K-045

高密度二次元コードのブロック誤りとランダム誤りに対応する二重誤り訂正システムの検討

Discussion of the double error-correction system corresponding to the block error and random error of a high-density two dimensional code

寺浦　信之 1)　井上　徹 2)　櫻井　幸一 3)
Nobuyuki Teraura　Toru Inoue　Kouichi Sakurai

1. まえがき

(1) 背景

現在用いられている二次元コード[1][2]は，バーコードの記憶容量が少なく，ＩＤ番号のみを記憶し，ＩＤ番号に対応するデータはサーバーに記憶して参照しているのに対して，具体的なデータを記憶するデータキャリアとして開発されたものである．そして，さらに大容量のデータが記憶できれば，その応用はさらに広がると考えられる．

大容量を実現するためには，単純に大きな二次元コードを用いることでも実現できる．しかし，ここでは同じ大きさの二次元コードでより大きな容量を記憶すること，すなわちデータの記録密度を大きくすることを検討する．

(2) 既存の研究

データの記録密度を大きくするための手法として，セルを多値化する研究がされており，セルを多色化する手法[3-6]及びセルを分割する手法[7]がある．我々は，セルを微細化した上で，多値化する多色多値二次元コード（高密度二次元コード）を提案している[8].

(3)　課題

データ密度を大きくするために，高密度二次元コードを縮小していくに従って，色識別の誤識別率が高まる．誤識別率が高密度二次元コードに組み込まれている RS（リードソロモン）符号[9]に基づく誤り訂正機能の限界を超えると，復号できなくなり，読取不可となる．RS 符号は，データコードシンボル単位の誤り訂正を行うので，ブロック誤りについては有効な訂正システムである．一方，データコードシンボルに 1 ビットづつの誤りが発生するランダム誤りについては，訂正性能力は低い．

高密度二次元コードで誤りが発生しはじめる大きさでの誤りは後述するようにランダム誤りである．そこで，ランダム誤りに対応する新たな誤り訂正の仕組みを導入することにより，より小さなセルサイズの二次元コードを読取り可能にし，記憶できるデータ密度の向上を図る．

(4)　論文の構成

以下に，提案する高密度二次元コードの概要を説明し，スマートフォンを用いた読取実験の結果を示す．その読取誤り原因がランダム誤りであることを検証し，ランダム誤りに対する訂正システムとして，縦横パリティ方式と拡張ゴレイ符号方式を検討する．そして，上記二つの方式を先の読取実験の結果について適用することにより，訂正の可能性についてシミュレーションを行う．

1)　テララコード研究所, Terrara Code Research Institute
九州大学社会人博士課程
2)　広島修道大学, Hirosima Shudo University
3)　九州大学, kyushu University

2．高密度二次元コードの構成

（1）セルの構造

既存の二次元コードとの互換性を維持しつつ，記憶領域を追加するために，二次元コードの基本単位であるセルの多値化を行う必要がある．多値化の手法として，多色化と多領域化が知られている．多領域化は，セルを複数の領域（サブセル）に分割し，それぞれに独立した情報を与える方式である．ここでは，互換性と大容量化のために，多色化と多領域化を併用する．そして，二次元コードとして，QR コード[1]を対象とする．この事例を図 1 左側に示す．QR コードのセルが白または黒の 1 ビットを表現しているが，それを多値化するために，図 1 右側に示すセルの構造を採用する．

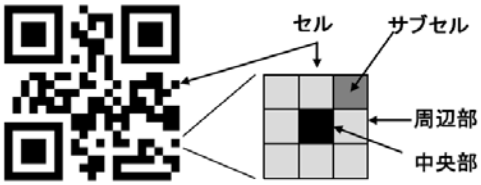


図 1　QR コードとセルの分割

（2）サブセルの多色化

4 節で述べるように，距離尺度として RGB 値間のユークリッド距離を用いる．この距離尺度では，RGB の三次元空間で相互に最も離れた色セットが識別が最も容易となる．そこで，相互に最も離れた位置は RGB 空間の立方体の端部であり，当該位置にある色を選択する．

選択した各色の RGB の具体値を表 1 に示す．表 1 に，各色の輝度を示したが，輝度(Y)と RGB 値の変換式は，ITU-R BT.601[10]で規定されている次式を用いた．

$$Y = (0.299R + 0.587G + 0.114B) / 255 \quad (1)$$

輝度の値を基に，各色を白または黒グループに分類した．

表 1　カラー8 色の選択と符合化データ

色群	色コード	RGB			輝度	色	符号化データ
		R	G	B			
白グループ	000	255	255	255	1		00
	001	255	255	0	0.89		01
	010	0	255	255	0.70		10
	011	0	255	0	0.59		11
黒グループ	100	255	0	255	0.41		00
	101	255	0	0	0.30		01
	110	0	0	255	0.11		10
	111	0	0	0	0		11

