K-017

セルの微細分割による二次元コードの情報ハイディング

Information hiding of two-dimensional code
by segmenting the cell

寺浦 信之† 櫻井 幸一‡
Nobuyuki Teraura **Kouichi Sakurai**

１．はじめに

１．１　背景

二次元コードは、バーコードに比較して多くのデータを記憶し、また誤り訂正機能を有し汚れに強いので、当初業務用途で使用されてきた。その後、携帯電話で読取れるようになってきたので、業務用途以外のＷＥＢ誘導など広い用途で使われている。

従来は、二次元コードの読取りが困難であったので、二次元コードそのものに秘匿性があった。しかし、二次元コードは誰でも読めることを目指して開発されたので、上記のように携帯電話でも読む事が可能であり、誰でもその内容を知ることが可能になってきた。

一方、特別な権限を有する者のみが読める用途も存在する。そこで、既存の二次元コードとしての公開部と付加部分である非公開部を有する二次元コードを検討する。

１．２　動機

現在普及している二次元コード[1]［17］［18]は、１９９０年代に発明され、その時代の撮像技術を前提に設計されている。しかし、ディジタルカメラの爆発的な普及を機に、撮像素子の画素数は飛躍的な拡大を遂げ、現在では１０００万画素以上の撮像素子が安価に入手可能である。

従って、これらの大きな画素数の撮像素子を前提にすれば、現在の二次元コードのセルの中に、さらに微細な構造を構成しても、これらを識別することが可能である。そこで、微細構造を活用して非公開部を有する二次元コードを実現し、公開部は既存の二次元コードと全く同様に扱え、それに秘匿性の高い部分を付加した二次元コードを検討することとした。

１．３　既存の研究

第三者からの読取りを防止するため、非公開領域を有する二次元コードが開発されている[7]。これは、公開領域と非公開領域を有し、公開領域は通常の二次元コードとして読取り可能であるが、非公開領域は暗号化キーを知っている読取り装置のみが、読取り可能である。

また、カラーの二次元コードに電子透かしを挿入し、コピー品を検出する手法が提案されている[6][15]。同じく、二次元コードに電子透かしを挿入して、認証を行い、また偽造の防止を行う提案もなされている[5][16]。

著者らは、通常の二次元コードの上に、互いに異なる波長領域の赤外線を吸収する印材を用いて二次元コードを多層に形成して積層したセルにデータを分散し、またセルを分割してデータを分散させる手法を提案した[2]。

この他に、二次元コードの大容量化を実現するために、セルの微細分割の限界の研究[14]、カラーコードを用いた多ビット符号化[8][9][10]や誤り訂正の高度化[12]の研究もなされている。

１．４　課題

この論文で目的とする秘匿領域が存在する二次元コードを実現する上での課題は、
①新しい印刷、表示、撮像技術に対応した新しい光学的情報媒体を提供する、
②既存の二次元コードとの上位互換性を維持する、
③上位互換部分を既存の読取り装置で読取り可能とする、
④非互換部分において、高い秘匿性を実現する、
ことである。

１．５　既存研究との比較

前記の赤外線を吸収する印材を用いた多層の情報媒体[2]は、データの秘匿化とコピー防止を目的としていた。それに対して、本論文の目的は通常の印材を用いた二次元コードのセルを構造化することにより、データの秘匿化を実現することにある。構造化に当たって、拡張ハミング符号を導入することにより、安定した読取りに資するとともに、実質的な誤り訂正率の向上を可能とした。

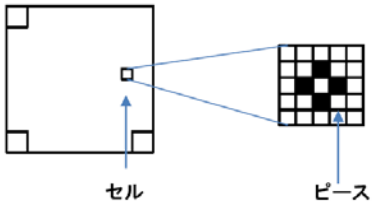
また、非公開領域を有する二次元コード[4]は、通常の非公開領域を有しない二次元コードのデータ保持形式を維持したまま、暗号化されたデータを記憶させるものであり、秘匿の原理は暗号化にある。それに対して、本論文で提案している方式の秘匿性は、適用可能な符号化テーブルの数が大数であり、真のテーブルの秘匿にある。

２．構造

この論文の提案では、通常の携帯電話などで読取れ, 従来の二次元コードと上位互換の公開部と読取れない非公開部を有する二次元コードを目指している。そこで、全体のセルの構造は通常の二次元コードと同じである。そして、それらのセルの中に微細な構造を設け、そこに１ビットから 6 ビット程度のデータを記憶させ、秘匿された方法で微細構造を解析することによりデータを取得可能とする。また、二次元コードとして、ＱＲコード[3]［14］[17]を例にして、議論を進める。

２．１　セルの分割

二次元コードはセルから構成されるが、そのセルをさらに分割した要素をピースと呼ぶ。ピースを正方形とするために、セルの縦横は同じ比率で分割する。セルとピースの関係を図１に示す。



† テララコード研究所、九州大学システム情報科学府　社会人博士後期課程, Terrara Code Research Institute
‡ 九州大学システム情報科学府, Information Science and Electrical Engineering, Kyushu University

