

# 目次

第 1 章 序論	1
第 2 章 QR コード	2
2.1 Reed-Solomon 符号 . . . . .	2
2.2 QR コードの概要 . . . . .	4
第 3 章 Aesthetic QR コード	7
3.1 色変換手法 . . . . .	7
3.2 ランダム法 . . . . .	8
第 4 章 実験	9
4.1 実験環境 . . . . .	9
第 5 章 結果	10
5.1 評価 . . . . .	10
5.1.1 最小ハミング距離 . . . . .	10
5.1.2 $MSE$ (平均二乗誤差) . . . . .	10
5.1.3 $PSNR$ (ピーク信号雑音比) . . . . .	11
謝辞	13
参考文献	14
付録 A その他	15

## 第1章 序論

QR コード<sup>[1]</sup>は、1994年に株式会社デンソーウェーブが開発した二次元バーコードであり、食品や交通分野など多方面の分野で利用されている。白と黒の正方形のモジュールから構成され、ランダムな見た目で表される。一般的な QR コードはデザイン性を考慮していないが、一方で広告、サービス業界ではデザイン性を考慮した QR コードが求められている。そのため、一定のルールからマトリックスを変更することによって、QR コード上にロゴ画像（以後、目的画像と述べる）を埋め込んだものがある。このような QR コードを Aesthetic QR コード<sup>[3]</sup>という。

Aesthetic QR コードの従来研究は大きく三種類に分けることができる。

1. QR コードの一部に目的画像を埋め込む方法。
2. 画像のヒストグラムを考慮して目的画像を埋め込む方法。<sup>[2]</sup>
3. RS 符号のパディングビットを考慮して目的画像を埋め込む方法。<sup>[3]</sup>

本研究は、上に述べた三つ目に分類される RS 符号のパディングビットを考慮して目的画像を埋め込む方法を考察し、目的画像に近い Aesthetic QR コードを自動生成する方法について検討する。また、論文<sup>[3]</sup>で提案されているランダム手法のアルゴリズムをベースとし、ソフトウェアを実装している。

## 第2章 QR コード

### 2.1 Reed-Solomon 符号

Reed-Solomon 符号 (以下、RS 符号) は誤り訂正符号の一つであり、有限体や拡大体上で実装される。その誤り訂正能力は高く QR コードでも応用されている。付加した符号を用いることで誤りを検知でき、決められた数以下のノイズであれば誤りを訂正できる。

拡大体  $GF(p^j)$  上において、符号のデータは複数のビットを 1 つのデータ単位として扱い、このデータ単位のことをシンボルと呼ぶ。

RS 符号は RS 符号の符号長を  $n$  シンボル、シンボル・エラー訂正最大数を  $d$  個、データ長を  $k$  シンボルと置くとき、 $(n, k)$  符号と呼ぶ。RS 符号のデータ長  $k$  シンボルはデータ部と呼ぶ。データ長  $n - k$  シンボルはパリティ部と呼び、誤り訂正に使われる。

計算は有限体や拡大体で行われる。符号で用いられる値は原始多項式の元  $\alpha$  で定義される。RS 符号の計算を行う拡大体が  $GF(p^j)$  上では、原始多項式  $G(x)$  は次のようになる

$$G(x) = x^j + b_j x^{j-1} + \cdots + b_1 x^1 + 1 = 0 \quad (2.1)$$

ここで  $b_j, \dots, b_1 \in GF(p^j)$  である。

データである  $k$  シンボルを係数に持つ多項式は情報多項式と定義する。これは

$$F(x) = \alpha_n x^{n-1} + \alpha_{n-1} x^{n-2} + \cdots + \alpha_{2d+1} x^{2d} \quad (2.2)$$

のように表すことができる。ここで  $\alpha_n, \dots, \alpha_{2d+1} \in GF(p^j)$  である。

RS 符号の生成多項式は  $d$  個のシンボル・エラーを訂正でき、連続した  $\alpha$  のべき乗での  $2d$  個の根を持つ。

$$g(x) = (x - \alpha^1)(x - \alpha^2) \cdots (x - \alpha^{2d}) \quad (2.3)$$

符号化によって生成するパリティ符号の数は  $2d$  シンボルであり、パリティ部は  $n - k$  シンボルであるため、

$$2d = n - k$$

が成り立つ.