(3)互換性の実現

既存のスマートフォンの二次元コードの読取りソフトウェアは，セルの切り出し後，各セルの中央部の画素の色，すなわち白色または黒色かを判別している．そこで，互換部のデータの識別には，周辺部の寄与は小さい．そこで，中央部を互換領域に割り当てることにより，互換性を実現する．

しかし，周辺部の色が，互換部と反対の色である場合には，誤って識別される可能性が高くなる[11]．すなわち，互換部が白の場合において，周辺部が黒の場合には，影響が大きくなる．これは，手振れやピントボケ，ＯＳによる画像処理が原因として考えられる．そこで，周辺部の色を白グループ色と黒グループ色に分類し，互換部が白色の場合には，周辺部を白グループ色を割り当て，黒色の場合には黒グループ色を割り当てる．ただし，これらのカラー色は，印刷される場合には，印刷時のインクの発色や経時劣化による変色で，白または黒グループの範囲に止まらない可能性がある．そこで，白または黒グループ色の周辺部への割り当ては，中央部の識別に与える影響を低減するための補助的な手法と言える．

(4)符号化

サブセルは，セル色によって，白グループまたは黒グループの 4 色によって符号化される．

4 色による符号化では 2 ビットを表現できる．図 2 に示すように，特定の位置のサブセルを，二つの白黒の同一サイズの二次元コードの同一位置のセルの値を表 1 の符号化データによって色を選択することによって符号化する．例えば，互換部が黒であるセルにおいて，サブセルが両方共に白（0）の場合には 00 の 2 ビットを表現し，赤色に符号化される．

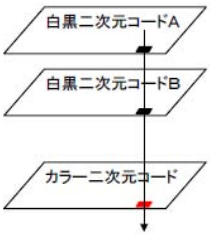


図 2 二次元コードの色符号化

(5) 仮想的な積層構造

選択した 8 色について，白グループと黒グループの色はそれぞれ 4 色であるので，新たに 2 ビットを表現できる．提案する構成では，符号化領域が 8 個あるので，セル周辺部は 16 ビットを表現する．そこで，提案する構造の二次元コードは，従来と同容量の互換領域とその 16 倍の容量の追加領域からなる二次元コードを表現する．

この構成は，一つのセルが 17 ビットを表現しているので，図 3 に示すように，白黒の既存の 2 次元コードが 17 層重なっているのと同等である．

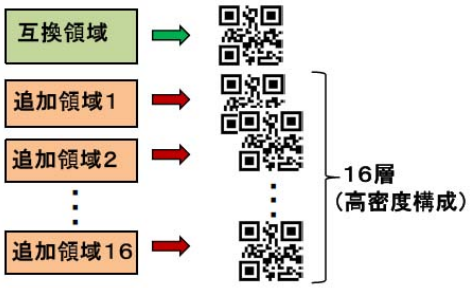


図 3 仮想的な積層構造

(6) 色識別手法

カラー色を用いた二次元コードを紙に印刷する場合には，印刷に用いるインクやトナーに依って，元の RGB 値から離れた色になったり，それらをスマートフォンで読取る場合には，用いられる撮像素子の特性によって，入力される RGB 値がさらに変化することが知られている[6]．さらに，印刷された色が経時劣化によって，変化することが想定できる．

そこで，本提案では，これらの影響を回避できる基準色との比較方式を採用する．比較方式（パレット方式）は上記のインクによる発色の差異や経時劣化について，基準色とセル色が同様に差異が発生し，経時劣化するので，それらの影響を回避可能である．

基準色（パレット色）を二次元コードの定められた位置に配置する．図 4 に示すように，白及び黒グループのパレット色は，それぞれファインダーパターン[1]の白部または黒部に配置する．その比較対象の色群をここではパレットと呼ぶ．QR コードでは，定められた位置にファインダーパターンを有しているので，そこへパレットを配置する．また，これらのパレット色が汚れなどによって正しい色を表現できなくなる可能性があるので，三つあるファインダーパターンのすべてにパレットを設定する．

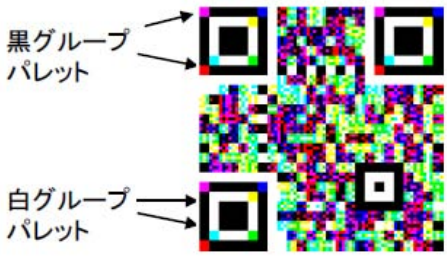


図 4　パレット色の設定

3. 識別方式

各サブセルの色を識別する識別処理は，比較対象であるパレット色とサブセル色を準備するデータ抽出処理と比較処理から構成される．

(1) **データ抽出処理**

色識別処理での比較する対象は，パレット色とサブセル色である．パレット色はすべてのサブセルの比較に共通である．パレット色は，図 5 左図に示すように，パレットを構成するセルを 9 個の領域に分割し，中央部を除く周辺部 8 個の RGB 値の平均値とした．比較はサブセル色とパレット色（白または黒のグループ色の 4 色）と行う．また，パレットは，図 4 に示すように，各ファインダーパターンの中に３セット設定している．3 セットに含まれる色を独立