図１　セルとピースの関係

２．２　ピースの配置と反対色の数

セルの一部のピースをセル色の反対色にすると、セルの色識別に影響を与える可能性がある。そこで、反対色にするピースの位置と数は、セルの白黒判別に影響を与えないように選択する必要がある。

ピースの数については次節において、配置については以下に検討する。

２．３　配置

セルの外縁部は、他のセルとの境界であり、セルの識別にとって重要であり、ノイズの無いセル本来の色であることが望ましい。撮像時の手振れなどで外縁部のピースは、他の接するセルの画像と重畳するので、緩衝領域でもある。

また、ピースレベルのコードを読取る装置は、従来の装置そのままではなく、専用の装置となるので、特別な手振れ対応が可能である。それに対して、セルレベルのコードを読取るのは、従来の携帯電話などであり、特別な対策をとれないので、上記の緩衝領域の役割は大きい。加えて、復号時に二次元コードのセルを区分する為に、経線と緯線を検索するが、その指標となり得る。

そこで、分割されたピースの外周部はセルの色と同じとし、その内部のピースのみを反対色にすることとする。従って、セルの分割数は、３ｘ３ピース以上となる。

３．　符号化

ここでは、３ｘ３，４ｘ４，５ｘ５のピース分割の場合の符号化について検討する。

３．１　３ｘ３構成の場合

セルの最小の分割は３ ｘ ３である。先に述べたように、外周部はセル色を維持しているので、データを保持する可変部は中央の１ピースのみとなる。従って、符号化は中央部が白か黒で１ビットを表現することとなる。この例を図２に示す。

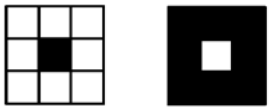


図２　３ｘ３構成のピースの符号化パターン

この場合は、中央の１ピースのみのデータであり、白か黒かのみの符号化となる。このような符号化だけでは、秘匿性は極めて低く、保持データを２倍にする効果しか期待できない。

しかし、第６節で議論するように、セキュリティＱＲコード（ＳＱＲＣ）を併用すると、秘匿性は格段に向上する。すなわち、各ピースに配置する白黒データを暗号化キーで暗号化した非公開領域を有するＳＱＲＣのルールに基づいて、配置するのである。

事実上、二層の二次元コードが存在すると見做すことができ、そのピース色で表現された二次元コードの構造をＳＱＲＣと同じにする。

３．２　４ｘ４構成の場合

４ｘ４の場合には、セル内部に２ｘ２のピースを有しており、4 ピースの白黒でデータを表現可能である。言い換えると、1 ピースから 4 ピースまでの反対色化が可能である。しかし、4 ピースが全て反転する場合には、セルレベルでの復号で誤りが発生する可能性が高まる。反転率が全体の 16%まで許容されるとすると、16*0.16=2.4 であり、2 ピースの反転が可能である

４ｘ４構成の 2 ピース反転時のピース配列の例を図３に示す。

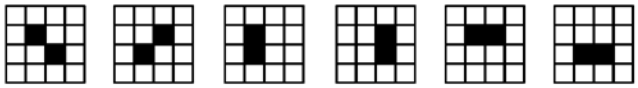


図３　４ｘ４構成の符号化パターン

この場合は、符号化のパターン数 $P_E(4)$ は、

$P_E(4) = {}_4C_2 = 6$

となる。また、$2^2=4$ を超えているので、2 ビット符号化が可能である。

・多ビット符号化

2 ビット符号化の場合には、二つのデータ配置を想定することができる。一つは、連続した一つのデータセットであり、その大きさはセルレベルで記憶されるデータの 2 倍である。もう一つの想定は、各セルが有する 2 ビットのデータを、仮想的な 2 層のデータの各ビットと考える方式である。この場合には、異なる二つのデータセットとなり、それぞれのデータ量はセルレベルのデータ量に等しい。そこで、以後の議論では、多ビット符号化では、図４に示すように、ピースレベルで複数の二次元コードを仮想するデータ構造を前提とする。

この場合には、各セル毎に、3 ビットの符号ブロック $\boldsymbol{u} = (u_0, u_1, u_2)$ が得られる。ここで、u_0 はセルレベルの配色データであり、u_1, u_2 はピースレベルの配色データである。

図４　セルレベルとピースレベルの二次元コード

・符号化テーブルの数

上記のように、符号化パターンの組み合わせの数は６通りである。１ビット符号化では、これを白や黒に割り当てる符号化テーブルの数 $N_E(4,1)$ は、半数に白または黒を割り当てるので、

$N_E(4,1) = {}_6C_3 = 20$

となる。

表１　４ピース中２ピース反転時の符号化の例

符号化色	復号色	
	1ビット符号化	2ビット符号化
白白黒黒	黒	黒白
白黒白黒	白	白黒
白黒黒白	白	白白
黒白白黒	黒	黒黒
黒白黒白	黒	黒白
黒黒白白	白	白黒

