

# HiddenQR コード：一般画像に埋め込まれた QR コードについて

HiddenQR code : QR code embedded in the general image

草薙 祥広 †      古本 啓祐 ‡      森井 昌克 ‡  
Yoshihiro Kusanagi   Keisuke Furumoto   Masakatu Morii

## 1 はじめに

近年、QR コード [1] はカメラ機能を標準搭載した携帯端末の普及に伴い様々な場面で利用されている。QR コードに代表される二次元コードは縦横 2 方向に情報を保持できるため、バーコードのような一次元コードと比較して格納できるデータ容量が飛躍的に増加しており、高密度な情報の共有を可能としている。また、QR コードは優れた位置検出能力や誤り訂正能力を備えているため、非常に短い処理時間で優れた認識精度を有している。QR コードの主な用途に特定の Web サイトへの誘導が挙げられ、URL などの情報を格納して広告紙面上に数多く掲載されている。このように QR コードは優れた性能と対応端末の普及もあり広く利用されているが、いくつかの課題も存在する。例えば、QR コードの明暗の格子状のパターンを紙面のデザイン性を損なうことなく広告紙面上に掲載するのは容易ではない。また、QR コードに画像を埋め込んだデザイン QR コードを広告紙面上に掲載する場合、デザイン QR コードは視認性の向上を目的としており多くの方式ではモジュールの明暗部分の多くを残しているため、紙面上に独特の明暗のパターンが残ってしまう。

QR コードデコーダを搭載している携帯端末のカメラ性能は近年大きく向上しており、広告紙面上に QR コードを可能な限り知覚できない範囲で埋め込みを行った場合でも、カメラを画像に合わせることで埋め込んだ QR コードの情報を読み取ることは可能である。つまり、広告紙面上に QR コードを掲載する際に、QR コードの明暗模様を広告紙面上の模様と可能な限り対応させ、デザイン性を損なうことなく QR コードを掲載できる手法が求められている。本稿では一般的な画像の中にデザイン性を損なうことなく QR コードの埋め込みを可能とした HiddenQR コードの生成方法を提案する。既存の HiddenQR コードの例を図 1 に示す。図 1 の例では画像の模様を QR コードの明暗にに合わせて変更したのみであり、モジュール単位で多数のノイズが HiddenQR コード内に残っている。提案手法では QR コードを一般画像に埋め込む際に QR コードの広範囲な部分を一般画像が占め、さらに QR コード内の画像部分がよりノイズの少ない形で表現できる QR コードの生成を可能としている。また、デザイン QR コードを生成する際に RS 符号の非組織符号化を利用して QR コードの広範囲な領域に画像を埋め込む方式 [3] が提案されているが、本稿の提案方式では RS 符号の非組織符号化を利用した際によりノイズの少ない形で画像の模様と QR コードの明暗を対応付けることが可能である。

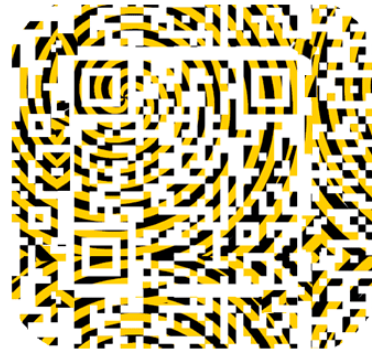


図 1 HiddenQR コード (文献 [2] より引用)

## 2 QR コード

QR コードのデザイン性について考慮したとき、QR コードの明暗の模様を目立たなくさせるために画像を埋め込むことが有効な手段である。現在、QR コード内に画像を埋め込む手法が複数提案されている。そこで本章では 2.1 節において QR コードの基礎知識について述べ、2.2 節、2.3 節で既存のデザイン QR コードについて説明する。

