

3次元カラーバーコード生成，読み取りアプリケーションの開発 Development of 3D Color BarCode Generation and Reading Application

奥田 真也[†]
Shinya Okuda

六井 淳[†]
Jun Rokui

1. はじめに

昨今，様々な2次元バーコードが開発され，普及している．2次元バーコードは数字・英字・漢字・カナ・ひらがな・記号・バイナリ等，様々なデータを扱うことが可能である．世界的にはPDF417が知られている[1]．日本ではカメラ付き携帯電話の普及により，QRコードが様々な場面で用いられるようになった[2]．一方，QRコードをはじめとする2次元バーコードは扱うことが可能な情報量に限界がある．従来の2次元バーコードを大容量化する方法は，アルゴリズムの改良と多色化の2種類に大別される[3]．本研究が開発した3次元カラーバーコード，Microsoft社のHCCB(High Capacity Color Barcode)等は多色化による大容量化を実現したカラーバーコードである[4]．

現在，3次元カラーバーコードをはじめとするカラーバーコードの安定した読み取り方法は確立されていない．カラーバーコードは印刷，読み取りの環境による配色の変化が生じ，読み取りが困難になる．本研究では配色が変化し，元データと異なる状態になることを「配色の劣化」と定義する．読み取り方法を提案するにあたり，カラーバーコードのRGBヒストグラムに着目した．配色の劣化の原因は，環境の影響によるRGB値の変化である．配色の劣化前のRGBヒストグラムと，劣化後のRGBヒストグラムとの形状を比較することで，欠損を確認することができる[5]．

提案手法ではカラーバーコードのRGBヒストグラムを統計モデルの一種である混合正規分布と見立てる．RGBヒストグラムは各階調における画素数の疎密を表したグラフである．RGBヒストグラムを混合正規分布に見立てることにより，組み合わせ最適化問題として定式化する．その際，未知モデルパラメータを劣化した各配色の分布の平均と分散と定める．確率分布を用いたカラーバーコード配色推定を提案し，検証実験によって提案手法の有効性を検証した[5]．

結果，本提案手法の検証実験は手動で行われる作業が含まれており，信頼性に欠ける結果となった．手動作業を自動で行うデバイスを開発する必要がある．本研究では，3次元カラーバーコード生成，読み取りアプリケーションを開発し，作業を自動化する．本アプリケーションは3次元カラーバーコード生成，読み取りの2種類の機能を備えている．前者は3章，後者は4章にて説明する．

本研究では開発の際，カメラ付き携帯端末上で動作するアプリケーションを想定する．提案手法を組み込むために，十分なスペックを有するスマートフォンやタ

ブレットPCを開発端末に用いる．現在普及している端末のOSとしてiOSやAndroidが挙げられる[6][7]．本研究ではiOSを用いる．iOSはiPhoneやiPadに搭載されている携帯端末用のOSである．本研究ではApple社が提供するタブレットPCであるiPad2を開発に用いた[6]．

2. バーコード

2.1. 2次元バーコード

元来，バーコードは商品の在庫管理等を目的として普及した．近年，情報量の大容量化等を目的として，水平，垂直方向に情報を持たせた2次元バーコードが開発された．2次元バーコードは目的に応じ，index-basedコードとdatabaseコードに大別される[3]．index-basedコードは堅牢で信頼性の高いバーコード読み取りを目的とする．コード自体のデータ容量が少なく，デザイン性を重視できる[8]．一方，データベースとネットワークシステムが必要となり，一元的な管理が必要である．index-basedコードには，Shot CodeやVisual Code等がある[3]．index-basedコードの例を図1に示す．

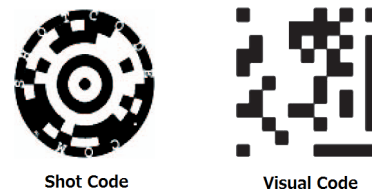


図1 index-basedコード

databaseコードはデータ容量の向上を目的とする．実コード自体を記憶媒体とすることによって，大容量のデータが符号化される[8]．databaseコードには，Data MatrixやQRコード等がある[3]．databaseコードの例を図2に示す．



図2 databaseコード

一方，2次元バーコードを更に大容量化する方法の1つに，多色化がある．多色化により小スペースのバーコードに多くの情報を付与できる．更に，多色化により，セキュリティ性の向上，デザイン性の向上等の利点

[†]島根大学大学院総合理工学研究科数理・情報システム学専攻, Department of Mathematics and Computer Science Interdisciplinary Graduate School of Science and Engineering Shimane University

がある [8] . 多色化したバーコードには , Color Code や HCCB(High Capacity Color Barcode) 等がある [3][4] . 多色化したバーコードの例を図 3 に示す .

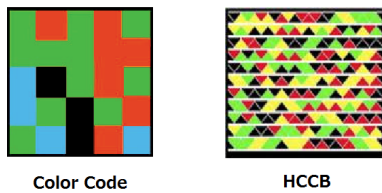


図 3 多色化したバーコード

本研究室が開発した 3 次元カラーバーコードは , database コードに分類され , 多色化によって大容量化したバーコードである . 本アプリケーションは , 3 次元カラーバーコードを用いて開発を行った .

