元のためには、まず、観測した画像から Be-code のフレームマーカを検出する。ここで、観測画像は白黒の二値で表現されているものとして議論を進める。観測画像の二値化については、3.3 で詳しく述べる。観測画像中の黒色領域をラベリングし、連結領域に分割する。このとき、図 4 に示すように、それぞれの分割領域に外接する長方形のうち、面積が最小となるものを求める。この長方形は、rotating calipers 法によって求めることができる [6]。フレームマーカは黒色で塗りつぶされた矩形領域であるので、外接する長方形の内部は図 4(b) に示すように黒色領域の割合が高くなる。その他の一般領域では、図 4(b) に示すように黒色でない領域が多く含まれることになる。黒色である割合に閾値を設定して、フレームマーカを検出する。フレームマーカの一辺の大きさ m を大きくするほど閾値を高く設定できるので、一般領域との判別が容易となる。

原画像上の四隅のフレームマーカの外側の点 $(0, 0)$, $(w+2m, 0)$, $(w+2m, h+2m)$, $(0, h+2m)$ を (x_i, y_i) ($i = 0, 1, 2, 3$) と表すことにする。Be-code は平面上に貼られているので、観測画像から原画像への変換は 3×3 の平



(a) フレームマーカの 外接長方形
 (b) 一般領域の外接長方形

図4 フレームマーカの探索

面射影変換 P で記述することができる。観測画像上で四隅のフレームマーカの外側の点 (x'_i, y'_i) ($i = 0, 1, 2, 3$) を求めることができれば、原画像上の4点 (x_i, y_i) ($i = 0, 1, 2, 3$) との関係は P を用いて以下のように記述される。

$$P = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_0 & b_0 & 1 \end{pmatrix} \quad (1)$$

$$\forall i \ x'_i = \frac{a_1 x_i + b_1 y_i + c_1}{a_0 x_i + b_0 y_i + 1} \quad (2)$$

$$\forall i \ y'_i = \frac{a_2 x_i + b_2 y_i + c_2}{a_0 x_i + b_0 y_i + 1} \quad (3)$$

式 (2), (3) によって、8つの未知数に対して8つの方程式を立てることができ、 P の各要素の値を求めることができる。

w, h は、観測画像上でフレームマーカの大きさとエンブレムエリアの大きさの比から推定することができる。これによって、 w, h をあらかじめ規定することなく Be-code を読み取ることが可能になる。推定手順は以下の通りである。まず、仮の射影変換行列 P' を求めるために、任意の定数 w', h' を用いて (x_i, y_i) ($i = 0, 1, 2, 3$) を定義する。 P' は式 (2), (3) から求められる。 P' を用いることで、観測画像から Be-code が長方形に映るような画像を得ることができる。このときに、縦横比は正しくは得られないが、フレームマーカの大きさとエンブレムエリアの大きさの比を求めるためには問題とならない。 P' を用いて復元されたフレームマーカの大きさが $m'_x \times m'_y$ となるすると、 w と h の推定値 $\tilde{w}, \tilde{h}$ は以下の式で求められる。

$$\tilde{w} = \frac{m}{m'_x} w', \quad \tilde{h} = \frac{m}{m'_y} h' \quad (4)$$

このときに、 m が小さければ $\tilde{w}, \tilde{h}$ が1画素単位で正しく求まることは期待できない。そこで、原画像の大きさ w, h がそれぞれ N の定数倍であるという制約を設け、推定するときには $\tilde{w}$ と $\tilde{h}$ に最も近い N の定数倍とすることで推定誤りを避ける。今回の実装では $N = 10$ とした。このとき、 $\tilde{w}$ と $\tilde{h}$ を四捨五入することで原画像の大きさ w, h の推定値を得ることができる。 w, h が推定できれば、観測画像と原画像の間の平面射影変換行列を改めて求め、原画像を復元する。