符号化は情報多項式を生成多項式で割り、その余りを計算して情報多項式に加えることである. RS 符号化によって計算された余りは  $F(x)$  を  $g(x)$  で割った際にできるものであるため最大次数が  $2d-1$  となり、それをパリティ多項式  $a(x)$  と表す.

$$F(x) \bmod g(x) = a(x) = \alpha_{2d}x^{2d-1} + \cdots + \alpha_1 \quad (2.4)$$

ここで  $\alpha_{2d}, \dots, \alpha_1 \in GF(p^j)$  である.

有限体、拡大体上での演算は XOR 演算で行われ、情報多項式にパリティ多項式を足すことにより、情報多項式の余り部分が消える. そのため、情報多項式は生成多項式で割り切れるようになる.

情報多項式とパリティ多項式の和  $F(x) + a(x)$  は

$$F(x) + a(x) = \alpha_n x^{n-1} + \alpha_{n-1} x^{n-2} + \cdots + \alpha_1 \quad (2.5)$$

と表すことができ、これを  $R(x)$  と定義する. この  $R(x)$  の係数が RS 符号である.

RS 符号のデータ部は固定長である. QR コードに入れる文字列が既定の数より少ない場合、データ部の係数を所定の数に合わせるためパディングビット（埋め草コード）を付加する. 入力データを RS 符号の係数に落とし込んだ時の個数を  $\hat{k}$  と表すとき、入力データ部分を表す RS 符号の係数は

$$\alpha_{\hat{k}}, \dots, \alpha_1 \quad (2.6)$$

である.  $\hat{k} < k$  の時、RS 符号のデータ部の係数を  $k$  個に合わせるため、パディングビットを付加する. これにより、RS 符号のデータ部の係数は

$$\alpha_k, \dots, \alpha_{\hat{k}+1}, \alpha_{\hat{k}}, \dots, \alpha_1$$

と表す. なお、パディングビットは

$$\alpha_k, \dots, \alpha_{\hat{k}+1} \quad (2.7)$$

と表す.

RS 符号の全体図を図 2.1 に示す.