した色として扱い，仮想的に 12 色と比較し，最も距離の小さいパレット色を選択する．

データコードシンボルを構成するデータ部のサブセルのサンプル値は，図 5 に示すようにサブセルの中心の RGB 値とする．

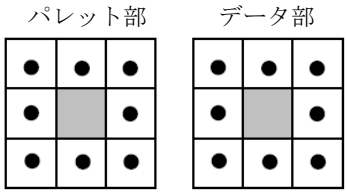


図 5　データ抽出位置

(2) **比較処理**

比較処理は，データ抽出処理で作成したパレット色 12 色とサブセル色の各 RGB 値を比較する．パレット色の RGB データ Cp を，

$$Cp = (Ri, Gi, Bi) \quad (2)$$

と表現する．

ここで，i は選択候補であるパレット色 12 色の指標である．Ri ,Gi, Bi は各パレット色の RGB 値であり，0 から 255 の間の値をとる．

同様に，サブセル色の RGB データ Cc は，

$$Cc = (Rc, Gc, Bc) \quad (3)$$

で表現する．i 番目のパレット色とサブセル色の RGB 値のユークリッド距離 Di は

$$Di = \sqrt{(Rc - Ri)2 + (Gc - Gi)2 + (Bc - Bi)2} \quad (4)$$

となる．

Di が最小値となるパレット色をサブセル色と判定する．

4. スマートフォンを用いた識別試験

第二節で概説した多色多値二次元コードについて，スマートフォンを用いた識別試験の内容及びその結果について述べる．

試験に用いた高密度の多色多値二次元コードを図 6 に示す．また，同じ互換部の白黒の二次元コードを比較のために示す．二次元コードの作成条件は下記を用いた．

バージョン：バージョン 2（25 x 25 セル）
誤り訂正　：レベルH

読取りは，下記の条件で実施した．

スマートフォン：GALAXY Note 2　(SAMSUNG 製)
照明　　：白色蛍光灯
二次元コードとの距離：　55mm に固定
焦点合わせ　　：スマフォによる自動焦点

また，識別の対象画像は，画面への表示画像(720x1280)を用いた．

表 2　高密度二次元コードの収容データ

互換領域	追加領域
吾輩は猫であ	吾輩は猫である。名前はまだ無い。 どこで生れたかとんと見当がつかぬ。何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。吾輩はここで始めて人間というものを見た。しかもあとで聞くとそれは書生と

高密度二次元コード　　白黒二次元コード

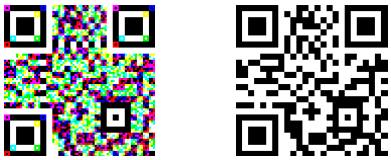


図 6　検証に用いた二次元コードの例

4.1　セルレベルの識別率

識別試験は，一辺の長さが 40mm から 10mm までの 7 種のサンプルについて，それぞれ 5 回づつ読取を行った．試験に用いたバージョン 2（25 x 25）では、データコードシンボル 44 個から構成されている．これらの領域（44 X 8 x 8）のサブセルについての識別結果を図 7 に示す．

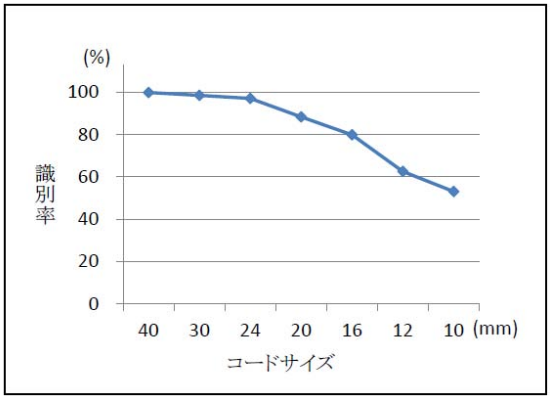
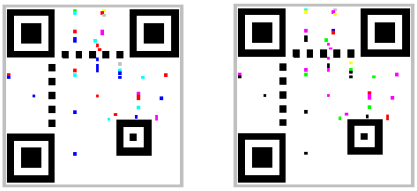


図 7　高密度構成の識別結果

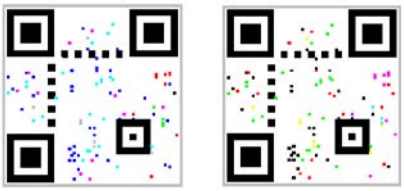
コードサイズが 30mm までは，ほぼ 100%の識別率であるが，24mm,20mm では 97.0%,88.4%と低下し，16mm よりも小さくなると識別率が大幅に低下してくる．

誤りの発生位置と誤った色を図 8 に示す．左図が誤識別色，右図が元色である．白はグレー色で表示した．コードサイズ 20mm の場合には，誤識別は全ての色で発生していた．