3.3. 観測画像の適応的二値化

フレームマーカの検出および原画像の復元のためには、観測画像を二値化する必要がある。照明環境により、観測画像の濃淡はマーカ内でも変化する可能性があるため、位置に対して適応的に閾値を設定することが必要となる。

QRコードでは、コード内のマーカで区切られたブロックごとに二値化の閾値を設定している。これによって、局所的に陰影が生じる場合にも、正しく二値化を行うことができる。ただし、この処理は各ブロック内に白黒の領域が両方含まれることを前提に行われている。Be-code では、QRコードのように小さなブロックごとに適応的に閾値を決定しようとする、あるブロックでは白の画素あるいは黒の画素のみしか含まれないことが起こり、正しい閾値を得ることができない。そこで、図5に示すように観測画像を中心から上下左右で4分割し、それぞれの領域で適応的に二値化の閾値を決めることとする。Be-code が中央に配置されていれば、フレームマーカがそれぞれの分割領域で観測されることになるので、それぞれの領域が白黒両方の領域を含むことが保証できる。閾値の決定には、大津の閾値判定法 [7] を用いた。この判定法では、明るさのヒストグラムが双峰性を持つと仮定して、2つの山の境界となる値を二値化の閾値とする。

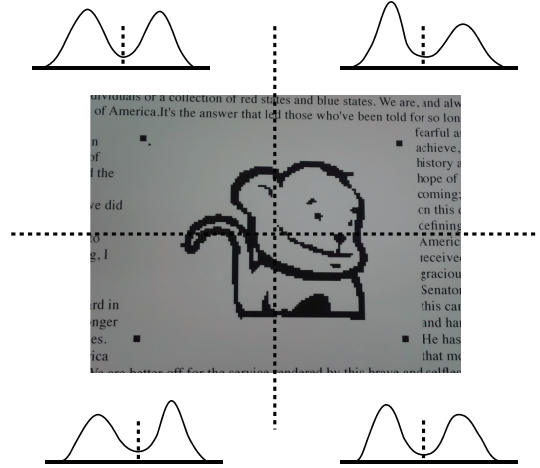


図5 画像分割による適応的二値化

3.4. グレースケール画像への情報埋め込み

前述の可変得点を用いた情報埋め込みと情報読み取りの手法は、二値画像を対象としていた。一般的に明暗の2色で構成されているように見えるロゴ画像でも、ロゴの縁の部分ではアンチエイリアシングが施されていることが多い。アンチエイリアシングは、エッジ付近の画素にグレースケールの値を割り当てることで実現される。アンチエイリアシングが施された画像を二値化すると、ジャギーが目立ち、デザイン性が損なわれることがある。この問題は、コード化の対象となる画像を二値化処理の妨げとならないようにグレースケール化することで解決できる。

アンチエイリアシングを含むグレースケール画像中の

注目画素の値を v 、二値化の閾値を g_{th} とする。 g_{th} はグレースケールの原画像に対して大津の閾値判定法を用いることで決定する。 v が g_{th} に近ければ、照明変化などにより二値化の結果が変わることがある。そこで、照明変化の影響を受けやすい範囲を黒側の基準値 $b_{th} = g_{th} - d_b$ と白側の基準値 $w_{th} = g_{th} + d_w$ で定義し、すべての画素がこの範囲内の値を取らないように以下の処理を行う(図6)。ただし、 d_b 、 d_w は照明変動の大きさを反映した閾値である。

1. $b_{th} < v < g_{th}$ であるとき、 $v = b_{th}$ とする
2. $g_{th} \leq v < w_{th}$ であるとき、 $v = w_{th}$ とする
3. $v \leq b_{th}, v \geq w_{th}$ であるとき、変更しない