### 2.1 QR コードの規則

QR コードにはモジュールの配置方法や誤り訂正に関する複数の規則が存在する。本稿に関連する QR コードの規則について以下に示す。

- QR コードを構成する最小単位であるモジュールは明暗の 2 値を持ち明は 0、暗は 1 に対応しており、それぞれのモジュールが 1 ビットの情報を保持している。
- 格納するデータ容量に応じて、バージョン 1 からバージョン 40 までのバージョン (型番) が存在する。バージョンが 1 増加するごとに 1 辺あたりの 4 モジュール加えられ、バージョン 1 は  $21 \times 21$  モジュール、バージョン 40 は  $177 \times 177$  モジュールで構成される。
- 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から構成される機能パターンが存在する。位置検出パターンは明暗の比率がどの角度から見ても等間隔になるように定められており、QR コードの位置などを検出するために使用される。位置合わせパターンはバージョン 2 以上の QR コードにおいて配置され、位置検出パターンと同様に明暗の比率がどの角度から見ても等間隔となっており、画像中の歪みなどを補正するために使用される。タイミングパターンは交互に並んだ明暗のモジュールにより構成されており、各モジュールの座標を特定する際に使用される。図 1 に機

† 神戸大学工学部, Faculty of Engineering, Kobe University

‡ 神戸大学大学院工学研究科, Graduate School of Engineering, Kobe University

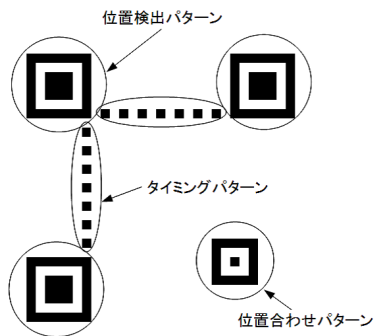


図2 機能パターン

能パターンの配置を示す。

- ・符号化領域はデータコード語、誤り訂正コード語、形式情報などから構成されており、データコード語は格納情報と埋め草コード語から成っている。QRコードに格納するデータは符号化が施され、配置規則に従い符号化領域に格子状に配置される。
- ・QRコードにはBCH符号[4]やRS符号[4]といった誤り訂正符号が使われている。BCH符号は後述する誤り訂正レベル指示子(2ビット)とマスクパターン参照子(3ビット)を含む形式情報の符号化の際に使用されている。RS符号は局所的な誤りであるバースト誤りに強い誤り訂正符号であり、有限体 $GF(2^m)$ 上の元をシンボルとして構成されている。QRコードでは $GF(2^8)$ 上の短縮RS符号が利用されており、RS符号の符号語は格納するデータから生成される。誤り訂正レベルは4段階(L:7%, M:15%, Q:25%, H:30%)規定されており、それぞれに2ビットの誤り訂正レベル指示子が割り当てられている。例えば、誤り訂正レベルがLの場合は最大で全体のおよそ7%以下であれば誤りが加わっていても訂正可能である。
- ・全てのコードの配置が終了した後に、QRコードの読み取り精度を向上させるためにマスク処理と呼ばれる操作を行う。マスク処理は明暗が集中的に現れることを防ぐために行う操作である。8種類のマスクパターンが用意されており、それぞれに3ビットのマスクパターン参照子が割り当てられている。
- ・一般的なデコーダ各モジュール中心にサンプリンググリッドを設定し、サンプリンググリッド上の明暗情報をそのモジュールの明暗情報として認識する。

## 2.2 誤り訂正能力を利用したデザインQRコード

QRコードに画像を埋め込み、QRコードのデザイン性を持たせることで視認性を向上させる手法として、誤り訂正能力を利用する方式[5],[6]が提案されている。

誤り訂正能力を利用したデザインQRコードでは、QRコードの誤り訂正レベルを最大のHに設定し、誤り訂正能力を利用して符号語部分に画像を埋め込んでいる。しかし、画像を埋め込むことで誤り訂正能力を利用するため、一般画像を埋め込む領域が符号語の約30%以内に制限されてしまう。また、誤り訂正レベルをHに設定する必要があるため、QRコードに格納できるデータ容量が小さくなってしまふ。さらに、画像を埋め込む際に誤り訂正能力を使用しているため、データ部分に誤りが発



図3 誤り訂正能力を利用したデザインQRコード(文献[5]より引用)



図4 埋め草コード語を利用したデザインQRコード(文献[8]の手法)