コードサイズ 30mm



コードサイズ 24mm



コードサイズ 20mm

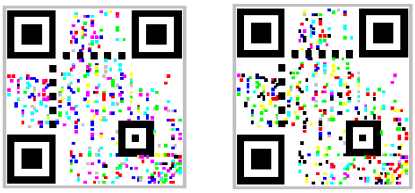
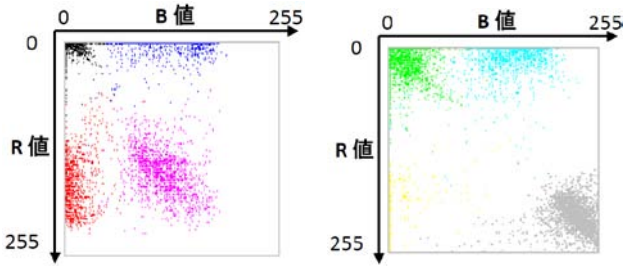


図 8　高密度構成の誤りセル分布

次に，それぞれの色の RGB 値の R と B の値の分布図を図 9 に示す．この図は，5 回の読取のすべての計測値をプロットしたものである．

コードサイズ 30mm の色分布



コードサイズ 20mm の色分布

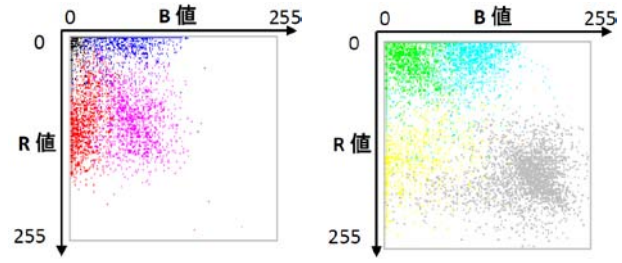


図 9　高密度構成のサンプリング色の分布

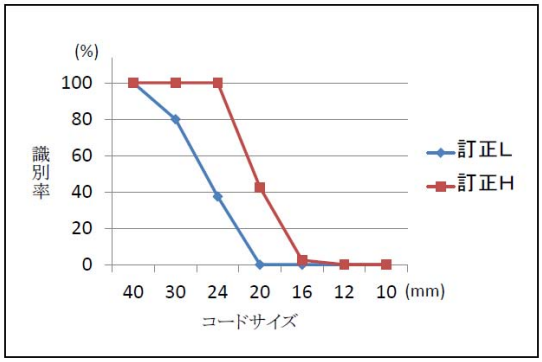
コードサイズが 30mm の場合には，それぞれの色の RGB 値の分布範囲は分離しており，各色を識別可能である．しかし，コードサイズが 20mm になると，色の RGB 空間上の縮退が大きく発生しており，各色の RGB 値の分布範囲が重なってきており，重なり部分は識別不可能である．

4.2　コードレベルの識別率

二次元コードでは，誤り訂正機能を具備しており，定められた誤り率以下であれば，誤り訂正が可能であり，正しく復号可能である[1]．検証に用いたサイズでは，総データコードシンボル数が，44 個である．誤り訂正レベルが，L，M，Q，Hについて，誤りのあるデータコードシンボルが，それぞれ 4 個，8 個，11 個，14 個以下であれば，すべての誤りを訂正し，コードレベルで正しく読取が可能である．

図 7 の 8 個のサブセルの色の誤識別を，それぞれ復号後の奇数層と偶数層の二つの白黒の二次元コードの誤りに展開した．そして，それらの誤りセルがどのデータコードシンボルに属するかを計測し，誤りセルの存在するデータコードシンボル数をカウントした．この誤りコードシンボル数と各誤り訂正レベルの訂正可能な数と比較した。訂正可能な数より小さい場合には、誤りデータコードシンボルの誤りデータを訂正可能と判断し、読取り可能性をシミュレーションした．訂正可能な場合を 100%，訂正不可能な場合を 0%としてその平均値を計算した．奇数層及び偶数層のコードレベルの識別率を図 10 に示す．　奇数層と偶数層は，表 1 の符号化データの第 1，第 2 ビットで構成される仮想的な白黒の二次元コードである．

奇数層のコードレベルの識別率



偶数層のコードレベルの識別率

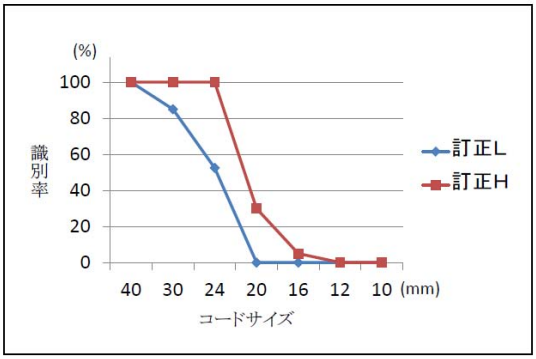


図 10　高密度構成のコードレベルの識別率

図 10 に示す奇数層と偶数層の識別率に差が出るのは，識別した誤り色によって奇数層と偶数層に及ぼす影響が異なるからである．例えば，白（符号化データが 00）を黄色（01）や青（10）と誤識別すると，それぞれ偶数層，奇数層のみが誤り，奇数層と偶数層は正しい．また，緑（11）と誤識別すると奇数層と偶数層の両方が誤る．

