

# イメージ付きカラー化二次元コードの構成と読み取り特性の研究

Samretwit Damri

## 1. はじめに

QRコード[1]は二次元コード[2]の一種であり、Quick Responseコードの略である。名前の通り高速で読み取れ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）が開発したものである。QRコードは、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、一次元のバーコードよりも記録できる情報量を飛躍的に増加させたコードである。

## 2. 研究の概要

本研究では、通常のQRコードにドット絵などのイメージを重畳し白黒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するカラー化二次元コードエディタ（カラーQREditor）を開発する。また、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する。

## 3. ドット絵付き二次元コードのカラー化

カラー化ドット絵付き二次元コードの作成にあたり、以下の条件を考慮して二次元コードエディタを構成する。

### 3.1 ドット絵を挿入する領域に関するルール

QRコードでは、形式情報及び型番情報を除くシンボルの符号化領域は、データを符号化したときの0と1の組み合わせで右下部から各ドットの上方向に配置され上端に達すると1ドット左にずれて下方向にデータが配置され、これを順次繰り返し、さらに左側から順に誤り訂正（RS：ReedSolomon）コード語を配置してQRコードのデータが配置される。ここで型番を大きめに選ぶと冗長性が増加し、実データの入っていない冗長部が左側の誤り訂正コード部を除いた中央部に埋め草コード部の領域となる。この埋め草コード部は、埋め込むデータを保持したまま自由に操作することが可能である[3]。

### 3.2 “暗”ドットと”明”ドットの識別ルール

萩原の文献[4]によると、「二次元コードの符号化と復号では要となる処理に差異がある」であり、これによれば、「復号器は即応符号を構成するドットとして“暗い色”は“暗”ドットと認識され、“明るい色”は“明”ドットに認識される」。以下、これを「暗い色の四角」と「明るい色の四角」のルールと言う。従って、各ドットのカラーの輝度を算出し、輝度が基準値以下であれば“暗”ドット、基準値以上であれば“明”と認識される。既存のQRデコーダをそのまま用いて読み取り実験を行った結果[4]から、輝度値が80以下では“暗”、160以上では“明”と判定され、これらの値を参考にする。

### 3.3 ドット絵付きカラー化二次元コードエディタ

文献[5]で開発された白黒QRエディタを元にカラー化するため、前節に示したカラーの輝度をHSB色空間によって指定可能なウインドウを設けカラー変換することとした。図1にこのインタフェースを示す。実際にはカラーの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、“暗”と“明”に判定している。

## 4. 実験とその結果

### 4.1 カラー化ドット絵付き二次元コードの作成実験

QRコードにドット絵を重畳してカラー化ドット絵付き

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。

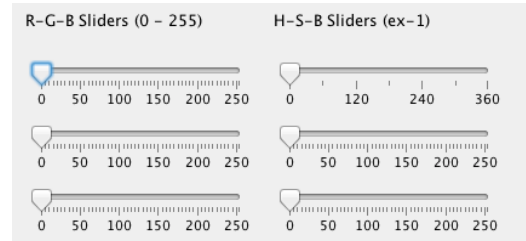


図1 カラー設定用インタフェース

表1 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	L, M, Q, H の 4 レベル
QRコード作成ソフト		QR Editor[6]
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
編集プログラム		Gimp 2.8.0[7]
読み取りプログラム		i-nigma[8], QRReader[9]
		Best Barcode Reader[10]

### 4.2 カラー化ドット絵付き二次元コードの作成実験

次に、カラーのドット絵を重畳するカラー化ドット絵付き二次元コードの作成方法について述べる。まず、図2に示すようにドット絵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イラスト二次元コードを作成できる。

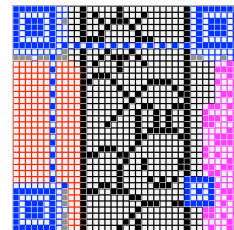


図2 ドット絵付き二次元コード



図3 ドット絵付き二次元コードのカラー化

二次元コードの型番（バージョン）を5以上に選ぶ。これは、誤り訂正のRSブロックは型番が3以上でブロックが複数に分離されバラバラになって配置され、さらに型番が7以上になると位置合せパターンが複数現れて邪魔することになるので、二次元コードのバージョンを5、大き