生した場合にその誤りを訂正できなくなってしまうという恐れがある。誤り訂正能力を利用した画像を埋め込む手法の例としてデザインQRコードを図3に示す。

## 2.3 埋め草コード語を訂正能力を利用したデザインQRコード

符号化の方式を変更することにより、誤り訂正能力に頼らないデザイン化を可能とする手法として、埋め草コード語部分を利用する方式[7][8]が提案されている。

埋め草コード語部分を利用したデザインQRコードでは、QRコードを構成する際RS符号のエンコード時に埋め草コード語を任意に設定することで、画像の埋め込みを実現している。埋め草コード語は有意な情報を持たない符号語であり、任意に書き換えることが可能である。この方式では、誤り訂正能力の使用を最小限に抑えることができるため、符号語部分の誤りに耐性がある。しかし、QRコードはRS符号の組織符号化によって符号化が行われるため、埋め草コード語の位置も一意に決定される。したがって画像を埋め込める領域がQRコードの中心部分に限られてしまう。また、QRコードに多くの明暗模様が残ってしまうという欠点も存在する。ここで、埋め草コード語を利用した画像埋め込みに関して文献[8]の手法を用いて生成したQRコードを図4に示す。

## 3 RS符号の非組織符号化を利用したデザインQRコード

QRコード内に画像を埋め込む方法については、2章で述べたように誤り訂正能力を利用する方式や埋め草コード語部分を利用する方式が提案されているが、その他にRS符号の非組織符号化を利用した方式[3]が提案されている。RS符号の非組織符号化を利用することで、QRコードに配置するモジュールの明暗を決定する際にある程度の自由度が生まれるため、画像の模様に応じてモジュールの明暗を決定することが可能である。したがって、画像を埋め込む際に誤りとなる箇所が減少するため、より広範囲に画像を配置することが可能である。本稿で扱うHiddenQRコードに対してもRS符号の非組織符号化を利用することで、QRコードのより広い領域を一般画像の模様に対応させることが可能である。RS符号の非組織符号化について3.1節で述べ、3.2節でRS符号の非

組織符号化を利用した画像埋め込みの手法について説明する。

### 3.1 RS 符号の非組織符号化

RS 符号は  $(n, k, d)$  の符号において最小距離は  $d = n - k + 1$  となるため MDS 符号である。MDS 符号においては非組織符号化による符号語の生成が可能である。従来の QR コードの符号語の生成の際に使用される RS 符号の組織符号化では情報点部から検査点部が一意に決定されるが、RS 符号の非組織符号化を使用することで符号語の生成の際に  $nC_k$  の自由度を持つことができる。RS 符号の非組織符号化を使用することで、 $(n, k, d)$  の符号から任意に選択した  $k$  個のコード語をデータコード語とし、残りの  $n - k$  個のコード語を誤り訂正コード語とすることが可能である。QR コードに画像を埋め込む際には、埋め込む画像から生成されたコード語と QR コードの各符号語に対して選択した  $k$  コード語を一致させ、また残りの  $n - k$  コード語を RS 符号の非組織符号化により導出する。  $k$  コード語の位置を選択することで、QR コードに画像を埋め込むことが可能である。

符号語を導出する際には符号  $C$  とパリティ検査行列  $H$  の関係式  $HC^T = 0$  を利用する。 $(n, k, d)$  の符号の  $k$  個の情報点部  $s_1 s_2 s_3 \dots s_k$  から  $n - k$  個の検査点部  $p_1 p_2 p_3 \dots p_{n-k}$  を導出する流れは以下ようになる。

Step1  $(n, k, d)$  の符号の  $n$  個のシンボルから  $k$  個のシンボルを任意に選択する。

Step2 検査行列  $H$  の  $n$  行の内、選択した  $k$  個のシンボルに対応する行を行・列変形により対角化する。

Step3 検査点部の要素  $p_n$  を  $p_n = \sum_{i=1}^k a_{ni} \times s_i$  により導出する。式中の  $a_{i,j}$  は検査行列  $H$  の  $i$  行  $j$  列目の要素を表している。

### 3.2 RS 符号の非組織符号化による画像埋め込み

誤り訂正能力を利用する方式や埋め草コード語部分を利用する方式では画像を埋め込む領域に制限があった。そこで、文献 [3] では非組織符号化を利用して QR コード全面に画像を埋め込む手法が提案されている。