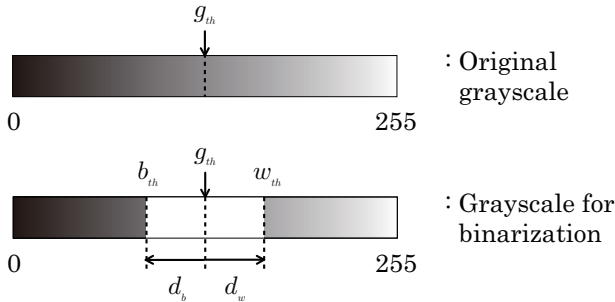


図6 二値化処理のためのグレースケール化

g_{th} は画像ごとに異なるので、 d_b と d_w を固定値とすると、白側や黒側のいずれかのレンジが極端に狭くなることがある。 d_b と d_w をそれぞれ 0 から g_{th} および g_{th} から 255 の幅を考慮して相対的に決定することで、この問題を避けることができる。幅の割合を決定するための係数を c ($0 \leq c \leq 1$) とすると、 d_b と d_w は以下の式で求められる。

$$d_b = c g_{th} \quad (5)$$

$$d_w = c (255 - g_{th}) \quad (6)$$

さらに、情報埋め込みの白黒の反転の際にも、以下のように画素値を変更することで、原画像からの変化を少なくすることができる。

1. $v \leq b_{th}$ を白に変更するときには、 $v = w_{th}$ とする
2. $v \geq w_{th}$ を黒に変更するときには、 $v = b_{th}$ とする

図7に二値化画像をコード化したものとグレースケール画像をコード化したものを示す。グレースケール画像をコード化したものでは、エイリアシングが軽減されていることがわかる。

3.5. リード・ソロモン符号を用いた誤り訂正と読み取り成否の判定

観測画像から原画像の復元を行うときには、画素の反転が起こることがある。そこで、Be-code に埋め込む情報には、QRコードにも採用されている誤り訂正符号の一種であるリード・ソロモン符号を採用する。リード・

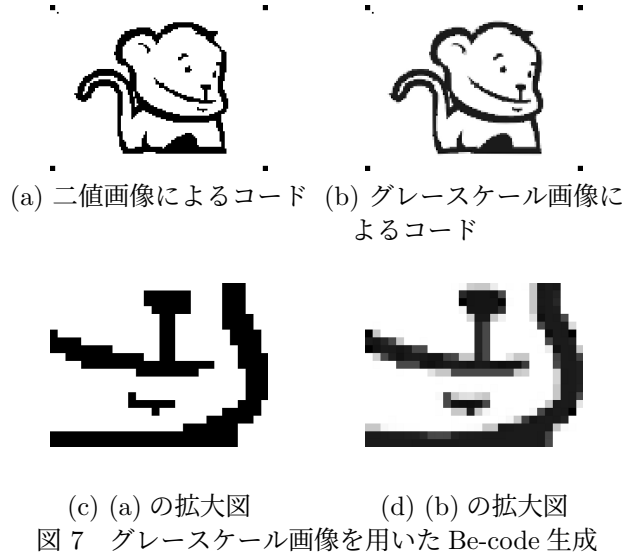


図7 グレースケール画像を用いた Be-code 生成

ソロモン符号を用いることで、少数の誤りに対しては訂正できるようになる。訂正能力を超えた誤りが起こった場合にも、読み取りに失敗したことが判定でき、読み取れるまで画像取得を続けることを促すことができる。

リード・ソロモン符号を用いた情報埋め込みでは、入力されたデータ符号を8ビット1シンボルとして区切り、有限体に基づいて生成される多項式によって、誤り訂正符号を求める。誤り訂正符号をデータ符号に付加したものを通信符号とし、画像に情報を埋め込む。情報の読み取りでは、通信符号に誤りを含んだ受信符号が取得される。受信符号に対し、まず誤りの検索を行い、受信符号に誤りが含まれているか、また、その誤りが訂正可能であるかを判断する。誤りが含まれており、さらにその誤りを訂正可能である場合には、誤り位置多項式によって誤り位置を算出し、訂正する。これにより、誤りのない元の通信符号を復号することができる。誤りが含まれているが、その誤りが訂正不可能である場合には、読み取りに失敗したと通知し、利用者に再度読み取りを促すこととする。