２ビット符号化では、６通りの符号化色を、白白、白黒、黒白、黒黒の４つの復号色に割当てるので、その符号化テーブルの数 $N_E(4,2)$ は、

$N_E(4,2) = {}_4C_2*({}_6C_2*{}_4C_2*2C_1*1C_1) = 1080$

となる。

表１に 1 ビット及び 2 ビット符号化の符号化テーブルの例を示す。表中の符号化色における白と黒は、セル色とその反対色を示している。

３．３　５ｘ５構成の場合

この場合には、セルの内部に３ ｘ ３のピースを有しており、9 ピースの中の白黒でデータを表現可能である。ここでも反転率が 16%まで許容されるとすると、25*0.16=4 であり、1 ピースから 4 ピースの反転が可能である。

５ ｘ ５の 4 ピース反転時のピース配列の例を図５に示す。

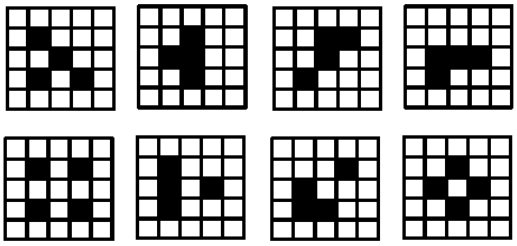


図５　５ｘ５構造の４ピース反対色の例

また、セルからピースの切り出し、識別を容易にするために、中央部をセル色に固定することを検討する。５ ｘ ５構成の場合、可変部は３ ｘ ３ピースとなるが、このピースを区分する境界が、画像によって不鮮明になる。ピース中央部をセル色に固定することで、この位置をピース切りだしのマーカとして利用でき、その境界線を引く一助となり得る。また、既存のセルレベルの二次元コードの読取り装置が、セルの中央付近の画素を識別対象としている場合も多く、既存機器対応のためにも、中央ピースをセル色とする意味がある。

中央ピースをセル色に固定した場合、8 ピースの中で、4 ピースの反転が符号化が可能である。この場合、符号化のパターン数 $P_E(5)$ は、8 ピースから 4 ピースの反対色を選択するので、

$P_E(5) = {}_8C_4 = 70$

となる。2^6=64 を超えているので、最大 6 ビットの符号化が可能である。表２に 1 ビット、2 ビット、3 ビットの符号化テーブルの例を示す。

表２　８ピース中４ピース反転時の符号化の例

符号化色	復号色		
	1ビット符号化	2ビット符号化	3ビット符号化
白白白白黒黒黒黒	黒	黒白	黒白黒
白白白黒黒黒黒白	白	白黒	白黒白
白白白黒黒黒白黒	白	白白	白白白
・・・			
黒黒黒黒白白白白	黒	黒黒	黒黒黒

・符号化テーブルの数

この配色を用いた 1 ビット符号化の符号化テーブルの数 $N_E(5,1)$ は、

$N_E(5,1) = {}_{70}C_{35} \fallingdotseq 1.1\times10^{20}$

となる。同様に、2 ビット符号化の符号化テーブルの数 $N_E(5,2)$ は、

$N_E(5,2) = {}_{70}C_{17}*{}_{53}C_{17}*{}_{36}C_{17}*{}_{19}C_{17}*2 \fallingdotseq 7.5*10^{41}$

となる。同様に、3 ビット符号化の符号化テーブルの数 $N_E(5,3)$ は、

$N_E(5,3) = {}_{70}C_8*{}_{62}C_8*{}_{54}C_8*{}_{46}C_8*{}_{38}C_8*{}_{30}C_8*{}_{22}C_8*{}_{14}C_8 *{}_6C_1*{}_5C_1*{}_4C_1*{}_3C_1*{}_2C_1 \fallingdotseq 1.7*10^{63}$

５ ｘ ５構成の場合には、6 ビット符号化まで可能である。その符号化テーブルの総数 $N_E(5)$ は、

$N_E(5) = \sum_{i=1}^{6} N_E(5,i) \fallingdotseq 1.2*10^{100}$

となる。

４．誤り訂正符号の導入

新たに導入したピースの大きさは比較的小さく、汚れなどの影響を受けて、誤った判断をする可能性はセルと比較して大きいと言える。そこで、セル内部に配置されたピースのデータに誤り訂正機能の導入を検討する。

４．１　拡張ハミングコード

ハミング符号は、ある整数 m に対し、

符号長　：n　＝　2^m-1

データ長：k　＝　n－m

で構成される。ここでデータ長とは元のデータのビット数、符号長とは生成される符号全体のビット数である。さらに、拡張ハミング符号は、1 ビットと 2 ビットの誤りの区別を判定するために、符号全体のパリティビットを追加して構成される。

ここで、m＝３の場合には、データ長４ビット、訂正ビット長３ビット、全体のパリティビット１ビットの８ビットで構成される。この８ビットの拡張ハミングコードを用いて、誤り訂正を行うことを検討する。

