

目次

第 1 章 序論	1
第 2 章 QR コード	2
2.1 RS 符号	2
2.2 QR コードの概要	4
2.2.1 QR コード上の RS 符号	5
第 3 章 Aesthetic QR コード	8
3.1 色変換手法	8
3.2 ランダム法	10
第 4 章 バージョン 1 の Aesthetic QR コードのソフトウェア実装と評価	11
4.1 実験環境	11
4.2 Maple による QR コード上の RS 符号のデータ部の生成	12
4.3 評価	13
4.3.1 最小ハミング距離	14
4.4 実験結果	14
第 5 章 結論	16
謝辞	17
参考文献	18
付録 A プログラムリスト	19

第1章 序論

QR コード^[1]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。一般的なQRコードは、白と黒の正方形のモジュールのみで構成されており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。デザイン性を考慮したQRコードでは、一定のルールからQRコード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^[3]という。

Aesthetic QRコードの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^[2]。
3. QRコードに利用されているRS符号中のpadding codewordsと呼ばれる領域（以下、埋め草コード語）を考慮して目的画像を埋め込む方法^[3]。

本研究は、上に述べた三つ目に分類される埋め草コード語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。Aesthetic QRコードを自動生成する手法として、本研究では、Kuribayashiらの論文^[3]で提案されているランダム手法のアルゴリズムを用い、Aesthetic QRコードを生成するソフトウェアを開発することを研究の目的とする。

以下、第2章ではQRコードを構成するReed-Solomon符号とQRコードの概要について述べ、第3章では本研究のベースとなっているKuribayashiらの論文^[3]のランダム法について述べる。第4章では実験により得られた結果について述べる。第5章では結論と今後の課題について述べる。

第2章 QRコード

この章では AestheticQRcode をの構成要素である Reed-Solomon 符号（以下、RS 符号）と QR コードについてを述べる。

2.1 RS 符号

RS 符号とは符号理論における誤り訂正符号の一つである。その高い誤り訂正能力から、QR コードなどに応用されている。

以下に、RS 符号の各用語について説明する。

1. 符号多項式

符号長 n の線形符号 C の任意の符号語をベクトル表現

$$u = (u_0, u_1, u_2, \dots, u_{n-1}) \quad (2.1)$$

としたとき式 2.1 の多項式表現は

$$u(x) = u_0 + u_1x + u_2x^2 + \dots + u_{n-1}x^{n-1} \quad (2.2)$$

である。ここで変数 x^i は単に記号 u_i の位置を示すだけである。式 2.2 のようにある符号語に対応する多項式を特に符号多項式と呼ぶ。

2. 生成多項式

ある情報記号と対応する符号多項式（以下、情報多項式） $q(x)$ からこれの誤り訂正を行う符号語 $u(x)$ を生成することを考えた際

$$u(x) = q(x)g(x) \quad (2.3)$$

と表される $g(x)$ を生成多項式と呼ぶ。

3. 体

体とは代数学においてある性質を満たした集合である。体の性質の中でも最も特徴的な点は、元 (体の要素をこのように呼ぶ) の四則演算は結果も元になる (体の中で完結する) という点である。例えば実数は体であり、実数を用いた四則演算は計算結果が全て実数になる。しかし自然数は体ではなく、例えば $1-2$ の演算結果は負の値となりこれは自然数ではないのでこれは体とは言えない。他にも満たすべき性質があるが今回は省略する。

4. ガロア体 (有限体)

体の中でも元が有限なものをガロア体 (Galois field) と呼び、有限体とも呼ばれる。元の数 q のガロア体を $GF(q)$ で表し、元 q は素数、あるいは素数のべき乗である必要がある。例えば $GF(2)$ の元は一般的に 0 と 1 である。計算例として以下に $GF(2)$ 上の加算減算の計算結果を以下に示す。

表 2.1 $GF(2)$ 上の加算結果

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 0 \end{aligned}$$

表 2.2 $GF(2)$ 上の減算結果

$$\begin{aligned} 0 - 0 &= 0 \\ 0 - 1 &= 1 \\ 1 - 0 &= 1 \\ 1 - 1 &= 0 \end{aligned}$$

このように $GF(2)$ 上では加算減算の結果が同じになる。

5. 拡大体, 原始元, 原始多項式

ガロア体 $GF(p)$ 上の既約多項式 (これ以上因数分解できない多項式) $g(x)$ を選び、その根 ($g(x) = 0$ となるような x の値) を α とする。この α を $GF(p)$ の元に追加することで新たな体が生成でき、そうしてできた新たな体を拡大体と呼ぶ。この時の α を原始元といい、この既約多項式は原始多項式という。拡大体の例として複素数が挙げられる。複素数は実数の拡大体であり実数上の既約多項式 $x^2 + 1$ の根を i として実数の元に追加したものである。