非組織符号化では  $n$  個の符号語のうち  $k$  個を任意に選択して符号化を行っているため符号語の構成に  $nC_k$  の自由度が生まれ、QR コードのモジュールの配置を一般画像に近付けることが可能になる。その生成方法は一般画像からビット列を導出し、そのビット列と格納データをもとに RS 符号の非組織符号化を利用して符号語を  $N$  個生成してその中から最良の符号語を選択し QR コード上に配置することで QR コード全面に画像を埋め込むというものである。そのため、この手法を用いると、誤り訂正能力を利用する方式や埋め草コード語部分を利用する方式のように画像を埋め込む領域の制限が無く、QR コード全面に画像を埋め込むことが可能である。しかし、この手法で画像を埋め込んだ QR コードを生成する過程において、一般画像の 1 ピクセルを QR コードのモジュールに合わせているため、QR コードに画像を埋め込んだ際に画像の解像度が低下し、ノイズが QR コードのモジュール単位で発生してしまう。そこでこの問題を解決する手法を次章にて提案手法として説明する。

1	2	3
4	5	6
7	8	9

図 5 モジュール内のピクセルの配置

## 4 RS 符号の非組織符号化を利用した HiddenQR コード

本稿では QR コードを生成する際、携帯端末などに搭載されている従来の QR コードデコーダで読み取りが行えることを条件としている。また QR コードを読み取る際に位置検出パターンなどの機能パターン情報を取得できなければ一般的なデコーダで読み取ることは困難である。よってこの QR コードの機能パターンは本稿の提案手法では残すこととし、その上でその他の広範囲の領域をよりノイズの少ない形で一般画像が占める QR コードを生成する方式を提案する。まず、4.1 節では従来のモジュール単位ではなくピクセル単位で一般画像と QR コードの明暗の対応付けを行うことで 3.2 節の従来手法よりもノイズを減少させた方式を示し、4.2 節ではさらに生じたノイズの画像内に占める割合を減少させた方式を示す。

### 4.1 ピクセル単位で画像との対応付けを行った手法

提案手法では QR コード内における一般画像の領域をより広くするため RS 符号の非組織符号化を利用して、QR コード全面に画像を埋め込む。これにより、QR コード内のより広い領域を一般画像が QR コードが占めるといった目的は達成できる。このとき、文献 [3] におけるノイズがモジュール単位で発生してしまう問題を改善する手法として、図 5 のように一般画像の  $3 \times 3$  の 9 ピクセル分を QR コードのモジュールに合わせる方法を提案する。これを提案手法 1 とする。なお図 5 において外側の大きな正方形が 1 モジュールを表し、モジュール内の 9 個の小さな正方形がそれぞれ 1 ピクセルを表す。また、ピクセル内の数字  $i = 1, 2, \dots, 9$  を用いて各ピクセルをピクセル  $i$  と定義する。

文献 [3] の手法ではモジュール単位で RGB 値が決定されていたため、図 6 のように 1 つのモジュールが同じ色で表現されてしまい、画像の解像度が低下するため、ノイズが多くなってしまった。一方提案手法では 1 モジュールを 9 ピクセル分に合わせることで、図 7 のように QR コードの 1 つのモジュールにおいても異なる RGB 値を持つピクセルを存在させることができるため、解像度が向上し、ノイズを減少させることが期待できる。

