

カラー化イメージ付き二次元コードの読み取り特性の検討

サムレットウィット ダムリ[†] 若原 俊彦[‡]

[†] 福岡工業大学大学院 情報通信工学専攻 〒811-0295 福岡県福岡市東区和白東 3-30-1

E-mail: [†] mgm11007@bene.fit.ac.jp, [‡] wakahara@fit.ac.jp

あらまし 現在、インターネットや携帯電話の普及に伴い QR コード*が世界中に広まっており、各種用途に QR コードが使われている。二次元コードの代表である QR コードは白黒 2 値のドット表示であるためデザイン性が良くないので、ドットをカラー化して見栄えを良くする方法が用いられる。しかし、ドットの色を変更したり見栄えを良くするためにイメージを重ねたりするとイメージ損傷になり QR コードの読み取り特性に影響を及ぼす。また、紙に印刷したりディスプレイに表示した QR コードの場合には、カメラで撮影する際の照明条件および表示条件によって、QR コードの読み取り特性が異なる。

本論文では、通常の QR コードにドット絵などのイメージを重ねし白黒 2 値からカラー化してイメージ付き二次元コードを構成し、QR リーダとして現状のスマートフォンなどに用いられている QR デコーダをそのまま用いた場合に、カラー化する際のパラメータによってどの程度正常に認識できるのか実験により読み取り特性を評価し、その結果を報告する。

キーワード 二次元コード, QR コード, 誤り訂正能力, カラー化, イメージ

Reading Characteristics of Two-Dimensional Color Code with Images

Damri SAMRETWIT[†] Toshihiko WAKAHARA[‡]

[†] Fukuoka Institute of Technology, Graduate School of Information and Communication Engineering

3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295 Japan

E-mail: [†] mgm11007@bene.fit.ac.jp, [‡] wakahara@fit.ac.jp

Abstract In recent years, QR code is widely used in the accessing to the Internet by cellular phone. As the design of this code is a simply black and white dotted symbol, it is very simple and insipid. In order to improve this weakness, colors dotted symbol are developed.

In this paper, we have studied the structure of the QR code, the way of multiplexing the dotted images and colorization method. In order to improve the QR reading characteristics, the several image multiplexing methods are proposed. The experiments are tried and the reading characteristics of the two-dimensional colorized code are measured by using the existing common QR decoder. The colorization method of the two-dimensional codes is evaluated.

Keyword Two-dimensional code, QR code, Error correction capability, Colorization, Image

1. はじめに

QR コード[1]は、二次元コード[2]の一種であり、「リーダにとって読み取り易いコード」となることを目的にデンソーウェーブ（開発当時は株式会社デンソーの一部門）が開発し、1994 年に発表したものである。QR コードとは、縦、横二次元方向に白と黒の明暗パターン情報を持つことで、一方向だけに情報を持っている

バーコードに対し、記録できる情報量を飛躍的に増加させたコードである。

本論文では、この QR コードの冗長部分にイメージなどの画像を各種重ね合わせ、見栄えを良くするためにカラー化したカラー化イメージ付き二次元コードの読み取り特性を実験的に検討した結果を示したものである。

*QR コードは（株）デンソーウェーブの登録商標である。

This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.

以下、2 章では QR コードの概要について述べ、3 章では QR コードに各種イメージを重ね合わせた二次元コードについて示す。4 章ではカラー化の手法とカラー化イメージ付き二次元コードの概要を説明する。5 章では、このカラー化イメージ付き二次元コードを通常の QR デコーダを用いて読み取り実験を行った結果を示す。最後に 6 章でまとめを述べる。

2. QR コードの概要

QR コードの主な仕様を表 1 に示す。このシンボルのバージョン（型番）には、同表に示すように、1 型～40 型までの 40 種類がある。1 型は 21×21 モジュール、2 型は 25×25 モジュール、最大 40 型は 177×177 モジュールというように、型番が一つ上がる度に一辺につき 4 モジュールずつ増加する。

表 1 QR コードの主な仕様

	仕様	
誤り訂正符号	RS コード (Reed-Solomon)	データコード語
		誤り訂正コード語
	BCH コード	形式情報
		型番情報
モード	数字	10 ビット/数字
	英数字	11 ビット/文字
	8 ビットバイト	バイナリ 8
	漢字	13 ビット/文字
バージョン (型番)	1	21×21 モジュール
	2	25×25 モジュール
	.	.
	40	177×177 モジュール
誤り訂正レベル	L	約 7%
	M	約 15%
	Q	約 25%
	H	約 30%
位置検出パターン	1:1:3:1:1	3 同心の正方形: 7x7, 5x5, 3x3 モジュール
位置合せパターン	1:1:1:1:1	3 同心の正方形: 5x5, 3x3, 1x1 モジュール