い冗長エリアが得られる誤り訂正レベルの最も低いLを選ぶのが良い。この手法により、ドット絵のデザインを描くのに多少時間がかかっても二次元コードの上に自由にデザインできる上に「暗い色の四角」と「明るい色の四角」のルールを利用して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出力は見栄えも良い。図3にその例を示す。

#### 4.2 モノクロ輝度値の変換式による評価

各ドットのカラーをRGBに分解し、それぞれ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値をグレースケール化し、“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4は、グレースケール化して“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5はグレースケール化して“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。



図4 “暗”の部分を入れ替えた例



図5 “明”の部分を入れ替えた例

これらの二次元コードを既存のQRリーダーで読み取った結果、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”（時々読み取れなかったりして、読み取り時間を要した）と認識された。一方、RGBのモノクロ平均値が50以下の場合には完全に“暗”と識別され、80以下の場合にはほぼ“暗”（時々読み取れなかったりして、読み取り時間を要した）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。

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

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

#### 4.3 カラーQREditorプロトタイプの作成

実験から得られた色の輝度値や計算結果を参考にしてカラー化ドット絵付き二次元コードエディタ（カラーQREditor）のプロトタイプを試作し、読み取り実験を行った。このエディタを用いて作成した例を図6、図7に示す、これらはプロトタイプを用いて描いたドット絵をカラー化し二次元コードに変換したものである。カラー値は、スライドバーのRGB値やHSB値により設定する。画面の下には選択した色のRGB値を表示している。さらにその選択した色の明るさを示している。カラー調整を

RGBのバーで自由に設定できるが、輝度値の値によっては“明”か“暗”の設定を判断できない色もあるので、確実に選

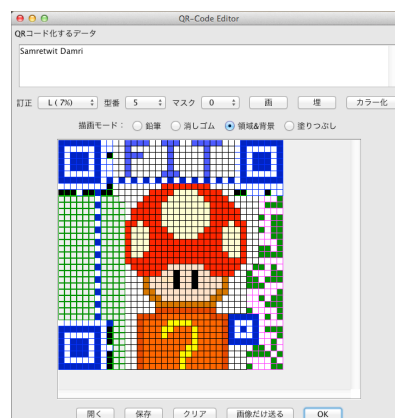


図6 ドット絵編集の例



図7 作成したカラードット絵付き二次元コード  
扱できる色として50%以下の輝度値は“暗”，90%以上の輝度値は“明”と設定した。これらの中間の値では暗と明のいずれにも判定され、読み取りが難しかった。

## 5. まとめ

今回、白黒QREditorを改造してカラー化し、カラードット絵付き二次元コードエディタ（カラーQREditor）を開発した。これを用いた実験により、既存のQRデコーダをそのまま用いて読み取り特性を評価した。この結果、カラーをモノクロ輝度値に変換した値が200以上であれば“明”と、50以下であれば“暗”と判定出来、「暗い色の四角」と「明るい色の四角」のルール通りの判定が出来ることを確認した。また、本エディタには静止画などのカラーイメージも重畳させることが出来、文字通りイメージ付きカラー化二次元コードを開発出来た。

## 参考文献

- [1] QRCode <http://www.qrcode.com/aboutqr.html>
- [2] 二次元コードシンボル QR コード 基本仕様 (JISX0510) 日本規格協会 2004 年
- [3] 若原俊彦、山元規靖、越智裕樹：“2 次元バーコードの情報多重化手法の一検討” 信学技報, vol. 109, no. 205, LOIS2009-26 (2009 年 9 月)
- [4] 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 年
- [5] サムレットウィット ダムリ, 若原俊彦：“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”, 信学技報, vol.112, no.308, MOMUC2012-34, pp.13-18 (2012 年 11 月)
- [6] 越智祐樹：“QR コードへの画像の挿入”, 山元研究室 福岡工業大学 情報通信工学科 卒業論文 (2008)
- [7] Gimp, <http://www.gimp.org/>
- [8] i-nigma, <http://www.i-nigma.com/i-nigmahp.html>
- [9] QRReader for iPhone, <http://itunes.apple.com/us/app/qr-reader-for-iphone/id368494609?mt=8>

[10] Best Barcode Reader,  
<https://itunes.apple.com/us/app/best-barcode-scanner-scanner/id454087075?mt=8>

【指導教員】 若原俊彦