拡大体 $GF(2^m)$ 上の RS 符号について考える。

拡大体 $GF(2^m)$ の原始元を α とするとき、 $\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2^t}$ を根として持つ $GF(2^m)$ 上の生成多項式 $g(x)$ により生成される符号を t 重誤り訂正 RS 符号と呼ぶ。

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、 21×21 モジュール、2型は、 25×25 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、 177×177 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2.1にQRコードの構成要素を表す。

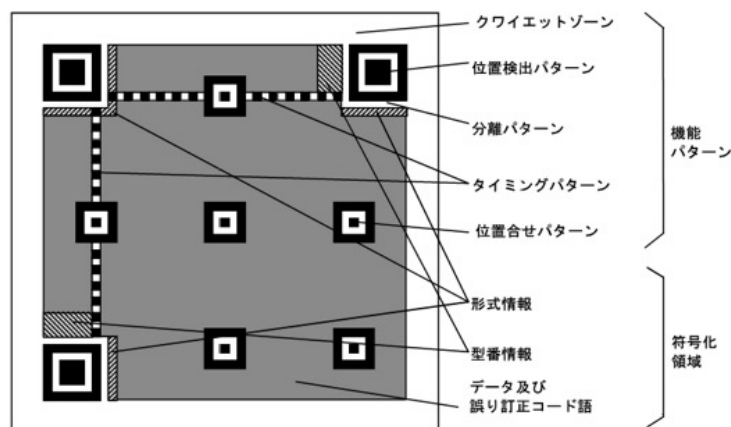


図2.1 QRコードシンボルの構造^[7]

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 2.3 に、下方向のデータビットの配置は表 2.4 に示す。

表 2.3 上方向のビット配列

0	1
2	3
4	5
6	7

表 2.4 下方向のビット配列

6	7
4	5
2	3
0	1

またデータの格納方法にも種類があり、英数字モードや8 ビットバイトモードなどがある。今回は Maple を用いて実装する都合により8 ビットバイトモードを使用した。詳しくは第4章にて記載する。

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 2.5 に示す。

表 2.5 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%

2.2.1 QR コード上の RS 符号

以下に QR コード上での RS 符号について示す。

QR コードにおいて拡大体は $GF(2^8)$ を使用し、原始多項式は $x^8 + x^4 + x^3 + x^2 + 1$ を使用する。この原始元 α を用いて計算を行う。

本来の RS 符号と QR コード上の RS 符号の違いは、RS 符号は符号長の長さが情報多項式によって決まるのに対し、QR コード上の RS 符号は情報多項式(今回の場合では入力情報を多項式表現したもの)に関わらず、RS 符号の長さが固定されているという点である。例えば

バージョン1のQRコードを構成するRS符号の長さは26シンボルとなっている。よって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埋め草コード語という情報を持たないシンボルを付加する必要がある。入力情報をRS符号化した際のシンボルの個数を \hat{k} と表すとき、入力情報部分を表すRS符号の係数は

$$\alpha_1, \dots, \alpha_{\hat{k}} \quad (2.4)$$

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、埋め草コード語を付加する。これにより、RS符号のデータ部の係数は

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

と表す。なお、埋め草コード語は

$$\alpha_{\hat{k}+1}, \dots, \alpha_k \quad (2.5)$$

と表す。

QRコード上のRS符号の全体図を図2.2に示す。

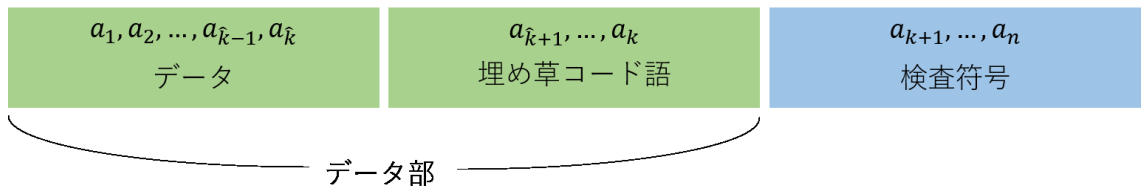


図2.2 QRコード上のRS符号の全体図

QRコード上の生成多項式^{[5] [6]}は $n-k$ 次多項式であり、これを $g(x)$ とする。

$$g(x) = (x-1)(x-\alpha) \cdots (x-\alpha^{n-k-1}) = \prod_{i=0}^{n-k-1} (x-\alpha^i) \quad (2.6)$$