QR コードには、汚れや破損などがあっても正しく読み取れるように誤り訂正符号を用いており、データコード語に対して修正するために付けられる誤り訂正コード語の割合として、下記の 4 レベルがある。

- レベル L：コード語の約 7%が復元可能
- レベル M：コード語の約 15%が復元可能
- レベル Q：コード語の約 25%が復元可能
- レベル H：コード語の約 30%が復元可能

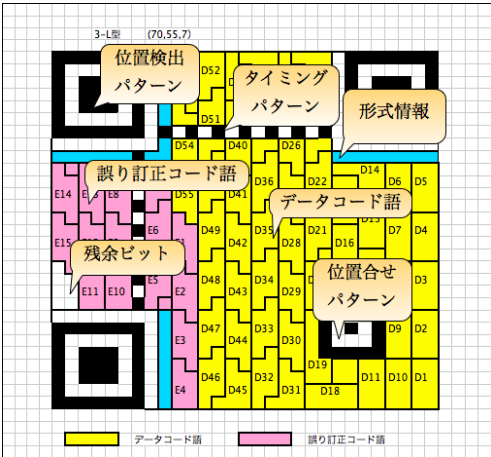


図 1 QR コードの例（バージョン 3）

さらに、QR コードを読み取り易くするために行う処理としてマスク処理があり、そのパターンは 8 種類用意されている。その中で“明”モジュール数と“暗”のモジュール数を均一化し、画像の高速読み取り処理の障害となるパターンの発生が抑えられるマスクを採用することになっている。マスク処理では、符号化領域のビットパターンとマスク処理パターンを XOR（排他的論理和）する。

図 1 は、白と黒の正方形の格子状に配置されているバージョン 3 の QR コードの例を示す。QR コードには、すべての方向の高速読み取りを有効にするために、3 つのコーナーに配置されたシンボルの位置を示すための 3 つのファインダーパターン（位置検出パターン）がある。

3. 各種二次元コードの概要

近年、QR コードのデザイン性を改善するため高度な二次元コード（QR コードの仕様に準拠していない）の新しいタイプが開発されており、特にイメージや画像を重ね合わせたものが多いが、携帯電話などの QR リーダ（QR デコーダ）で簡単に読みとれる。例えば、QRForest[3]は、図 2 に示すように、ユーザが位置形状、データ領域の形状、配置形状と重ね合わせた画像を変更することができる二次元コードである。



図 2 イメージを重ね合わせた QRForest

その他の二次元コードの新しいタイプとして、メロディーや画像を保存することができる着信メロディ付きの QR コード [14] やイメージを合成できる QR-JAM[4], DesignQR[5]などがあり図 3 と図 4 に示す。



図 3 QR-JAM



図 4 DesignQR

さらに誤り訂正用の RS(Reed-Solomon)符号を非組織符号化法により構成した新しいイメージ付きの二次元コードが開発されており、図 5 に示すように画像は任意の位置に多重化される[6]。また、アニメーション QR コード[7]は、図 6 に示すように、最適化アルゴリズムを使用して設計されている。



図 5 Image displayable
DesignQR code



図 6 Animated
QR code

4. イメージ付き二次元コードの概要

4.1. QR コードの冗長部へのイメージ重ね合わせ

この二次元コードは、図 7 に示すように Image Multiplexing QR Editor[8] (以降 QREditor と表示) のプログラムを使用して QR コードから作成する。このプログラムでは、QR コードのマスクの代わりに任意のイメージ情報を挿入できる特徴がある。イメージ情報を挿入するため、符号化すべき情報に必要なバージョンよりも大きな型番を選択すると、中央部分に埋め草コード語のエリアを確保することが出来る。この QR コードの冗長エリアに自由にドット絵などイメージをデザインする事が可能になる。図 7,8 には、ドット絵で「NORTH」の文字を記述しており、通常の QR リーダで読み取ることが出来る。

4.2. 「暗い色の四角」と「明るい色の四角」のルール

QR コードにイメージ損傷を与えた二次元コードを作成しても誤り訂正用の RS 符号に影響させない方法もある。例えば、萩原：“デザイン二次元コード”[9]によると、“符号化と復号では要となる処理に差異がある”を利用する方法である。この文献によれば、“復号器は即応符号を構成する小正方形を「暗い色の四角」と「明るい色の四角」の二つに大別して認識する。