図 10 の結果から，誤り訂正レベルが H レベルでは 24mm，L レベルでは 40mm のコードサイズまで読取可能である．

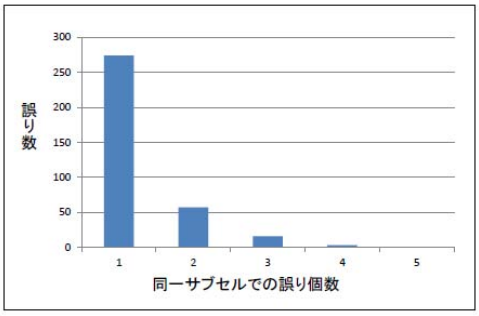
5　誤りの分析

（1）同一サブセルの誤り数

図 10 に示したサブセルレベルの識別結果は 5 回の読取りの結果である．5 回の読取で，同一サブセルで発生した識別誤りの個数を，コードサイズ 24mm 及び 20mm について図 11 に示す．

誤りの中で，1 回または 2 回の誤りが発生したサブセルの個数は，コードサイズ 24mm では 274 個と 54 個であり，コードサイズ 20mm では 648 個と 295 個であり，両者の合計は，それぞれ全体の 94.6%と 81.9%を占めている．従って，これらの誤りは，特定のサブセルに何らかの原因があって，同じサブセルで繰り返し発生しているのではなく，読取試行毎に誤り箇所が異なっており，ランダムに分散して発生している．そこで，これらの誤りは，ランダム誤りであると言える．

コードサイズ　24mm



コードサイズ　20mm

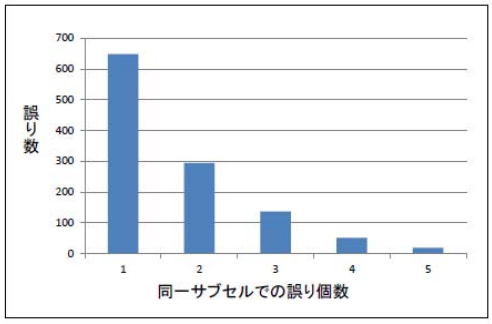


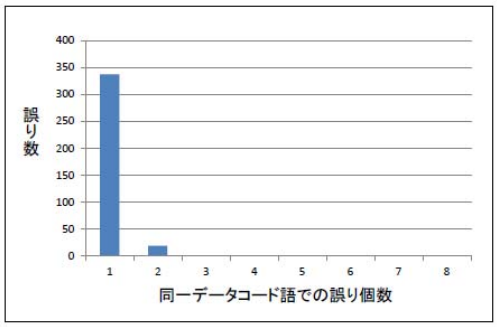
図 11　同一サブセルの誤り数

（2）ブロック誤り対応単位（ＲＳ符号のデータコードシンボル）の誤り数

4.2 節で述べた識別検証結果において，RS 符号の 44 個のデータコードシンボルにおいて，発生した誤り個数をコードサイズ 24mm 及び 20mm について図 12 に示す．識別対象のデータコードシンボルの個数は、16 層分について 5 回の読取を行っているので、3520 個である．

誤りの中で，データコードシンボルの中に 1 個のサブセルに誤りが発生したデータコードシンボルの個数は，コードサイズ 24mm では総誤りデータコードシンボル数 356 個中 337 個であり，コードサイズ 20mm では 1139 個中 814 個であり，それぞれ全体の 94.7%と 71.5%を占めている．

コードサイズ　24mm



コードサイズ　20mm

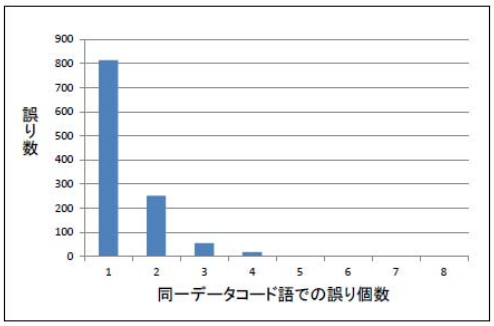


図 12　データコードシンボルの誤り数

この結果からも，誤りはランダムに発生しており，同一のデータコードシンボルに集中して発生していないことが判る．

また，コードサイズ 24mm,20mm では，大半のデータコードシンボル中の誤りは 1 ビットであり，データコードシンボルを構成する 8 ビット中の 1 ビットを訂正できれば，ブロック誤り用の訂正符号である RS 符号との連携により，多くの誤りが訂正可能となる．