表３　拡張ハミング符号の例

データビット				訂正ビット			パリティ
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	1	1	0	1
0	0	1	1	1	0	1	0
0	1	0	0	1	1	1	0
0	1	0	1	1	0	0	1
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0

データビット				訂正ビット			パリティ
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	0	1	1	0
1	0	1	1	0	0	0	1
1	1	0	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	0	1	0	0	0
1	1	1	1	1	1	1	1

５ ｘ ５構成のピースは、内部側に３ ｘ ３の 9 個のピースを有しており、これに任意にセル色またはその反対色を割り当てる場合については、前節で述べた。３ ｘ ３の内側ピースには 9 個のピースがあるがその中央のピースに、セル色を配色すると可変部分は 8 ピースとなる。この 8 ピースへの配色パターンを表３に示す拡張ハミング符号のビット

配列に対応させることを考える。すなわち、表３のテーブルの中の‘０’をセル色に、‘１’をセルの反対色に対応させる。このような構成にすれば、復号時にセルと同色を‘０’に、反対色を‘１’に復号して、拡張ハミング符号の検査を行えば、汚れなどでピースが誤って識別された場合には、2 ピースまでの誤りを検出し、1 ピースの誤りを訂正することができる。

また、後述するように、ピースレベルの黒もしくは白の配色についても、セルレベルと同じ二次元コード全体の誤り訂正機能を有しているので、セル内の誤り訂正で、訂正できない場合においても、上記の誤り訂正の範囲内の誤りであれば、訂正される。

表２の符号化テーブルを、拡張ハミング符号に対応させた例を表５に示す。

表５　拡張ハミング符号に対応した符号化の例

符号化色	復号色		
	1ビット符号化	2ビット符号化	3ビット符号化
白白白黒白黒黒黒	黒	黒白	黒白黒
白白黒白黒黒白黒	白	白黒	白黒白
白白黒黒黒白黒白	白	白白	白白黒
白黒白白黒黒黒白	黒	黒黒	黒黒白
白黒白黒黒白白黒	白	白白	白白白
白黒黒白白白黒黒	黒	黒白	黒白白
・・・			
黒黒黒白黒白白白	黒	黒黒	黒黒黒

・符号化テーブルの数

表３の中で、4 ピースが反対色となるは 14 列であり、これを符号化配列として採用する。従って、符号化のパターン数 $P_E(5)^*$は、

$P_E(5)^* = 14$

となる。この場合、$P_E(5)^*$が $2^3=8$ を超えているので、最大 3 ビットの符号化が可能である。

この場合の符号化テーブルの数 $N_E(5)^*$は、

$$N_E(5)^* = \sum_{i=1}^{3} N_E(5,i)^*$$
$$= {}_{14}C_7 + {}_{14}C_3 * {}_{11}C_3 * {}_8C_3 * {}_5C_3 * 2 + {}_{14}P_8 * 6!$$
$$\fallingdotseq 8.7 * 10^{10}$$

となる。

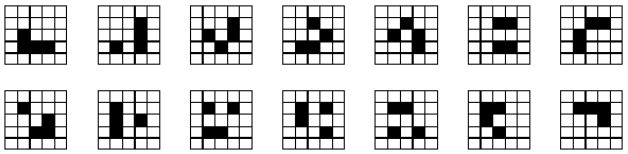


図６　拡張ハミング符号に対応したピース配置例

図６は、表３のビット配列を３ x ３構成の左上から順次右側及び下側に、‘１’を黒に‘０’を白として表記したものである。表３のビット配列の具体的配置を表４に示すようなテーブル形式で保持すれば、拡張ハミング符号の各ビットを任意の位置のピースに割当てることが可能となる。

この配置テーブルの数 N_Pは、8 個のピースの順列組合せとなり、

$N_P = 8! \fallingdotseq 4.0 \times 10^4$

となる。そこで、すべての組合せ N_Aは、

$N_A = N_E(5)^* * N_P \fallingdotseq 3.5 * 10^{15}$

となる。

４．２　ピースレベルの誤り検出

前節で述べたように、拡張ハミング符号を用いて色配置を行っているので、セル内のピースレベルの識別には、1 ビット誤りの訂正と 2 ビット誤りの検出の機能がある。

しかし、何らかの原因で上記の条件以外のピースレベルの誤りが発生する可能性は十分ある。そこで、誤りを検出できない条件およびその比率について検討する。

拡張ハミング符号の符号化の中で、すべてが‘０’または‘１’の場合を除いて配色しているので、セル色及び反対色はそれぞれ 4 ピースに固定されている。そこで、例えば、汚損などで反対色が増加または減少した場合には、誤りとして検出可能である。従って、検出できないのは、セル色と反対色について同数誤りが発生する場合である。しかし、1 ピースずつの誤りは、拡張ハミング符号の機能から検出可能である。