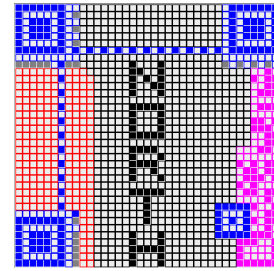


図 7 QR Editor によるドット絵挿入

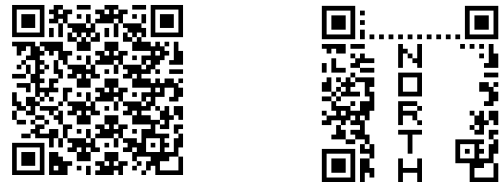


図 8 QR Editor で作成した二次元コード

黒い四角、群青色の四角、深緑色の四角などはいずれも「暗い色の四角」として認識される。同様に、白い四角、淡いピンク色の四角、黄色の四角などはいずれも「明るい色の四角」として認識される。”と述べられている。図 9 に示す例では、それぞれ赤と青は「暗い色の四角」に相当するので、これらの読みとりが可能である。

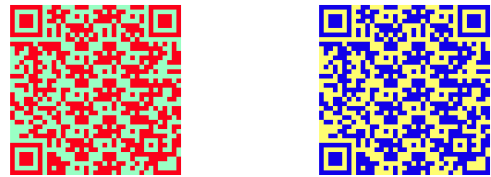


図 9 「暗い色の四角」の例（赤と青）

4.3. カラー化イメージ付き二次元コードの作成方法

“二次元コードのカラー化に対する読み取り特性の検討”[10]によると、カラー化イメージ付き二次元コードの作成にあたっては、まずQREditorを用いてQRコードを作成する。このプログラムを使用すれば、QRコードのマスクの代わりに任意のイメージ情報を挿入する事ができる。これによりQRコードの冗長エリアに自由にドット絵などをデザインし、イメージ付き二次元コードを作成出来る。次に、フリーの画像編集ソフトであるGIMPを用いて白黒ドットの色を変更して、白を“明るい色”に、黒を“暗い色”に変更し、カラー化イメージ付き二次元コードを作成する。前述のように中央の冗長エリアがある二次元コードを作成してその冗長エリアにイメージをデザインする方法を用いる。この方法では、誤り訂正レベルがQやHのように大きい場合には誤り訂正コード語部分が大きくなり、冗長エリアが小さくなるので、誤り訂正レベルが小さいLなどの二次元コードを使用するのが良い。更に、図10に示す

ように、バージョン5以上のQRコードには中央の冗長エリアに位置合わせパターンが生ずるので使用すべきではないが、バージョン5以上になるとRSブロック数が2以上になり、ブロックが分割されるので影響は少なくなり、バージョン5レベルLのQRコードを使用するのが良い。

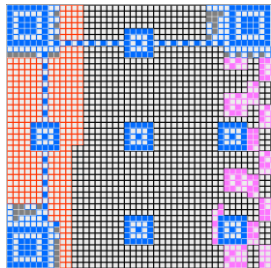


図 10 バージョン 5 レベル L の QR コード

QREditorで作成した図 10 の QR コードを人間の目で見ると冗長エリアの中は、白と解釈出来てデータがないように見える。しかし、実は空白と見えていてもダミーデータが入っている。実際には、図 11 に示すように冗長エリアの情報を見ているのは、そのデータとマスクの値の排他的論理和(Exclusive OR)をとった値であり、白い四角に見える。このデータとマスクを排他的論理和した後にデザインした絵が重畳されるため、マスクの処理と排他的論理和の処理の影響を受け、空白であってもドットのデザイン次第でデータが変わる。

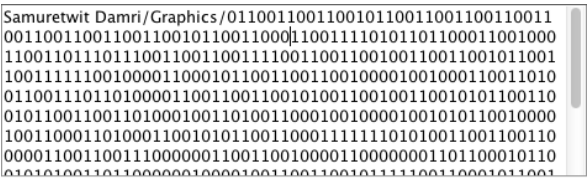


図 11 二次元コードの冗長エリアの中のデータ

次は、カラー化イメージ付き二次元コードを作成する方法について述べる。まず、図 12 および図 13 に示すようにイメージ付き二次元コードを作成する。“暗い色”の四角と“明るい色”の四角のルールに基づいて、作成した二次元コードのドットの色を自由に変えてカラー化イメージ付き二次元コードを作成する。