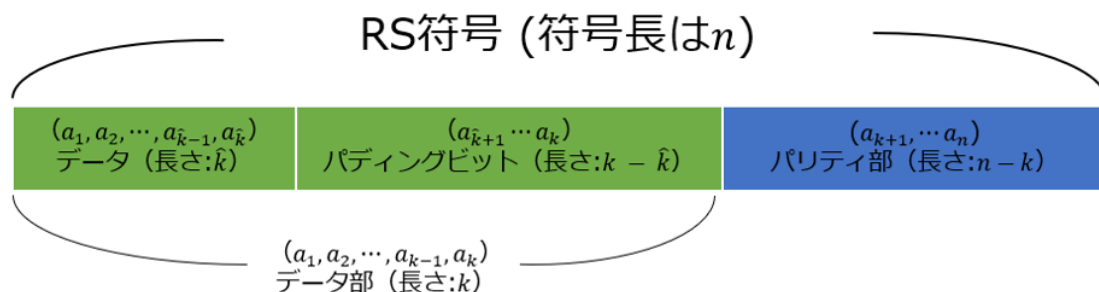


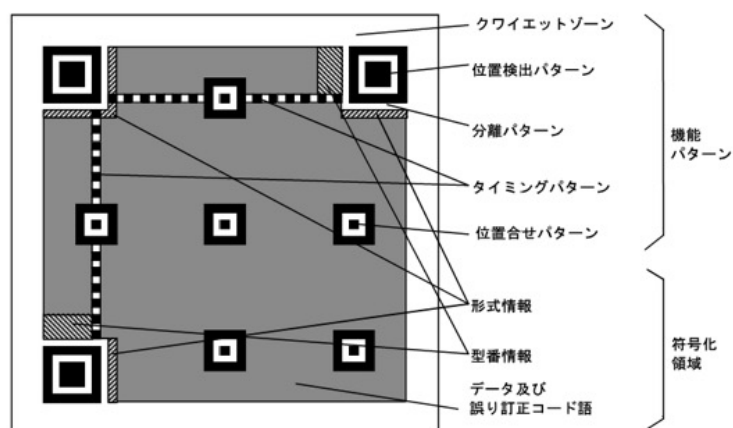
図 2.1 RS 符号の全体図

パディングビットは QR コードに入力するデータとは関係ないため、自由に変更できる. パディングビットを変更することにより、RS 符号を形成する多項式の係数は変わる. そのため、RS 符号化で得られる符号の値も変化する.

## 2.2 QR コードの概要

QR コードの構成要素の最小単位は白と黒で表されるモジュールであり、各モジュールには単一ビット値が割り当てられる. QR コードのサイズはバージョンによって決定され、そのバージョン ( $v$ ) は 1 ~ 40 である. 1 型は、 $21 \times 21$  モジュール、2 型は、 $25 \times 25$  モジュール、というように、型番が一つ上がるごとに一辺につき 4 モジュールずつ増加し、40 型は、 $177 \times 177$  モジュールとなる. したがって、バージョン  $v$  は  $(17 + 4v) \times (17 + 4v)$  モジュールである.

図 2.2 に QR コードの構成要素を表す.

図 2.2 QRコードシンボルの構造<sup>[4]</sup>

QRコードは、QRコード上にある符号化されたデータを正確に認識するために機能パターンを持っている。機能パターンは主に3つの構成要素から成り立っており、それぞれ位置検出パターン、位置合わせパターン、タイミングパターンと呼ばれる。

位置検出パターンはQRコードの左上、左下、右上の角にある3つの正方形のブロックである。それらの境目を明白にするために、形式情報との間に白いモジュールを置く。これを分離パターンと呼ぶ。

位置合わせパターンは小さな正方形のブロックで、位置検出パターンの垂直・平行座標に関係する位置に置く。バージョンによっては付加しない場合もあり、バージョン1には存在しない。

タイミングパターンは左上の位置検出パターンから右上の位置検出パターンへと、左上の位置検出パターンから左下の位置検出パターンへの白黒が交互に並ぶ2つのラインのことである。

QRコードのデータビットは、QRコードの右下から始まり、2モジュール幅の列上に配置する。列が最上部に達すると、次の2モジュール列は右端から始まり、下方向へ続く。現在の列が端に達すると、次の2モジュールの列に移動して方向を変更する。データビットは機能パターン（位置検出パターン、タイミングパターン、位置合わせパターン）の位置では、次のモジュールへ配置される。

上方向のデータビットの配置は表 2.1 に、下方向のデータビットの配置は表 2.2 に示す。

QRコードは、誤り訂正にRS符号を使用し、その能力はL、M、Q、Hの4つのレベルに昇順で分類される。各誤り訂正レベルはQRコード内の全シンボルの約7%、約15%、約25%、約30%までのシンボルを訂正することができる。それぞれを表 2.3 に示す。

表 2.1 上方向のビット配列

0	1
2	3
4	5
6	7

表 2.2 下方向のビット配列

6	7
4	5
2	3
0	1

表 2.3 誤り訂正レベル

レベル	L	M	Q	H
誤り訂正能力	約 7%	約 15%	約 25%	約 30%

## 第3章 Aesthetic QR コード

### 3.1 色変換手法

QR コードの標準形式にしたがって、各モジュールは基本的に黒と白で表現される。しかし、以下の理由によりモジュールに色を使用することは可能である。

---

**Algorithm 1** 論文<sup>[3]</sup>の色変換手法

---

入力: サイズ  $L \times L$  の目的画像

出力: 閾値  $\bar{Y}$

方法: 1. 入力画像の大きさは、QR コードのバージョン  $v$  と同じサイズに予め変更しておく。RGB 色成分は YUV 色成分に変換され、輝度 (Y) 成分  $Y_{i,j}$  ( $1 \leq i, j \leq L$ ) が得られる。