式2.6の $g(x)$ を展開した多項式を

$$g(x) = g_1 x^{n-k} + g_2 x^{n-k-1} + \cdots + g_{n-k+1}, \quad g_1 = 1 \quad (2.7)$$

とする。その係数列 $g_1 = 1, g_2, \dots, g_{n-k+1}$ から定まる $k \times n$ 行列

$$G_0 = \begin{bmatrix} 1 & g_2 & \cdots & g_{n-k-1} & 0 & \cdots & \cdots & 0 \\ 0 & 1 & g_2 & \cdots & g_{n-k-1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & g_2 & \cdots & g_{n-k-1} & 0 \\ 0 & \cdots & \cdots & 0 & 1 & g_2 & \cdots & g_{n-k-1} \end{bmatrix}$$

を生成行列と呼ぶ。組織符号と呼ばれる、情報部とデータ部が分かれている符号を得るには、行列 G_0 を”掃き出し法”によって $G = [I_k, P]$

(I_k は $k \times k$ 単位行列) という標準形に変形する。

データ u に対する符号語 v は標準形の生成行列 $G = [I_k, P]$ により、

$$v = uG \tag{2.8}$$

として表現することができ、この v が QR コード上の RS 符号である。

第3章 Aesthetic QR コード

本章では、QR コード上に配置された二値配列が埋め込む目的画像のモジュールパターンと類似した、より埋め込む画像に近い QR コードを得るために用いた Kuribayashi らの論文^[3]のランダム法について述べる。

Aesthetic QR コードを作成するにあたり、本研究では QR コードに埋め込む画像を目的画像と定義する。

Aesthetic QR コードの例を図 3.1 に示す。



図 3.1 Aesthetic QR コード例

3.1 色変換手法

ランダム法を使用する際に目的画像を二値化して QR コードとのハミング距離を測り、最もハミング距離の小さい QR コードを用いて Aesthetic QR コードを作る。目的画像を二値化画像にする際に閾値を指定するが、その閾値を Algorithm 1 に示す色変換手法で決定する。

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 Algorithm 2 に示す。

Algorithm 1 論文^[3]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。

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

Algorithm 2 論文^[3]の目的画像に対する二値行列の生成

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 \bar{Y}

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。

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

3.2 ランダム法

ランダム法とは最も見た目が目的画像に近い Aesthetic QR コードを生成する方法である。まず、AestheticQR コードの元となる QR コードを生成するための生成行列 G を求める。ランダム法ではこの G を掃き出し法により $(I|P)$ の形に変形した後、RS 符号中のパディングビットに当たる I の後半の列と P をランダムに入れ替え、これにより変形された G を用いて QR コードを生成する。次に、QR コードと目的画像の画像サイズを等しいものとして、QR コードの 1 モジュールと目的画像の 1 画素を対応させ、目的画像と QR コードがどれだけ近いものか比較する。具体的には、目的画像の二値化を行ったものに QR コードのデータ (RS 符号中のデータ部の中のデータにあたる箇所) を書き加えたものを用意し、それと生成した QR コードとのハミング距離を取る。以上を N 回繰り返しハミング距離が最小となる QR コードを求め、それを使って Aesthetic QR コードを作成する。

本研究では QR コードの中でバージョン 1 の QR コードを用いた。バージョン 1 の QR コードを作成する手順を Algorithm3 に示す。

Algorithm 3 論文^[3]のランダム法を用いたバージョン 1 の Aesthetic QR コード

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

出力: サイズ 21×21 のバージョン 1 の Aesthetic 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 (2.5) の α_t ($\hat{k} + 1 \leq x_t \leq n$) を変化させることにより、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、 $B_{i,j}$ とのハミング距離が最小となる RS 符号を見つける。一定の試行回数終了後、QR コード上の RS 符号は、最良の RS 符号に置き換える。
 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。
 5. 目的画像と QR コードをかさねて Aesthetic QR コードを作成する。
-

第4章 バージョン1のAesthetic QRコードのソフトウェア実装と評価

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 i5 7500 3.4GHz
 - OS : Windows 10 pro
 - 実装 RAM : 16.0GB
- 開発環境
 - Maple : Maple 2021.1

実験に使用した各パラメータを以下に示す.