そこで，16 層の中の幾つかの層を誤り訂正に割り当てることにより，より小さなサイズの二次元コードの読取りを可能とし，結果的により大きなデータ密度を実現できる可能性がある．

6 ランダム誤り対応の訂正システムの導入

ランダム誤り対応する訂正システムとして，縦横パリティ方式と拡張ゴレイ符号[9]方式について検討した．

6.1 縦横パリティ方式

縦横パリティ方式は，図 13 に示すように，RS 符号のデータコードシンボルを単位とする訂正システムである．一つのセルは 8 個のサブセルを有し，16 個の仮想的な白黒の二次元コードを表現する．そこで，奇数層と偶数層に分割し，8 個の層を単位とする縦横パリティデータセットを設定する．8 個の層の内，6 個がデータ層であり，2 個がパリティ層である．7 列目の層は 6 個の層の横パリティを表す．また，8 列目の層は各層毎のデータコードシンボルのパリティを表す．

この方式では，64 ビットの中の 1 ビットの訂正が可能である．また，この縦横パリティ方式はブロック誤り訂正を担当する RS 符号による誤り訂正の補助システムであるので，まず，従来通り RS 符号誤り訂正を試みる．8 層について，復号が可能であった層については，誤りビットの訂正を行う．すべての層について復号が不可能であった場合に，縦横パリティ誤り訂正を実施し，訂正可能であった場合には，訂正し，再び RS 符号の訂正を試みる．これらによって，訂正が不可能になるまで訂正処理を交互に繰り返す．

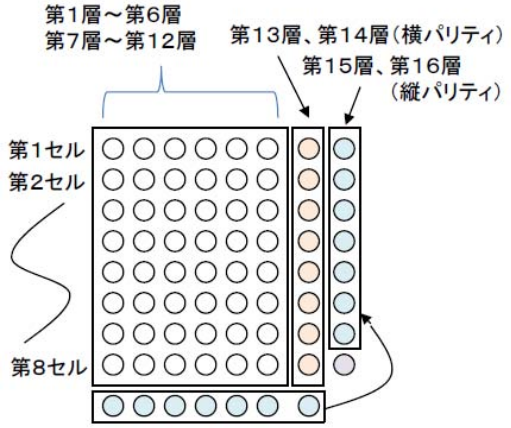


図 13　縦横パリティ方式のデータ構成図

6.2 拡張ゴレイ符号方式

二つ目のランダム誤りに対応する訂正システムとして，拡張ゴレイ符号方式を検討する．拡張ゴレイ符号としては 12 ビットのデータを 24 ビットの符号語に符号化 (24,12,8) する．図 14 にデータ構成図を示す．

拡張ゴレイ符号方式では，24 ビットを基本単位とするので，上記の縦横パリティ方式で用いた RS 符号の単位を 3 つ用いて，二つの拡張ゴレイ符号訂正データコードシンボルを構成する．

この方式では，24 ビット中 3 ビットの誤り訂正が可能であり，データ容量の半数を検査シンボル語に費やすが，訂正能力が高いので，より小さなコードサイズまで復号が可能である．

この場合においても，RS 符号の誤り訂正の補助システムであるので，RS 符号の復号をまず実行し，その後拡張ゴレイ符号の復号を実行する．これらの訂正処理を復号が不可能になるまで交互に繰り返す．

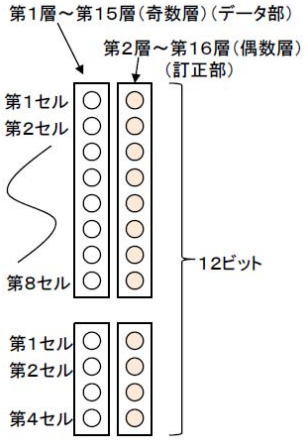


図 14　拡張ゴレイ符号方式のデータ構成図

7. 処理ステップ

ここで，本提案の高密度二次元コードの符号化と復号の処理について，拡張ゴレイ符号方式を例に述べる．

符号化処理

ステップ 1：データの準備

互換領域に収容するデータ d0 及び追加領域に収容するデータを各層に分配して得た各層に収容するデータ d1..d8 からなる収容データ D = (d0,d1..d8)を準備する．

ステップ 2：拡張ゴレイ符号検査シンボルの作成

収容データ D = (d1..d8)を基に、検査シンボル Dc = (d9..d16)を生成し、D を奇数層に、Dc を偶数層に配置する。

ステップ 3：各層の白黒二次元コード化

各層の全収容データ Da= (d0,d1..d16)を白黒の二次元コード化し，誤り訂正データを含めた二次元コードの白黒データ U = (u0,u1..u16)を得る．

ステップ 4：可変領域のセル色の決定

互換領域の各セルの白黒データ u0 に基づき，表 1 の白グループまたは黒グループの符号化データを用いて，追加領域の各セルの白黒データ *u1..u16* を符号化し，周辺部の 8 個のサブセルの色を決定する．