2. その中心の正方形 (元の画像サイズの  $\frac{1}{4}$ ) の値の平均  $\bar{Y}$  が計算する。

$$\bar{Y} = \frac{4}{L^2} \sum_{i=\frac{L}{4}}^{\frac{3L}{4}-1} \sum_{j=\frac{L}{4}}^{\frac{3L}{4}-1} Y_{i,j} \quad (3.1)$$


---

閾値を用いて目的画像から目的画像の二値行列を作成するアルゴリズムを以下に示す。

---

**Algorithm 2** 目的画像に紐づいた二値行列の生成

---

入力: サイズ  $L \times L$  の目的画像の輝度 (Y) 成分  $Y_{i,j}$  ( $1 \leq i, j \leq L$ )、閾値  $A$

出力: 二値行列  $B_{i,j}$

方法: 1. 目的画像を二値化する際、二値行列  $B_{i,j}$  は、以下の規則によって決定される。

$$B_{i,j} = \begin{cases} 1 & Y_{i,j} > A \\ 0 & otherwise \end{cases} \quad (3.2)$$


---



論文<sup>[3]</sup>は二値行列  $B_{i,j}$  を求める際に式 (3.2) の閾値  $A \rightarrow \bar{Y}$  を代入する.

## 3.2 ランダム法

ここにランダム法の説明を書く (これより前の章で RS 符号の符号化を説明しないといけない)

QR コードを作成する手順を以下に説明する.

---

**Algorithm 3** 論文<sup>[3]</sup>のランダム法

---

入力: バージョン 1 の QR コードに入る範囲内の文字列、かつサイズ  $21 \times 21$  の目的画像

出力: サイズ  $21 \times 21$  のバージョン 1 の QR コード

- 方法:
1. 目的画像の画素値を QR コードのモジュールに割り当て、Algorithm2 で決まった閾値  $\bar{Y}$  でモジュールを二値化し、二値行列  $B_{i,j}$  を作成する.
  2.  $B_{i,j}$  に所定のマスキングパターンを作用させる.
  3. 式 (2.7) の  $\alpha_{x_i} (\hat{k} + 1 \leq x_i \leq k)$  を変化させることにより、 $B_{i,j}$  とのハミング距離が最小となる RS 符号を見つける. 以後この手順を  $N$  回繰り返すことにより、最良の RS ブロックの組を決める. 一定の試行回数終了後、QR コード上 RS 符号は、最良の RS 符号  $\alpha_{x_i}$  に置き換える.
  4. マスク処理前である QR コードの各モジュールに対して、所定のマスクパターンを適用する.
- 

これにより、サイズ  $21 \times 21$  の QR コードを作成する.

## 第4章 実験

この章では

### 4.1 実験環境

実験に使用した PC 環境、言語を以下に示す.

- ソフトウェア実装環境
  - CPU : Intel(R)Core(TM) i7-7700 CPU @ 3.60GHz 3.60GHz
  - OS : Windows 10 pro
  - 実装 RAM : 16.0GB
- 開発環境
  - Java : Eclipse 4.11.0
  - Maple : Maple 2017.3

実験に使用した各パラメータを以下に示す.

- QR コード
  - 入力文字 : FUKUDA
  - バージョン ( $v$ ) : 1
  - マスクパターン : 001
  - 誤り訂正レベル : M

## 第5章 結果

### 5.1 評価

画像の評価は

1. 目的画像の二値行列  $B_{i,j}$  と得られた QR コードの二値行列におけるハミング距離の総数
2.  $MSE$ (平均二乗誤差)
3.  $PSNR$ (ピーク信号対雑音比)

で行う.