提案手法 1 で一般画像が埋め込まれた QR コードの生成する手順を以下に示す。

Step1 一般画像を用意し、各ピクセルの RGB 値を取得し、RGB 値の平均をしきい値として 2 値化を行い



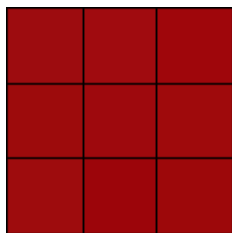


図 6 文献 [3] の手法における QR コードの 1 モジュール

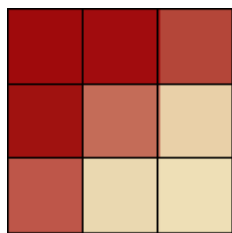


図 7 提案手法における QR コードの 1 モジュール

各ピクセルの明暗状態を決定する。

- Step2** 一般画像を 2 値化した画像データの 9 ピクセル分を QR コードのモジュールに合わせる。
- Step3** 明を 0, 暗を 1 と定義し, 画像のモジュールに合わせた部分における中心部分のピクセル (図 5 中のピクセル 5) の明暗状態からビット列  $b$  を取り出し, 決定しておいたマスクパターンと XOR 演算を行って, 符号語  $c_1$  を取り出す。
- Step4**  $c_1$  の先頭から順に格納データと置換し, 符号語  $c_2$  を導出する。
- Step5** 非組織符号化を利用して  $c_2$  の  $n$  個の符号語のうち, 選択する  $k$  個の位置を変更して符号語の候補を  $N$  個生成してそれぞれ  $c_1$  とのハミング距離を計算する。
- Step6** 符号語の候補の中からハミング距離が最も小さい候補  $c_1'$  を選択し, step3 で使用したものと同様のマスクパターンと XOR 演算を行い, 生成されたビット列を QR コード上に配置する。
- Step7** step6 において生成した QR コードを 2 値化する。
- Step8** 一般画像を 2 値化した画像データの明暗情報と, QR コードを 2 値化した画像データの明暗情報をモジュール内の全ピクセルにおいてそれぞれ比較する。
- Step9** ピクセル 5 の明暗が同じで, かつ残りの 8 つのピクセルのうち 4 つ以上において明暗情報が同じならば, そのモジュールは一般画像のモジュール内の 9 ピクセルの RGB 値それぞれを使用し, その条件を満たさないモジュールは step6 で生成した QR コードの RGB 値をそのまま使用する。

文献 [3] の手法では 3.2 節で述べたように Step2 の内容が「一般画像を 2 値化した画像データのピクセルを QR コードのモジュールに合わせる。」となっている。なお, Step2 を以上のように変更し生成手順の Step6 において生成した QR コードが文献 [3] の手法で生成する QR コードである。

提案手法では文献 [3] の手法から Step2 を変更して一般画像の 9 ピクセル分をモジュールに合わせたため, モジュール内でも RGB 値の変更が可能となり, 一般画像の RGB 値を使用する部分の面積が増加し, ノイズの数も減少するので, 一般画像の解像度も向上すると考えられる。また, 一般的なデコーダは, モジュールの明暗を決定する際にサンプリンググリッドを各モジュールの中心部分に設定することが多いが, ピクセル 5 の明暗情報のみをデ

コーダが読み取ることは困難であると考えたため, ピクセル 5 と同じ明暗情報を持つピクセルを周囲に残すことで読み取り精度を上げることができると考えられる。よって, Step9 において図 5 中のピクセル 5 の他にモジュール内の全ピクセルの明暗情報を比較し, 使用した。しかし, この手法では, ノイズになってしまうモジュールに Step6 で生成した QR コードの RGB 値をそのまま使用しているため, モジュールが単一色になってしまい, ノイズがモジュール単位で発生してしまう。そこで, この問題を解決する手法を 4.2 節で提案する。

#### 4.2 画像内のノイズを減少させた手法

提案手法 1 において, ノイズがモジュール単位で発生してしまう問題を解決するためにノイズになってしまうモジュール内の各ピクセルごとに色を変更する方法を提案する。これを提案手法 2 とする。この提案手法 2 においてモジュール内のピクセルの色の変更方法は一般画像の色やそのモジュールの周囲の色のことを考慮するものとする。

提案手法 2 での生成手順はまず提案手法 1 の Step9 を以下のように変更する。

- Step9'** ピクセル 5 の明暗が同じで, かつ残りの 8 つのピクセルのうち 4 つ以上において明暗情報が同じならば, そのモジュールは一般画像のモジュール内の 9 ピクセルの RGB 値それぞれを使用し, それ以外のモジュールは Step10 以降の処理を行う。

そして, Step10 以降の生成手順を以下に示す。なお, Step10 以降の処理は提案手法 1 においてノイズとなっていたモジュール内の処理である。Step10 でピクセル 5 の色を決定し, 残りの 8 つのピクセルの色を Step11 以降の処理で決定している。