フリーの画像編集ソフトである GIMP を用いて白黒ドットの色を変え、白を“明るい色”に、黒を“暗い色”に変更してカラー化すると、カラー化イメージ付き二次元コードが完成する。

ここで QREditor を用いてドット絵のイメージを描くのに多少時間がかかるが、二次元コードの中央の上下方向の冗長エリア上に自由にデザインし、さらに GIMP を用いて暗い色と明るい色を選んで編集すれば誤り訂正の影響を受ける事はない。また、この方法で作成した二次元コードは見栄えも良くなる。

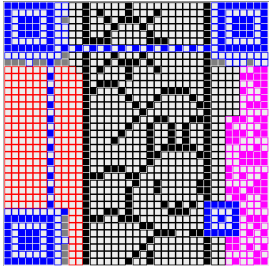


図 12 イメージ付き二次元コード



図 13 白黒→カラーイメージ付き二次元コード

しかし、残された問題は既存の QR リーダ（デコーダ）がどこまで読み取り可能かである。すなわち、既存の QR デコーダがどのような色を“暗”と判定し、どのような色を“明”と判定するのか不明である。このため、GIMP でカラー化する際に、色相、彩度および明度の 3 属性を変化させ、読み取り実験を行って解明することとした。なお、QR デコーダとしては、スマートフォンに用いられているソフトウェアをそのまま用いている。

5. カラーの認識の確認実験

5.1. 二次元コードの読み取り実験の概要

QR コードにイメージを重畳してカラー化イメージ付き二次元コードを各種作成し、その読み取り特性を評価する。実験条件を表 2 に示す。

表 2 二次元コード読み取り実験の条件

項目		諸元
QR コ ー ド	符号化データ	英字 15 文字
	型番(バージョン)	5
	誤り訂正レベル	H の 1 レベル
QR コード作成ソフト		QR Editor[8]
表示装置		MacBook Air(11 インチ)
環境条件		通常照明(蛍光灯)
環境モニタリング条件		80%輝度
QR 読み取り装置		iPhone 4
編集するプログラム		Gimp 2.8.0[11]
読み取るプログラム		i-nigma[12], QRReader[13]

5.2. カラー化二次元コードの読み取り実験の概要

この実験ではイメージの偏執ソフト GIMP2.8.0 を利用して、色のドットの RGB の値を変更しこの色を『明』

や『暗』の部分に入れ替えて読み取り特性を評価する。
図 14 は、GIMP の色の偏執の画面である。

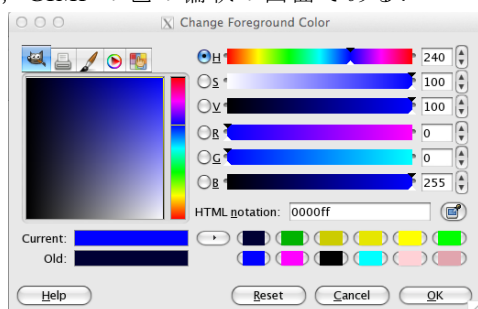


図 14 GIMP の色編集の画面

前述したように、既存の QR デコーダがどのような色を“暗”と判定し、どのような色を“明”と判定するのか明らかになっていない。このため、GIMP でカラー化の際に、色相、彩度および明度の 3 属性を変化させ、読み取り実験を行った。QR デコーダとしては、スマートフォン (iphone4 の QRReader for iPhone) を用いた。色の 3 属性の変化範囲としては、図 14 に示したように色相の値を 0 の赤から 360 の赤まで変化させると同時に彩度も 0 から 100 まで変化させて読み取りが可能か記録した結果である。なお、用いた符号データは英字 15 文字、QR コードの型番は 5、誤り訂正レベルは H の 30% で PC (MacBookAir) 上に表示させた。図 15 は彩度を変化させ“暗”ドットを入れ替えた例である。