- QRコード
 - RS 符号 : (26,16) 符号
 - 入力文字 : tahara
 - バージョン (v) : 1
 - マスクパターン : 001
 - 誤り訂正レベル : M

QRコードのバージョン1の構造を図4.1に示す.

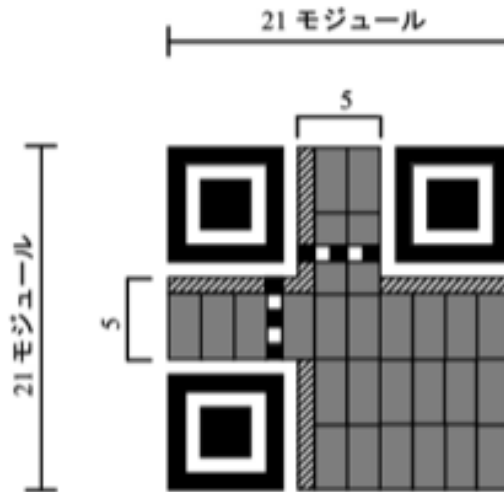


図 4.1 QRコードのバージョン1の構造

4.2 MapleによるQRコード上のRS符号のデータ部の生成

本研究ではAesthetic QRコードの生成に数式処理Maple(以下、Maple)を用いた。理由はRS符号を生成する際に必要な有限体の実装を容易に行うことができるからである。以下にMapleにおいて有限体や各多項式をどのように用いて、QRコード中のRS符号を生成したかを示す。

拡大体 $GF(2^8)$ 、原始多項式 $x^8 + x^4 + x^3 + x^2 + 1$ とその原始元 α は以下のように表される。Mapleにおいて原始多項式はGF関数の第3引数で定義され、原子元は a とした。

```
1 G8 := GF(2, 8, alpha^8+alpha^4+alpha^3+alpha^2+1):
2 a := G8:-ConvertIn(alpha):
```

生成多項式はベクトルを生成するVector関数を用いて以下のように定義した。

Vector関数の第一引数はベクトルの要素数であり、RS符号の符号長を n シンボル、データ長を k シンボルとした際 $n-k+1$ である。第二引数はベクトルの各要素を表しており、入っている値は生成多項式を因数分解した多項式 $\alpha^0 x^{10} + \alpha^{251} x^9 + \alpha^{67} x^8 + \alpha^{46} x^7 + \alpha^{61} x^6 + \alpha^{118} x^5 + \alpha^{70} x^4 + \alpha^{64} x^3 + \alpha^{94} x^2 + \alpha^{32} x + \alpha^{45}$ をの各 α の次数を格納している。

```
1 GP := Vector(n-k+1, [0,251,67,46,61,118,70,64,94,32,45]):
```

次にQRコード上のRS符号のデータ部を生成する。

今回はFP_binというベクトルにQRコード上のRS符号を格納していく。

まずこのベクトルにモード指示子と文字数指示子を加える。変数 L は入力情報の文字数を

表している. `binarize` 関数は自作の関数であり、これは引数の数値を二進数にし 8 ビットにすることで、QR コード上のコード語に変換している.

```
1 FP_bin := [0,1,0,0];
2 FP_bin := [op(FP_bin),op(binarize(L))];
```

以下は入力情報を 2 進数にしてリストに格納している箇所である. `str` には入力情報が入っており、これを `ToByteArray` 関数により数値に変換している. 得られた数値を二進数で表すと、QR コードの 8 ビットバイトモードに用いられている文字コードと一致する数値が得られるため、この関数を用いている.

```
1 Str_bin := ToByteArray(str);
2 for i from 1 by 1 to L do
3   tmp := op(binarize(Str_bin[i]));
4   FP_bin := [op(FP_bin),tmp];
5 end do;
```

以下は、`FP_bin` に終端パターンを付け加える箇所である. これは QR コード上の RS 符号のデータ部がここまでであるということを示している.

```
1 FP_bin := [op(FP_bin),op([0,0,0,0])];
```

以上の操作により、QR コード上の RS 符号のデータが生成できた.

これに埋め草コード語を追加することで RS 符号のデータ部が生成される. この埋め草コード語は `Algorithm1`, `Algorithm2` によって書き換えられた目的画像に、これまでの Maple の操作によって生成されたデータを埋め込んだものによって生成される. また QR コード上の RS 符号はバージョン 1 では 26 シンボルあるが、そのうちデータを示すシンボル以外の箇所から、`Algorithm3` よりランダムに埋め草コード語が入る箇所を決めて埋め草コード語を挿入する.