そこで、検出できないのは、

①２ピースずつの誤り
②３ピースずつの誤り
③４ピースずつの誤り（全ピース誤り）

である。

２ピースずつの誤り Ner(2)は、反対色の４ピースの中の２つを選択すると、セル色は自動的に定まるので、

$Ner(2) = {}_4C_2 = 6$

となる。同様に、

$Ner(3) = {}_4C_3 = 4$
$Ner(4) = {}_4C_4 = 1$

となる。そこで、誤りの場合の数の合計は、11 となる。全体の組み合わせは 2^8 であるので、未検出率 Rer は、

$Rer = 11/256 = 0.044$

となる。

しかし、このようなエラーの発生確率は、一様ではなく、近隣のピースがブロックで汚損される場合が多いので、未検出率 Rer は限りなく 0 に近いと考えられる。すなわち、セルレベルでの誤りはほぼ 100%検出可能である。

５．暗号化キー化

前節で、５ x ５構成の拡張ハミング符号を用いた場合の、ピースレベルの２次元コードの符号のための符号化テーブルとビット配置テーブルの説明を行った。この二つのテーブルと符号化ビット数の第三者への秘匿が、情報ハイディングの源泉である。そこで、これらは暗号化キーの役割を果たすことになるが、これらそのものを暗号化キーとすることも可能である。そこで、この符号化ビット数と二つのテーブルの暗号化キー化について検討する。

（１）符号化ビット数

符号化は 3 ビットまで可能であるので、これを示すためには、2 ビット必要である。

（２）符号化テーブル

符号化テーブルは、表５に示すように、１４個のビットの配列パターンに対する符号化を規定している。１４個の配列パターンは固定されているので、可変部のみを取り出

すと、多ビット符号化時の３ビットである。そこで、４２ビットで符号化テーブルが表現可能である。

（３）配置テーブル

配置テーブルは、表４に示すように８個のビット位置をそれぞれ規定するものである。その配置可能な位置は縦横それぞれ３つであるので、それぞれ２ビットで規定される。従って、３２ビットで配置テーブルが表現可能である

表４ ビットのピースへの配置テーブルの例

拡張ハミングコードビット位置	ピース位置	
	縦位置	横位置
1	1	3
2	2	3
…		
8	3	2

（３）全体の暗号化

表６に上記のビット配置を示す。このビット列が、暗号化キーそのものと見做すことができる。この 76 ビットを二次元コードの作成者と読取り者が共有することで、秘匿されたデータの受け渡しを可能とする。

表６ 暗号化キーの構造

符号化ビット数	符号化テーブル				配置テーブル			
	1列	2列	…	14列	1列	2列	…	8列
2	3	3	3	3	4	4	4	4
2ビット	42ビット				32ビット			
76ビット								

（４）公開領域化

表４、表５に示した例などをデフォルトの配置として、符号化側と復号側で共有すれば、上記の暗号化キーを必要としないので、ピースレベルの領域を公開領域とすることが可能である。公開領域とする場合には、記憶するデータ量の増加（４倍）と第６節で述べる誤り訂正率の増加（６８％）の特徴を活かした使い方となる。

６．二次元コードの符号化と復号

この節では、提案するセルの微細分割構造を有する二次元コードの符号化と復号の処理内容について述べる。

６．１ 符号化

ここでは、次の条件の場合の符号化について説明する。

ピース分割　　：５ｘ５のピース分割
符号化　　：3 ビット符号化
セルレベルデータ構造　：ＳＱＲＣ
ピースレベルデータ構造：ＳＱＲＣ

提案する二次元コードの符号化処理は、表３に示す 5 つのステップから構成される。セルレベルの誤り訂正処理は煩雑であり、本論文と直接関係しないので、省略する。

表７ 符号化処理

処理ステップ	処理内容
ステップ1	データの準備
ステップ2	セルレベルのデータ暗号化
ステップ3	セルレベルのデータ配置
ステップ4	ピースレベルの各層のデータ暗号化
ステップ5	ピースレベルの各層のデータ配置

これらの各ステップについて、以下説明する。

・ステップ１　データ準備

セルレベルとピースレベルの二次元コードの公開領域と非公開領域に収容するデータを準備する。すなわち、セルレベルを 2 セットとピースレベルを 6 セットの合計 8 セットのデータを準備する。

・ステップ２　セルレベルのデータの暗号化

セルレベルの非公開領域に収容するデータを与えられた暗号化キーＳによって、暗号化する。

・ステップ３　セルレベルのデータ配置

セルレベルの公開領域のデータ及び非公開領域の暗号化されたデータについて、与えられた二次元コードの条件（大きさ、誤り訂正レベル）に従って、各セルの白黒を決定し、配置する。これにより、各セル毎の 4 ビットの符号ブロック $\boldsymbol{u} = (u_0, u_1, u_2, u_3)$ について u_0 が定まる。

・ステップ４　ピースレベル各層のデータ暗号化

ピースレベルの 3 ビット符号化を行うので、仮想的に 3 面の二次元コードをピース領域に配置する。そこで、3 面のデータについて、非公開領域に配置するデータを各層の暗号化キーＰ１，Ｐ２によって暗号化する。