- Step10** 一般画像においてピクセル 5 に対応する部分の RGB 値から輝度を算出し, あらかじめ設定しておいたしきい値から明暗情報が反転するように輝度を変更してその色をピクセル 5 に対して使用する。
- Step11** 隣合うモジュールに対応する部分の一般画像の 3 ピクセル分の輝度の平均値  $a$  を求める。
- Step12**  $a$  とピクセル 5 の輝度の平均値  $a'$  を求める。
- Step13** 各ピクセルにおいて, 一般画像の対応する箇所の輝度を  $a'$  に変更して, その色を使用する。

なお RGB 値から輝度を算出するには  $r$  値,  $g$  値,  $b$  値をそれぞれ取得して以下の式を用いる。

$$rgb = 0.299 \times r + 0.587 \times g + 0.114 \times b \quad (1)$$

また, Step11 の隣合うモジュールの 3 ピクセル分の決定方法に関してピクセル 1 の輝度を決定する際の例を図 8 に示す。ピクセル 1 の輝度の決定には図 8 の青色部分のピクセルの輝度の平均値を使用している。以上の提案手法 2 では, 一般画像をもとに輝度を変更したため, ノイズとなる部分の色が一般画像の色に近い色で表現できると考えられる。また, モジュール内のピクセルごとに色を変更したためノイズとなる箇所でも解像度が向上すると考えられる。

	1	2	3	
	4	5	6	
	7	8	9	

図8 ピクセル1の輝度の決定

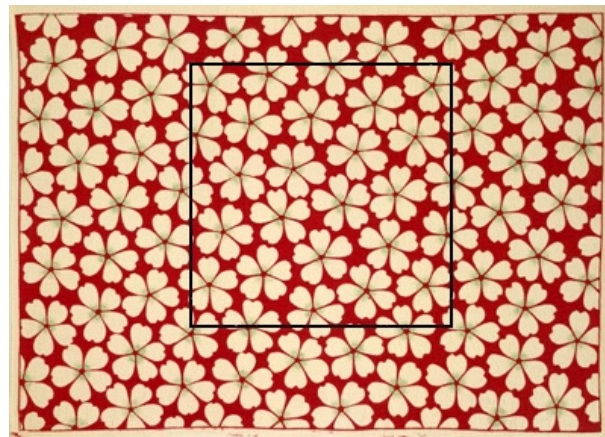


図9 一般画像(文献[9]より引用)

### 4.3 評価

本節では実際に提案手法1,2を利用してQRコードを生成し, 3.2節の文献[3]の手法での生成例と比較することで提案手法の評価を行う。

図9に今回使用した一般画像を示す。図9の一般画像中の線で囲まれた部分を使用している。図10にこの一般画像をもとに文献[3]の手法で生成したQRコードの生成例を, 図11に提案手法1で生成したQRコードの生成例を, 図12提案手法2で生成したQRコードの生成例をそれぞれ示す。ただし, 生成例はいずれもバージョン10, 誤り訂正レベルLで8ビットバイトモード, マスクパターン参照子を000に設定し, 格納データとして「<http://www.eedept.kobe-u.ac.jp/>」を使用している。また, 非組織符号化を利用して生成した符号語の候補数 $N$ は $10^6$ 個とした。さらに, 図12の提案手法2の生成例ではStep10において一般画像では明だが暗と認識されたい場合は輝度を65, 逆に一般画像では暗だが明と認識させたい場合は輝度を175と設定した。

図10のように文献[3]の手法で生成したQRコードは1モジュールが全て同じ色のため解像度が低いことが分かる。特に, 元の一般画像では曲線になっている部分の解像度が低くなってしまっており, ノイズが多くなっている。一方図11では提案手法1においてピクセル単位で色を変更したため, その部分の解像度が向上していることがわかる。実際に, 提案手法1で生成したQRコード内の1モジュール中の9ピクセル全てで一般画像のRGB値を使用しているモジュールの数は2535個であった。なお, バージョン10のQRコードは一辺のモジュール数が57なので全モジュール数は $57 \times 57 = 3249$ モジュールである。以上より, 提案手法1では約2500個のモジュールを一般画像に色に変更したためノイズの数を減少させることができた。しかし提案手法1ではノイズがモジュール単位で残ってしまっている。そこで図12の提案手法2の生成例を見ると, ノイズとなっているモジュール内のそれぞれのピクセルで異なる色を使用しているため, 提案手法1の生成例よりさらに一般画像に近くなっていることが分かる。また, ノイズとなっているモジュールの中心ピクセルの色を一般画像の色をもとに決定したため, 目立ちにくくなっている。よって, 提案手法2はQRコード内の画像のノイズ数を減らし, 解像度を向上させる手法として有効であることが確認された。