以上の操作により、Maple による QR コード上の RS 符号のデータ部の生成ができた.

4.3 評価

Aesthetic QR コードの評価は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、符号化領域中のデータ及び誤り訂正コード語のハミング距離をとることで行う.

ハミング距離は Kuribayashi らの論文^[3]で使われている評価であり、小さければ RS 符号と目的画像の二値化行列が近いことを指す. RS 符号の生成にかかった時間は短いほど、手軽に作成することが可能であることがわかる. 時間は 10 回生成して平均をとったものを評価している.

4.3.1 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 $b = (b_1, b_2, \dots, b_n)$ に対応するビット (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。

$$d(a, b) = \sum_{i=1}^n (a_i + b_i) \pmod{2} \quad (4.1)$$

4.4 実験結果

本実験ではQRコードに挿入する画像は以下の目的画像 (図4.2) を使用した。図4.3~4.7はランダム法の試行回数が $N = 1, 10, 100, 1000, 10000$ の場合の結果 (Aesthetic QRコード) を表す。実際にソフトウェアによって生成した画像が図4.3~図4.7である。

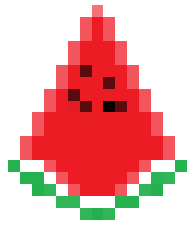


図 4.2 目的画像

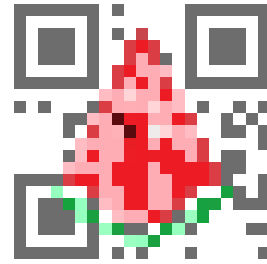


図 4.3 $N = 1$

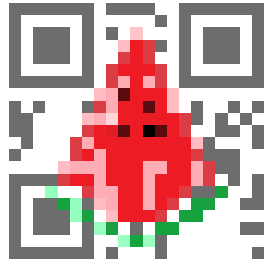


図 4.4 $N = 10$

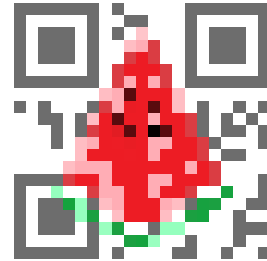
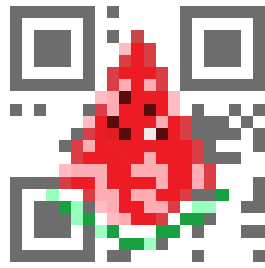
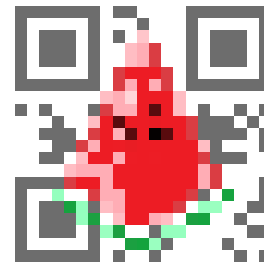


図 4.5 $N = 100$

図 4.6 $N = 1000$ 図 4.7 $N = 10000$

また、表 4.1 に各試行回数における実験で得られたハミング距離、 $PSNR$ 値、RS 符号の生成にかかった時間を表す。

表 4.1 結果のまとめ

生成した回数 (N)	1	10	100	1000	10000
最小ハミング距離	83	63	51	47	41
時間 (秒)	0.08	0.81	8.01	80.39	771.17

表 4.1 を見ると、試行回数を増やしていくたびに評価の値が向上しており、より良い Aesthetic QR コードが生成されているのがわかる。しかし、試行回数 10000 回の所を見ると RS 符号が生成するのにかかる時間が約 13 分もかかっており、あまり実用的ではない。

第5章 結論

本研究ではランダム法によって生成されるバージョン1のAesthetic QRコードのMapleによる実装を行った. 今後の課題として、

- 生成速度を短縮するアルゴリズムの実装
- 生成可能なAesthetic QRコードのバージョンの追加

などが挙げられる.

謝辞

本研究に際して、日々、様々なご指導をいただきました甲斐博准教授に心より感謝致します。最後に日頃から助言や励ましをいただきました諸先輩方、並びに同研究室の皆様に深く御礼を申し上げます。

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] ユークリッド復号法,技術の原点.https://www.jstage.jst.go.jp/article/essfr/4/3/4_3_183/_pdf
- [5] 池田和興, 例題が語る符号理論, 共立出版, 2007 年
- [6] J. Justesen and T. Hoholdt "A Course In Error-Correction Codes", European Mathematical Society Publishing House, 2004.
- [7] JIS X0510. 情報技術－自動認識及びデータ取得技術－QRコード バーコードシンボル体系仕様
<http://www.jisc.go.jp/app/pager?id=2738494>.

付 録A プログラムリスト