4. 実験

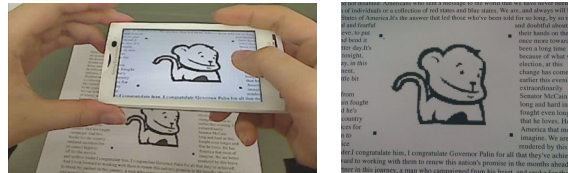
4.1. 観測画像からの情報読み取り

提案する Be-code の読み取り精度を検証するために実験を行った。撮影にはソニーエリクソン社製携帯端末 Xperia(SO-01B)を用いた。実験では、図8(a)および図8(b)、図8(c)に示す原画像から生成される Be-code を対象とし、一般的な室内環境で行った。フレームマークは縦横3画素とした。二値化処理を考慮したグレースケール化のための係数 c を $1/3, 1/2, 1$ としたものを対象とした。原画像と c を変えた Be-code を各100回撮影し、情報読み取りの精度を調べた。図9(a)は撮影の様子であり、図9(b)は実際の観測画像の例である。

c と読み取り成功率の関係を調べた。図8(a)、図8(b)、図8(c)に示した原画像から生成される Be-code を机の上に置き、100回ずつ自由に撮影を行ってもらった。成功率を表1に示す。

	原画像	$c = 1/3$	$c = 1/2$	$c = 1$
(a) サル				
(b) 手紙				
(c) 雪				

図 8 実験に用いた Be-code



(a) 実験環境 (b) 観測画像例
図 9 実験環境と観測画像例

表 1 画像 100 枚の成功率

c	1/3	1/2	1
サル	34	98	100
手紙	91	100	99
雪	51	100	100

$c = 1/3$ のときには、多数の観測画像で読み取りに失敗した。 $c = 1/2$ のときにはサルが 2 枚、 $c = 1$ のときには手紙で 1 枚のみで読み取りに失敗した。

読み取り結果を詳細に検証すると、 c を小さくしたときに読み取りができなかった観測画像では、うまく焦点が合っていないことが確かめられた。これは、エッジ付近にグレースケールの値を加えることで、エッジの高周波成分が失われ、カメラのオートフォーカス機能がうまく働かなかったためと考えられる。焦点が合っていないと、観測画像に焦点ぼけが生じ、正しい二値化画像を得ることができない。画像コードを接写して撮影すると焦点距離が短くなり、被写界深度が浅くなるために、結果として焦点ぼけが大きくなる。観測画像上を詳しく見る

と、焦点が合っていない領域があることが確かめられた。これによって、マーカ内部の一部の画素で白黒の反転が起こっていた。カメラを離れた上でズームして撮影することができれば、この問題は一部解決することができると思われる。

以上の結果から、 c を $1/2$ 以上に設定することで、高い読み取り精度を保證できるといえる。

一方で、Be-code にはリード・ソロモン符号が導入されているので、少数の誤りは訂正されており、また、訂正能力を超える誤りは検出することができる。訂正能力を超えた誤りが起こった場合には、ユーザに再撮影を促すようにシステムを設計している。

4.2. 読み取り精度とデザイン性の検証

3.4 で説明したように、情報の埋め込みに原画像のグレースケールの値を利用している。Be-code のデザイン性の確保のためには、グレースケール化のための係数 c は小さい方がよい。一方で、二値化による読み取り処理を頑健にするためには、 c は大きい方がよい。これらはトレードオフの関係にある。そこで、デザイン性を確保しつつ読み取りも頑健に行える c の値について、実験によって検証した。

図 8(a)、図 8(b)、図 8(c) に示すサル、手紙、結晶の原画像に対して、 $c = 1/3, 1/2, 1$ として Be-code を生成した。これらと $c = 0$ として生成された Be-code との比較を行い、PSNR を求めたものを表 2 に示す。 $c = 0$ と