彩度の 40 彩度の 60 彩度の 90

図 15 “暗”の部分を入れ替えた例

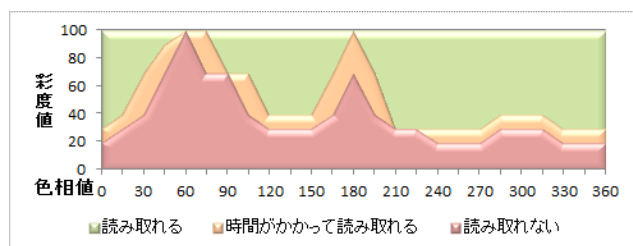


図 16 彩度と色相を変化させた時の読み取り特性

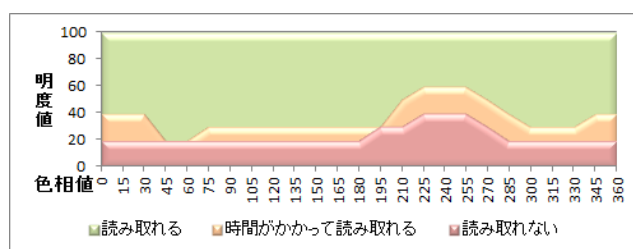


図 17 明度と色相を変化させた時の読み取り特性

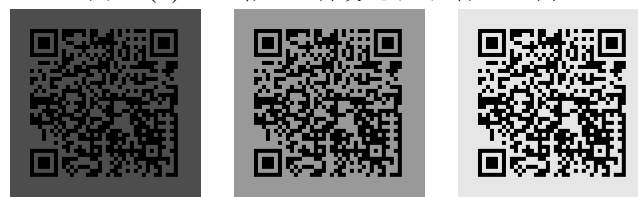
図 16 で色相値を変化すれば色が変わり、彩度値が小さいほど色が明るくなって読み取り難くなる。色相の値が 60 の場合は黄色に該当し、明度が高いので彩度値を変化しても読み取れない。色相の値が 180 の色はシアンであり明度も高いので読み取れても時間を要する。このように明るい色には注意が必要である。図 17 で明度値が低いほど色が暗く読み取り難くなることがわかる。色相の値が 0 と 360 の色は赤であり明度が高いので明度値を減らして、読み取れない事が大きい。色相の値が 240 の色は青であり赤よりも暗いので読み取る範囲がさらに大きい。つまり赤と青はかなり明度が低いので明の部分に入れ替えると読み取り難くなる。

このため、RGB から見ると、それぞれの明るさが持っているもので、それらの色の明るさを検討する。色を認識する場合、「暗い色の四角」と「明るい色の四角」のルールに従って色の濃さを判定すると考えられるためカラーの RGB 値をモノクロに変換してその輝度値で判定することとし、平均値を求める。実験では、カラーをモノクロ化して、その RGB 値をグレー化し、“暗”の場合の明るさを黒と認識し、“明”の明るさを白と認識する。図 18(1)は、グレー化して“暗”のドットを明度値 30, 60, 90 のドットに入れ替えて作成した二次元コードの例である。一方、図 18(2)はグレー化して“明”のドットを明度値 30, 60, 90 のドットに入れ替えて作成した二次元コードの例である。



明度値 30 明度値 60 明度値 90

図 18(1) “暗”の部分を入れ替えた例



明度値 30 明度値 60 明度値 90

図 18(2) “明”の部分を入れ替えた例

この読み取った結果で、RGB のモノクロ輝度値が 200 以上の値であれば完全に“明”と認識され、160 以上の場合はほぼ“明”と認識された。一方、RGB のモノクロ平均値が 50 以下の場合は完全に“暗”と識別され、80 以下の場合はほぼ“暗”と認識され、80 から 160 の間は“明”または“暗”と認識された。

RBG のモノクロ輝度値としては、下記の(1)式を計算して求める。

$$X = 0.294R + 0.615G + 0.091B \quad (1)$$

$$0 < R, G, B < 255$$

$$R = 170, G = 40, B = 0$$

$$0.294 \times 170 + 0.615 \times 40 + 0.091 \times 0 = 53.8 \quad \text{: 暗}$$

$$R = 0, G = 255, B = 255$$

$$0.294 \times 0 + 0.615 \times 255 + 0.091 \times 255 = 181.05 \quad \text{: 明}$$

$$R = 0, G = 255, B = 0$$

$$0.294 \times 0 + 0.615 \times 255 + 0.091 \times 0 = 158.1 \quad \text{: 両方}$$

図 19 (1)式によるモノクロ輝度値の計算の例

R, G, B の値は, GIMP の RGB の 0 から 255 までの値であり, X は出力の明るさである. この数式で RGB を入力して求めた X の値が 160 以上の場合は“明”と認識され, 80 以下の値の場合は“暗”と認識され, X が 80 と 160 の間の場合は“明”または“暗”と認識される. 図 19 は, モノクロ平均値の計算の例である.

色の明るさが分かりやすいように RGB の値から色相の値に変換し, 輝度の明るさとの関係を求めて図 20 に示す.