・ステップ５　ピースレベルの各層のデータ配置

各セルに配置するデータを各層から 1 ビットずつ選定し、3 ビットを構成し、この値を、予め選定された符号化テーブルから乱数によって、ピースの白黒配置を決定する。このピース配置によって、セル全体のピース構成を決定する。これにより、４ビットの符号化ブロック $\boldsymbol{u} = (u_0, u_1, u_2, u_3)$ について u_1, u_2, u_3 が定まる。このとき、セルが白の場合と黒の場合では、それぞれ反対色に切り替える。

これにより、二次元コードを全てのセルについて、ピース配置が決定されれば、二次元コードの全体の構造が決まり、印刷をすることが可能になる。

表８ 復号処理

処理ステップ	処理内容
ステップ1	各セルの白黒判定
ステップ2	各ビットレベルの白黒判定
ステップ3	セルレベルの公開領域の復号
ステップ4	セルレベルの非公開領域の暗号の復号
ステップ5	ピースレベルの復号
ステップ6	ピースレベルの公開領域の復号
ステップ7	ピースレベルの非公開領域の暗号の復号

６．２　復号

復号処理は、表８に示す 7 つのステップから構成される。

・ステップ１　各セルの白黒決定

撮像した画像から、セル画像を切り出し、セルの白黒を判定する。

これにより、各セル毎の 4 ビットの符号ブロック $\boldsymbol{u} = (u_0, u_1, u_2, u_3)$について u_0が定まる。

・ステップ２　各ピースレベルの白黒決定

撮像した画像から、セル画像を切り出し、セルの白黒判定に基づき、各ピースレベルでのセル色か反対色かを判定する。ピースの拡張ハミング符号による誤り検出および訂正処理も実施する。

・ステップ３　セルレベルの公開領域の復号

この処理は、通常のＳＱＲＣの公開領域の復号と同じ処理であり、公開領域のデータを復号する。

・ステップ４　セルレベルの非公開領域の復号

この処理も、通常のＳＱＲＣの非公開領域の復号処理と同じであり、非公開データについて、暗号の復号とデータの復号を行う。

これにより、４ビットの符号化ブロック $\boldsymbol{u} = (u_0, u_1, u_2, u_3)$について u_1, u_2, u_3が定まる。

・ステップ５　ピースレベルの復号

各セルについて、符号化テーブルに基づいて、3 ビットの白黒の復号を行う。

・ステップ６　ピースレベル公開領域の復号

仮想的に、3 面の二次元コードを作り出し、それぞれの公開領域のデータを復号する。

・ステップ７　ピースレベル非公開領域の復号

3 面の二次元コードについて、データの復号、暗号化データの復号を行い、データを抽出する。これらの処理により、セルレベル及びピースレベルの 8 セットのデータの復号が完了する。

７．誤り訂正能力の向上

通常の二次元コードでは、誤り訂正ができなかった場合には、どのセルが誤っているかは不明である。誤り訂正ができた場合のみ、誤っていたセルが特定される。

それに対して、本提案のピースレベルの判定色では、その構造に拡張ハミング符号を導入しているので、誤り検出がされず、または誤り訂正がなされれば、そのセルのピースレベルの判定色は正しい。一方、二次元コードが汚損されて多くのセルが正しい色を判定できなくなったとき、二次元コード全体の誤り訂正機能によって予め設定された誤り率以下であれば、誤り訂正が可能である。

ここで、セル内のピースレベルの配色を決定できない（誤り訂正できない）場合には、乱数によってその配色候補を定める事を考える。二次元コードの白と黒の数は概ね等しく配置される。すると、乱数で定めた不確定セルについては、概ねその半数が正しい結果となるので、確定セルと不確定セルの半数の和が正答となる。ここでの誤り率が誤り訂正の設定レベル以下であれば、誤り訂正が可能となる。

確定ピース率に対する正答率を表９に示す。二次元コードの一つであるＱＲコード[28]は、ＲＳ符号を用いた誤り訂正機能を具備しており、最大 30%の誤り訂正能力を有している。また、ピースレベルの二次元コードも、同じ構造を想定しているので、同じレベルの訂正能力を有している。

表９　乱数適用による訂正率の向上
（正答確率５０％時）

（%）

誤り率	確定正答率	正答期待値	誤り期待値	誤り訂正
50	50	75	25	○
60	40	70	30	○
70	30	65	35	×
80	20	60	40	×

表９より、誤り訂正レベルが 30%の場合には、誤り率が 60%のレベルの場合でも誤り訂正が可能となる。

乱数で決めたセルのピース色が 50%以上が誤り、復号できない場合には、再度乱数によって色の候補を決定すれば、何度かの試行の後に、誤り訂正が可能となる。正答期待値が 50%であるからである。

また、誤り率が 60%を超えても、正答率の分布によって、正答率が 50%を超える場合には復号できる可能性がある。確率 p の事象が n 回の試行中 m 回発生する確率 P(n,m)は、