ステップ 5：固定領域のセル色の決定

パレット色を割り当てられたパレット部に設定する．それ以外のファインダーパターンなどの固定領域は黒または白に設定する．

復号処理

ステップ 1：画像入力と画像抽出

撮像装置によって，二次元コードを含む画像を撮像し，二次元コードに含まれるファインダーパターンを基に二次元コードを検出し，二次元コードの画像を抽出する．

ステップ 2：互換部の識別
撮像した二次元コードを白黒の二次元コードとして識別し，互換領域のセルが白か黒のデータ u0 を得る．
ステップ 3：パレット色の抽出
ステップ 2 で切り出したセルについて，パレット色が格納されているセルから，パレット色のデータを取得する．
ステップ 4：可変領域のサブセル色の識別
二次元コードの可変領域のサブセルについて，12 個のパレット色との距離を(4)式により計算し，最小の距離の色を当該セルの色として選択する．
ステップ 5：各層の白黒二次元コードに復号
各セルの互換部の白または黒の色に従い符号化テーブルによって復号し，二次元コードの白黒データ U=(u0, *u1.. u16*)を得る．

ステップ 6：RS 符号による誤り訂正
U=(u0, *u1.. u16*)から RS 符号による誤り訂正を行い、成功した場合には、D = (d0, *d1.. d16*)を得る．
ステップ 7：ランダム誤り訂正
ステップ 6 での RS 符号誤り訂正において，訂正できなかった場合には，拡張ゴレイ符号誤り訂正を行う．拡張ゴレイ符号誤り訂正処理で訂正できたシンボルがある場合には，再度ステップ 6 の RS 符号訂正処理に戻る．
ステップ 7：白黒二次元コードの復号
ステップ 6 または 7 で誤り訂正に成功した場合には，各層に収納されたデータ D = (d0, d1.. d8)を得る．

8. シミュレーション

縦横パリティ方式及び拡張ゴレイ符号方式の誤り訂正システムを，既存のＲＳ符号を用いた誤り訂正システムに付加した場合の誤り訂正率のシミュレーションを行った．RS 符号による誤り訂正は，訂正レベルＬ及び訂正レベルH[1]について実施した．シミュレーションに用いたデータは，図 7 に示した実験結果と同じであり，実際にスマートホンで測定したデータである．

8.1 縦横パリティ方式のシミュレーション

初めに，縦横パリティ方式と RS 符号方式を併用した場合のシミュレーションの方法とその結果について述べる
シミュレーションは次のようにして実施した．読取り時に，生成時のパターンと比較し，誤った色識別をしたサブセルを特定し，さらに白黒の二次元コードに復号したときの誤り位置に誤りフラッグを立てて記録しておく．そして，縦横パリティ方式の訂正の単位である図 13 に示す 64 ビットの領域において，誤りビット数が 1 ビットである場合には，訂正可能であるとして，当該誤りフラッグをクリアーする．また，RS 符号については，その訂正レベルに対応した訂正能力の範囲内にある場合には，訂正可能であるとして，対応する縦横パリティののビット列の対応部分について誤りフラッグをクリアーする．
誤り訂正は，まず RS 符号による訂正を実施し，そこで復号できなかった場合には，縦横パリティ方式による誤り訂正を行った．これらの処理は，訂正が行えなくなるまで，交互に実施した．
コードレベルのシミュレーション結果を，図 15 に示す．

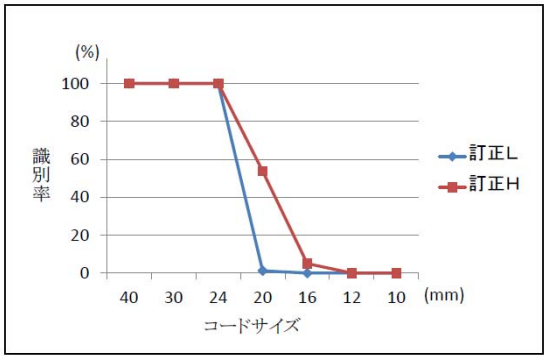


図 15　縦横パリティ方式を付加した識別率

訂正レベルＬでは，コードサイズが 30mm,24mm の場合には従来 80%,37.5%であったが，縦横パリティ方式を付加した場合には，それぞれ 100%,100%に向上した．また，訂正レベルＨでは，コードサイズが 20mm,16mm の場合には従来 42.5%, 2.5%であったが，縦横パリティ方式を付加した場合には，それぞれ 55%,5%に向上した．
訂正レベルＨは，検査シンボルをデータコードシンボルの２倍程度設定しており，縦横パリティ方式を付加しても大きな効果を得られなかった．しかし，訂正レベルＬは訂正レベルＨの６分の１の訂正の力を有していないため，縦横パリティ方式の付加で一定の効果を発揮している．
また，縦横パリティ方式では，64 ビット中の 1 ビットの訂正能力を有するに止まるので，識別誤り率が大きい場合には効果を発揮しえない．しかし，ランダム誤りが発生し始めるサイズ，すなわち誤識別率が低い場合には，効果がある．僅かなデータコードシンボルの損傷で RS 符号による誤り訂正が不可能となる場合に適用できる．