するときには、原画像と Be-code との画素値の差は高々 1 となり、原画像のグレースケールの値はほぼ保存されることになる。

表 2 $c = 0$ として生成される Be-code に対する c を変化した Be-code の PSNR(dB)

c	1/3	1/2	1
サル	28.8	26.2	19.3
手紙	28.3	26.9	21.5
雪	30.6	28.8	23.1

PSNR の値が大きいほど、比較される 2 つの画像に近いといえる。表 2 から、 c の値を小さくするほど原画像に近い Be-code が得られることがわかる。また、同じ c の値を設定したときにも、ロゴ画像によって PSNR の値に差があることもわかる。サルの画像から生成される Be-code では、 c を大きくすると他の画像よりも PSNR が小さくなった。

4.1 では、 $c = 1/3$ としたときと $c = 1/2$ としたときとで読み取り精度に差があることを述べた。読み取りのことを考慮すると、精度が低下しにくい $c = 1/2$ に留めておくべきである。具体的に c をどのような値にするかは、期待するデザイン性と必要な読み取り精度を考慮して、用途によって決定することが必要となる。

5. まとめ

ロゴ画像などの単純な画像に対して、デザイン性を確保したまま情報埋め込みを実現する Be-code を提案した。デザイン性を確保するために、グレースケール画像に対する情報埋め込みの手法を提案した。実験では、携帯端末による読み取り精度と確保されるデザイン性の関係について検証した。

今後の課題として、焦点を合わせやすいパターンの開発を目指す。実験では、パターンに高周波がなくなることによりオートフォーカスがうまく働かないことによる読み取りの失敗が見られた。これは画像全体で高周波成分が失われることが問題であり、オートフォーカスのために Be-code の一部に高周波成分を残しておくことで解決できると考えている。

また、今回の実験は一般的な室内環境で行ったが、屋外環境や特殊な光源環境での読み取り精度の検証はまだ行っていない。観測画像にハイライトがある場合や広範囲にわたる影が写り込む場合に、適切な二値画像を得ることができないことが予想される。適応的二値化の手法を改良することでこの問題に対処したい。また、複数枚の観測画像を使って、照明条件をキャンセルして正しい原画像を得ることも考えられる。

加えて、カラーのロゴ画像に対する埋め込みを実装する。これは、提案手法では、グレースケール画像への情報埋め込みを行っているが、実際のロゴ画像はカラー画像が多数であるためである。カラーのロゴ画像を、HSV 表色系に変換し、明度情報を抽出する。埋め込みは、彩度と色相の成分を保ったまま、明度成分のみを変化させる。明度成分に対して、3.4 で示した情報埋め込みを行

うことで実現できると考える。

謝辞

査読者より有益な助言、コメントを頂いた。ここに感謝の意を表する。

文献

- [1] ColorZip Media, Inc., <http://www.colorzip.com>, ColorCode, , 2000.
- [2] 野田嗣男, 師尾潤, 千葉広隆, “印刷型ステガノグラフィ技術,” 富士通ジャーナル, vol.57, no.3, pp.320–324, 2006.
- [3] 三宅哲平, 栗山繁, “図形輪郭線の明度変化を用いた情報埋め込み,” 電子情報通信学会論文誌, vol.J93-A, no.2, pp.73–81, 2010.
- [4] A. Pramila, A. Keskinarkaus, and T. Seppanen, “Reading watermarks from printed binary images with a camera phone,” IWDW 2009, vol.5703, pp.227–240, 2009.
- [5] M. Wu, and B. Liu, “Data hiding in binary image for authentication and annotation,” IEEE Transactions on Multimedia, vol.6, no.4, pp.528–538, 2004.
- [6] G.T. Toussaint, “Solving geometric problems with the rotating calipers,” IEEE MELECON, 1983.
- [7] 大津展之, “判別および最小 2 乗規準に基づく自動しきい値選定法,” 電子通信学会論文誌, vol.63, no.4, pp.349–356, 1980.