$P(n,m) = {}_nC_m * p^m (1-p)^{n-m}$

である。そこで、m 回以上発生する確率 Qm は、

$Qm = \sum_{i=m}^{n} P(n,i)$

である。ここで想定している二次元コードは、25x25 程度である。この場合、誤り対象となるデータ部のセル数 n=400 の場合について、Qm を計算した結果を表１０に示す。p=0.5 とした。

表１０　正答確率の発生分布

正答確率（%）	発生確率	試行回数期待値
50以上	0.52	1.9
51以上	0.36	2.8
55以上	0.025	39.2
56以上	0.0093	107
57以上	0.0029	340
58以上	0.00080	13000
60以上	0.0000037	269000

この表から、2 回試行すれば正答率 50%以上が得られ、100 回程度試行すれば正答率 56%以上が得られることが判る。そこで、何回かの試行で正答率が 56%以上が得られるとして、その場合の誤り訂正の可能性を表１１に示す。これから、誤り率が 68%程度でも復号できる可能性があることが判る。

表１１の結果は、セル内の１ピース誤りの場合の訂正効果を含んでいないので、訂正能力全体としては、この表の値以上の訂正能力を有していると言える。

表１１　乱数適用による訂正率の向上
（正答確率５６％時）

（%）

誤り率	確定正答率	正答期待値	誤り期待値	誤り訂正
50	50	78	22	○
60	40	73.6	26.4	○
68	32	70	30	○
70	30	69.2	30.8	×
80	20	64	40	×

図７　二次元コードの汚損

また、上記の誤り訂正能力の向上は、ピースレベルだけでなく、セルレベルにも同様の効果がある。拡張ハミング符号で正しいと判定されたセルは、セルレベルの判定も正しく判定されるからである。

ピースの配置に拡張ハミング符号を導入することによって、ピースレベルとセルレベルの両者の二次元コード全体の誤り訂正能力を向上させることが可能となる。

図７に、３０％誤り訂正レベルでの、訂正不可の汚損と訂正可能な汚損の例を示す。

８．ＳＱＲＣとの併用

ＳＱＲＣは、本来セルレベルで公開部と非公開部を有する二次元コードである。このＳＱＲＣの構成をピースレベルに適用することで、より秘匿性の高い二次元コードとすることが可能である。

さらに、セルレベルのＳＱＲＣと連携すれば、さらに秘匿性は向上する。すなわち、ピースレベルの暗号化キーに一部をＳＱＲＣの非公開部に持てば、ＳＱＲＣの暗号化キーを有し、さらにそのピースレベルの暗号化キーを知らなければピースレベルのデータを復号することができない。

そこで、ＳＱＲＣを併用することにより、データの秘匿レベルを、表８に示すように、基本的に４段階に設定することが可能になる。すなわち、

レベル０：セルレベルの公開領域

誰でもが読取り可能。

レベル１：ピースレベルの公開領域およびセルレベルの非公開領域

ピースレベルの暗号化キーを知る者のみが読取り可能。

セルレベルの暗号化キーを知る者のみが読取り可能。

レベル２：ピースレベルの非公開領域

セルレベルとピースレベルの両方の暗号化キーを知る者のみが読取り可能。

レベル３：

セルレベルとピースレベルの二次元コードは互いに独立であるが、ピースレベルのＳＱＲＣの暗号化キーの一部を、セルレベルの非公開領域に記憶させることにより、セルレベルの暗号化キーと同時に知る必要があり、ピース符号化テーブルの暗号化キーと合わせて、3 つの暗号化キーが必要となる。

表１２　秘匿レベル

セルレベル		ピースレベル	
構造	秘匿レベル	構造	秘匿レベル
QR	0	QR	1
SQRC	1	QR	1
QR	0	SQRC	2
SQRC	1	SQRC	2
SQRC	1	SQRC*	3
SQRC	1	SQRC**	5

*暗号化キイの一部をセルレベル非公開領域に記憶
** 3ビット符号化時、各層の暗号化キイの一部を別層の非公開領域に記憶

９．微細構造の実現性

９．１　印刷密度

現在、業務用で用いられてプリンターの印刷密度は、600DPI 程度であり、高精細型では 1200DPI 程度である。これらのプリンターを用いて実用可能な大きさの微細構造を有する二次元コードの印刷可能性について検討する。

印刷物において、1 ピース当たり２ｘ２ドットあれば、充分識別可能である。２５ｘ２５セルの二次元コードを５ｘ５構成のピース分割する場合には、600DPI のプリンターで印刷すると、その一辺の長さＬは、

$$L = 25x5x2x25.4/600 \fallingdotseq 10.6(mm)$$

であり、通常の二次元コードの大きさと同じ程度である。また、1200DPI のプリンターを用いれば、５０ｘ５０セルの二次元コードでも、同じ大きさで印刷可能である。

９．２　撮像画素数

現在、二次元コードリーダに用いられている読み取り装置の撮像素子の画素数は、400 万画素程度である。この撮像素子の１／４の面積に二次元コードの画像を投影して撮影する場合、撮像素子が縦横同じ画素数とすると、各方向に 1000 画素がある。そこで、NxN 構成の二次元コードでは、１ピース当たりのピクセル数 Pp(N)は、

$$Pp(N) = 1000/(Nx5)$$

となる。そこで、50x50 構成の二次元コードの場合では、

$$Pp(50) = 4$$

となる。通常、2x2 画素のブロックが存在すれば、安定した識別が可能であるので、５０ｘ５０セルの二次元コードにおいても、５ｘ５構成のピースへの微細分割が充分可能である。