8.2 拡張ゴレイ符号方式のシミュレーション

拡張ゴレイ符号と RS 符号の各訂正レベルの組み合わせて，コードレベルのシミュレーションの方法と結果について，述べる．
前記の縦横パリティ方式と同様に，読取時に誤った識別をした位置に誤りフラッグを立てて記録しておく．そして，拡張ゴレイ符号方式の訂正の単位である図 14 に示す 24 ビットの領域において，誤りビット数が 3 ビット以下である場合には，訂正可能であるとして，当該誤りフラッグをクリアーする．RS 符号による誤り訂正は，縦横パリティ方式と同様である．

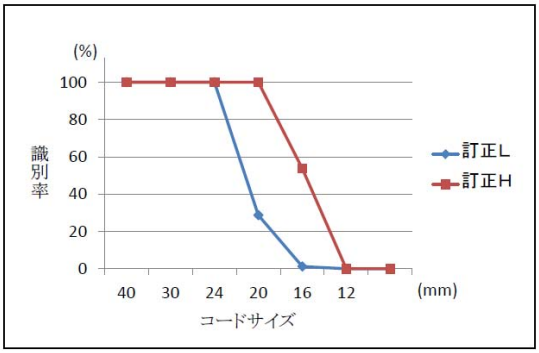


図 16　拡張ゴレイ符号方式を付加した識別率

誤り訂正は，まず RS 符号による訂正を実施し，そこで復号できなかった場合には，拡張ゴレイ符号方式による誤り訂正を行った．これらの処理は，訂正が行えなくなるまで，交互に実施した．

シミュレーションの結果を図 16 に示す．

訂正レベルＬでは，コードサイズが 24mm,20mm の場合には従来それぞれ 37.5%, 0%であったが，縦横パリティ方式を付加した場合には，それぞれ 100%,28.8%に向上した．また，訂正レベルＨでは，コードサイズが 20mm, 16mm の場合には従来それぞれ 42.5%,2.5%であったが，拡張ゴレイ符号方式を付加した場合には，それぞれ 100%, 53.8%に向上した．

また，拡張ゴレイ符号方式では，24 ビット中の 3 ビットの訂正能力を有するので，縦横パリティ方式と比較して大きな識別率の向上が見られた．これは，拡張ゴレイ符号方式によるランダム誤りが訂正されると，RS 符号の誤り訂正の範囲となるデータコードシンボルの誤り数となり，訂正可能となる層が出現する．すると，拡張ゴレイ符号の誤り訂正の範囲となる訂正対象が出現する．これらが複数回交互に出現することにより識別率を向上させている．

9. おわりに

多色多値二次元コードの読取に際して発生するランダム誤りについて，実際の読取データを用いてシミュレーションを実施した．ランダム誤りに対応する誤り訂正システムとして，縦横パリティ方式及び拡張ゴレイ方式を適用しその効果を確認した．

縦横パリティ方式は，RS 符号の訂正レベルが低い場合及び識別率が比較的高い場合に効果があり，拡張ゴレイ符号方式は，識別率が低い場合及び RS 符号の訂正レベルを高く設定する場合に訂正効果が増大する結果が得られた．

参考文献

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

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

[3] H. Kato, K. Tan, D. Chai, Development Of A Novel Finder Pattern For Effective Color 2D Barcode Detection, Proceedings of IEEE International Symposium on Parallel and Distributed Processing with Applications. ISPA '08. (pp. 1006-1013). Sydney, Australia. IEEE Computer Society,2008

[4] 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

[5] 寺田 遼平，藤本 敬介，中山 泰一，カラー二次元コードを高解像化するための認識アルゴリズムの実現と評価，信学技報，SS2008-57,2009-3

[6] 遠藤祐介,廣友雅徳，佐治勇樹，渡辺優平，森井昌克，多値二次元コードにおける高階調度認識アルゴリズムの提案，電子情報通信学会論文誌 D Vol.J95-D No.11 PP.1935-1943

[7] 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.

[8] 寺浦 信之，櫻井 幸一，秘匿領域を有する高密度二次元コードの互換性と識別性に関するスマートフォン実装による評価，2015 年暗号と情報セキュリティシンポジウム(SCIS2015), 1B2-2, 2015

[9] 西村芳一, データの符号化技術と誤り訂正の基礎，ＣＱ出版社，2010.8.

[10] http://www.itu.int/rec/R-REC-BT.601/e.

[11] 寺浦 信之，櫻井 幸一，‘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’，第 11 回情報科学技術フォーラム (FIT2012), 571-578, 2012