また, 提案手法で生成したQRコードが従来のQRコードデコーダで読み取り可能かどうか, 実験機器としてArrows X LTE F-05Dを用いて以下の3種のデコーダを使用して実験を行った。

- QRバーコードスキャナー (WB Development Team)
- QRコードリーダーEQS (イクス株式会社)
- ScanLife (Scanbuy, Inc.)

結果としては3種全てのデコーダにおいて読み取りが可能であり, よって, 提案手法で生成したQRコードは一般的なデコーダで読み取り可能であることが確認された。

## 5 まとめ

本稿ではHiddenQRコードを生成するにあたり, 一般画像の模様をQRコードの明暗に対応付けて, 広範囲な領域をよりノイズが少ない形で表現できるQRコードを生成する手法を提案し, 評価を行った。

画像をQRコードに埋め込む既存の手法としては誤り訂正能力を利用する方式, 埋め草コード語を利用する方式, RS符号の非組織符号化を利用する方式(文献[3]の手法)がある。文献[3]の手法ではQRコード全面への画像埋め込みを可能にしているが, 全てのモジュールが単一色で表されているためノイズが大きくなってしまっている。そこで提案手法1では, 一般画像の9ピクセル分をQRコードの1モジュールに合わせることで, 1モジュールの中でも色を変更することができるようになったため一般画像の色をそのまま利用できる部分の面積が広くなり, ノイズを減少させることに成功した。しかし, ノイズがモジュール単位で発生してしまっただけで, 提案手法2ではノイズとなっていたモジュールの色をピクセル単位で変更し, さらに一般画像の色や周囲の色を考慮して各ピクセルの色を決定することにより, 提案手法1で目立っていたノイズを目立たなくすることに成功した。

## 参考文献

- [1] 日本工業規格, “JIS,X0510, 二次元コードシンボル—QRコード—基本仕様,” 2004.



図 10 文献 [3] の手法を利用した生成例

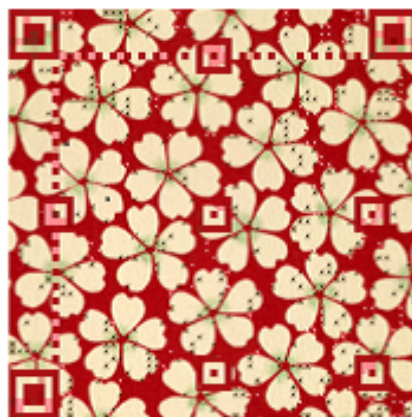


図 12 提案手法 2 を利用した生成例



図 11 提案手法 1 を利用した生成例

- [2] qrworld, <http://qrworld.wordpress.com/2011/08/04/decorating-a-qr-code—part-two/>
- [3] Kazuaki Fujita, Minoru Kuribayashi, and Masakatu Mori-i, "Expansion of image displayable area in design QR code and its applications," *Forum on Information Technology 2011(FIT2011)*, 2011.
- [4] 今井秀樹, 符号理論, 電子情報通信学会, 1990 年.
- [5] IT DeSign Inc., designQR, <http://d-qr.net/index.htm>, 2010.
- [6] A・Tcommunications CO., LogoQ, <http://logoq.net/logoq/index.html>, 2010.
- [7] HAGIWARA Manabu, QR-JAM, <http://staff.aist.go.jp/hagiwara.hagiwara/qrjam/index.html>, 2010.
- [8] NAMCO BANDAI Games Inc., 特許公開 2008 - 052588, 2008.
- [9] IMAGES ONLINE, [https://imagesonline.bl.uk/?service=search&action=do\\_quick\\_search&language=en&q=083438](https://imagesonline.bl.uk/?service=search&action=do_quick_search&language=en&q=083438)