９．３　手振れとボケ

撮影時の手振れやピントのボケによる不鮮明な画像は、セルやピースの識別に大きな障害となる。

この問題に対して、有効な対策を講じる必要がある。例えば、微細構造の存在を識別したら、再試行を行い、その際にフラッシュを点灯し、撮像時間の短時間化を計ること等が考えられる。

一方、上記の対応などで鮮明な画像を得る事ができれば、手振れを活かして、微妙に位置の異なる複数の画像を撮像し、それを組合せて高精細化する超解像の手法[11]を用いることが可能になり、さらに微細な分割が期待できる。

１０．終りに

本論文では、二次元コードについて、セルを分割して、ピース構造を構成し、それらを符号化してデータ記憶することによって、情報ハイディングを行い、秘匿性を向上させ、且つ通常の二次元コードと互換性を有する公開領域を有する二次元コードを提案した。

提案した二次元コードでは、微細なピースを識別の対象とするため、精密な画像が必要とされるが、汚損などの二次元コードそのものの障害に加えて、撮像時の手ぶれなどでの画像の不鮮明化などの障害がある。これらの障害の符号化レベルの対応策について、今後検討していきたい。

また、今後実際に印字サンプルを作成し、読取りを行う実験を予定している。

参考文献

[1] 寺浦 信之、ＲＦＩＤの源流　二次元コード、食品包装　２００９年２月号　p42−p46

[2] 寺浦 信之、多層式光学的情報媒体による二次元コードの情報ハイディング、CSS2012, pp. 211-216

[3] 長屋 隆之、高速読取り２次元コード［ＱＲコード］の開発、情報処理学会全国大会講演論文集 第 52 回平成 8 年前期(2), 253-254, 1996

[4] 小林 哲二、二次元コードのセキュリティ向上と応用、情報科学技術フォーラム一般講演論文集 2002(4), 225-226, 2002

[5] 苅田 光一郎、ＱＲコードへの電子透かし実装に関する研究、電子情報通信学会技術研究報告、EMM, マルチメディア情報ハイディング・エンリッチメント、 111(209), 7-10, 2011

[6] 小野 智司、電子透かしを用いたカラー二次元コードの複製検知、電子情報通信学会論文誌. D, 情報・システム J94-D(12), 1971-1974, 2011

[7] 原 昌弘, ‘二次元コードの生成方法およびその読取り装置, 特開 2008-299422

[8] 遠藤 祐介、高密度情報化を可能にするＱＲコード符号化方式について、情報科学技術フォーラム講演論文集 9(4), 151-156, 2010

[9] 助川 修司、ＱＲコードの多色化による 2 次元コードの大容量化について、情報処理学会全国大会講演論文集 第 70 回平成 20 年(4), 845-846, 2008

[10] 寺田 遼平、カラー二次元コードを高解像度化するための認識アルゴリズムの実現と評価、電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス 108(444), 55-60, 2009-02-23

[11] 加藤 祐二、2 値パターン拘束と超解像度を組み合わせた低解像度ＱＲコード認識、電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解 、110(187), 63-68, 2010

[12] 齋藤 圭輔、ＱＲコードの誤り訂正能力について : 限界距離以上の誤り訂正、および消失同時誤り訂正によるデータ復元について、電子情報通信学会技術研究報告、ICM, 情報通信マネジメント、110(374), 45-50, 2011

[13] 佐藤 創、ＱＲコードの符号化・復号アルゴリズム解説、専修大学情報科学研究所所報 76, 37-52, 2011

[14] Adams, G., Simske, S., Pollard, S.. 2D Barcode Sub-Coding Density Limits. NIP27: 27th International Conference on Digital Printing Technologies and Digital Fabrication, October 2-6, 2011, Minneapolis, Minnesota, USA, 696-699.

[15] J-J Shen, P-W Hsu, “A Fragile Associative Watermarking on 2D Barcode for Data Authentication,” International Journal of Network Security, vol.7, no.3, pp. 301-309.

[16] L Li, R-L Wang, C-C Chang, “A Digital Watermark Algorithm for QR Code,” IJIIP: International Journal of Intelligent Information Processing, vol. 2, no. 2, pp. 29-36, 2011.

[17] ISO/IEC 18004:2006 Information technology -- Automatic identification and data capture techniques -- QR Code 2005 bar code symbology specification.

[18] ISO/IEC 16022:2006 Information technology -- Automatic identification and data capture techniques -- Data Matrix bar code symbology specification.