#### 5.1.1 最小ハミング距離

2つの符号語つの符号語  $a = (a_1, a_2, \dots, a_n)$  と  $b = (b_1, b_2, \dots, b_n)$  で対応するビット (桁) で値 (0 または 1) が異なっているビット (桁) の数をハミング距離と言い、記号で  $d(a, b)$  と書く. その中でも一番ハミング距離が小さいものを最小ハミング距離と呼ぶ.

ハミング距離は2つの符号語  $a = (a_1, a_2, \dots, a_n)$  と  $b = (b_1, b_2, \dots, b_n)$  に対して以下の式で定義される.

$$d(a, b) = \sum_{i=1}^n (a_i + b_i) \pmod{2} \quad (5.1)$$

#### 5.1.2 $MSE$ (平均二乗誤差)

$MSE$  は元画像と処理画像との差の2乗誤差である. $MSE$  が小さければ小さいほど元画像に近い画像である.  $MSE$  は  $M \times N$  の2つの画像  $I, K$  において、以下の式で定義される.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i, j) + K(i, j)]^2 \quad (5.2)$$

### 5.1.3 PSNR(ピーク信号雑音比)

$PSNR$  は画質の再現性に影響を与える信号がとりえる最大の輝度と劣化をもたらすノイズの比率を示したものである.  $PSNR$  は以下の式で定義される.

$$PSNR = 10 \log_{10} \frac{MAX_I^2}{MSE} \quad (5.3)$$

$$= 20 \log_{10} \frac{MAX_I}{\sqrt{MSE}} \quad (5.4)$$

$MAX_I$  は画像がとりうる最大ピクセルを表す. ピクセルの1サンプルが8ビットで表現されている場合、 $MAX_I$  は255である.

画質の評価については以下に示す. 実際にソフトウェアによって生成した画像が図5.1~図5.6である.



図 5.1 従来の方法で作成した Aesthetic QR コード



図 5.2  $N = 1$



図 5.3  $N = 10$



図 5.4  $N = 100$

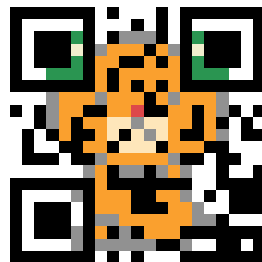


図 5.5  $N = 1000$

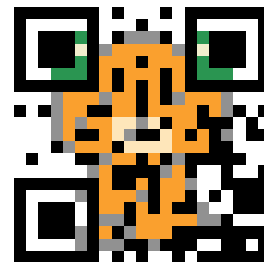


図 5.6  $N = 10000$

目的画像の二値行列  $B_{i,j}$  と得られた QR コードの二値行列とのハミング距離の総和は表 5.1 に示す.

表 5.1 目的画像の二値行列  $B_{i,j}$  と得られた QR コードの二値行列とのハミング距離の総和

生成した回数 ( $N$ )	従来の生成方法	1	10	100	1000	10000
最小ハミング距離	79	83	69	67	53	47

ランダム法の  $MSE$  値を表 5.2 に示す

表 5.2 ランダム法の  $MSE$  値

生成した回数 ( $N$ )	従来の生成方法	1	10	100	1000	10000
$MSE$ 値	24110.33	25075.25	23874.78	23680.28	21961.52	21490.57

ランダム法の  $PSNR$  値を表 5.3 に示す

表 5.3 ランダム法の  $PSNR$  値

生成した回数 ( $N$ )	従来の生成方法	1	10	100	1000	10000
$PSNR$ 値 ( $dB$ )	31.89	31.50	31.99	32.07	32.82	33.04

## 謝辞

お礼を書きます。

## 参考文献

- [1] QR code.com. <http://www.qrcode.com/en>.
- [2] Visualead. <http://www.visualead.com>.
- [3] M. Kuribayashi and M . Morii "Aesthetic QR Code Based on Modified Systematic Encoding Function", IEICE transactions on information and systems ,VOL.E100–D, NO.1,pp.42-51,2017.
- [4] JIS X0510. <http://www.jisc.go.jp/app/pager?id=2738494>.

## 付 録 A    その他

ここでは番号がすべてアルファベットに変わります.