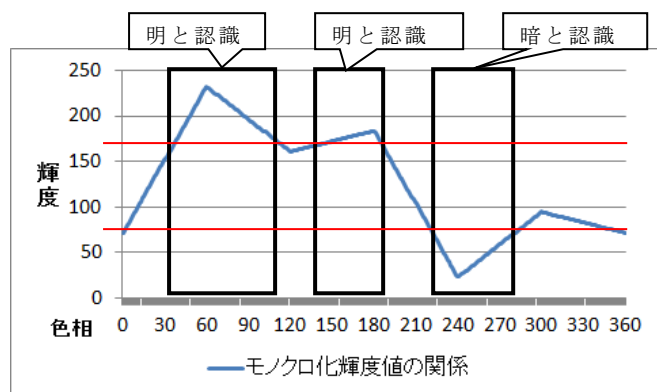


図 20 色相変化によるモノクロ化輝度値と輝度の関係

色相の値が 60 の場合は RGB の赤と緑色が混色した黄色に相当し, 最も明るい. 一方, 色相の値が 180 の場合は RGB の緑色と青を混色したシアンに相当し, 次に明るい. 最も暗いのは色相の値が 240 の青色であり, 次は色相の値が 0 と 360 の赤である. 80 と 160 の間の値である 158 の緑色や 94 のピンク色は, 周囲の環境により“明”または“暗”と認識される.

6. まとめ

以上, カラー化イメージ付き二次元コードの構成法について述べるとともに, カラー化した場合の既存の QR デコーダの読み取り特性を実験により示した.

今回行った実験により, カラー化イメージ付き二次元コードは, QR コードの冗長エリアにドット絵などイメージをデザインしてカラー化したものであり, 既存の QR デコーダをそのまま用いても, QR コードの白ドットを明るい色に黒ドットを暗い色に変換する限り,

このカラー化イメージ付き二次元コードを読み取り可能であることがわかった. また, RGB の値からモノクロ輝度値に変換して実験することにより, 以下に示すような特徴を有することが明らかとなった.

- (1) カラー化イメージ付き二次元コードは, 誤り訂正の影響を受ける事なく認識でき, 白黒からカラー化することにより見栄えが良くなる.
- (2) RGB からモノクロ化する際の数式は, 輝度信号を求める式と同じであり, この値が 160 以上の場合は“明”と認識され, 80 以下の場合は“暗”に認識され, これらの中間値の場合は周囲の色の値により“明”または“暗”に認識される.
- (3) 色相の違いにより, 黄色などはかなり“明”と認識され, 青色などはかなり“暗”と認識されるさらに, ピンク色をはじめ, “明”と“暗”の両方に認識される色もある.

文 献

- [1] ISO/IEC 18004:2000 Information Technology: Automatic Identification and Data Capture Techniques (Barcode Symbology) QR Code (MOD), June 2000
- [2] 二次元コードシンボル QR コード 基本仕様(JIS X0510) 日本規格協会 2004 年
- [3] QRForest, <http://qrforest.com/>.
- [4] QR-JAM, <http://staff.aist.go.jp/hagiwara.hagiwara/qjam/index.html>
- [5] DesignQR, <http://d-qr.net/index.htm>
- [6] 藤田和謙, 栗林稔, 森井昌克, “QR コードへの画像埋め込みに関する検討と提案”, 信学技報 LOIS2010-51, pp.39-44, Jan.2011
- [7] 小野智司, 森永健介, 中山茂: “最適化アルゴリズムを用いたアニメーション QR コードの作成”, 芸術科学会論文誌, Vol.8, No.1, pp.25-34 (2009)
- [8] 若原俊彦, 山元規靖, 越智祐樹; “QR コードの画像重畳評価法の検討”, 信学技報, vol. 110, no. 450, LOIS2010-65, pp. 1-5 (2011 年 3 月)
- [9] 萩原 学, “デザイン二次元コード”, 電子情報通信学会誌 Vol.94, No.4, PP.341-343 2011
- [10] Samretwit Damri, 若原俊彦, “二次元コードのカラー化に対する読み取り特性の検討”, 画像電子学会第 40 回年次大会, セッション S3-1 (2012 年 6 月)
- [11] GIMP <http://www.gimp.org/>
- [12] i-nigma, <http://www.i-nigma.com/i-nigmahp.html>
- [13] QRReader <http://itunes.apple.com/us/app/qr-reader-for-iphone/id368494609?mt=8>
- [14] QR コードの着信メロディ <http://qr.que1.jp/qmelo.php>