2.2. 3 次元カラーバーコード

3 次元カラーバーコードはデータを階層に分け , x, y, z 軸上に多層化することで 3 次元的にデータを配置する [8] . 3 次元カラーバーコードは多層化することにより , 層ごとに異なる情報を与えることを可能としている . 本カラーバーコードの仕様図を図 4 に示す .

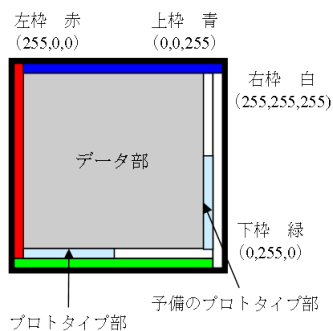


図 4 3 次元カラーバーコードの仕様図

本カラーバーコードは方向補正 , コード部抽出をするために上下左右の枠を色分けしている . コード部はデータ部 , プロトタイプ部 , 予備のプロトタイプ部を示す . プロトタイプ部 , 予備のプロトタイプ部に各配色の色パターン情報を付与する . 本カラーバーコードの色パターン数は 2^n 色 (n : 層数) となる . サンプルコードを図 5 に示す .

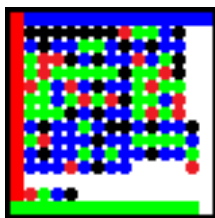


図 5 サンプルコード

図 5 は , 2 層 4 色の場合のサンプルコードである . 本カラーバーコードは正方形になるようにセルを配置す

る . 白色の箇所は空値を表す . 更に , 本研究では各セルの形状を , 従来通りの正方形ではなく真円で構成した .

本研究室では本カラーバーコードの良好な読み取り精度を確立するために読み取りアルゴリズムの提案 , 改良を行ってきた . 更なる良好な読み取り精度を確立するために , 仕様の再検討を行なう必要がある . 本研究では , 仕様の再検討の際 , 各セルの形状に着目した .

従来通り , 正方形の形状であった場合の問題点は , 各セルの境界線付近の配色の劣化が著しい点である . 一方 , セルとセルの間に空白を設けた場合 , データ部が圧迫されるという問題が発生する . 結果 , バーコードサイズが従来より大きくなってしまふ . 完全にセル同士が接することなく , かつ , データ部を圧迫しない仕様を構成する必要がある . 本研究では , 各セルを真円で表すことで , セル同士の接点を必要最低限に抑える , データ部を圧迫することなく可能な限り空白を設けることが可能となる . 本提案により , 配色の劣化の一因である , セルの境界周辺部の色の交わりを防ぐことができる .

3. 生成処理

3.1. 最適化処理

本アプリケーションは , ユーザが選択した任意のデータをバイナリデータとして扱う . バイナリデータの組み合わせから 3 次元カラーバーコードの各セルの配色を決定する .

一方 , 元データのまま配色を決定すると , 色パターンに冗長性が発生する恐れがある . 色パターンに冗長性が存在すると読み取りを行う際に , 誤りが発生し易くなる . 冗長性を解消するために , 一度最適化処理を挟む必要がある . 本研究室では , ZIP 形式や LHA 形式で扱われている Deflate Algorithm を用いて最適化を行い , 有効性を確認した [8] . Deflate Algorithm の概要図を図 6 に示す .

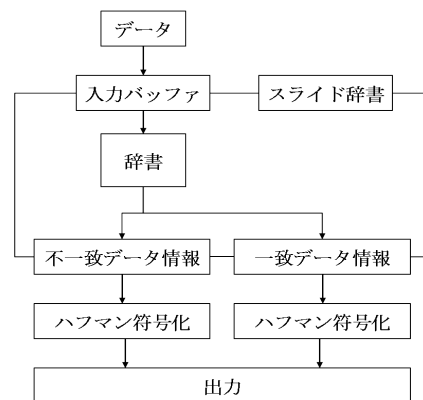


図 6 Deflate Algorithm の概要図

Deflate Algorithm はスライド辞書符号化とハフマン符号化からなる . スライド辞書符号化は入力データを辞書に見立てる . 辞書と新たな入力が一一致した際の情報を符号化する . 頻出するデータ列を発見し冗長性を解消するアルゴリズムである . 不一致データ情報と

一致データ情報は、異なるハフマン符号器に渡す。ハフマン符号化ではデータの出現頻度を計算する。出現頻度に適した長さのビットパターンを割り当てた上で符号化する。本アプリケーションは、最適化処理に ZIP 圧縮を用いた。

3.2. 多層化と重ね合わせ

本アプリケーションは、各セルの配色を決定する際に、多層化と重ね合わせ処理を行う。本アプリケーションにおける多層化と重ね合わせの概要図を図 7 に示す。

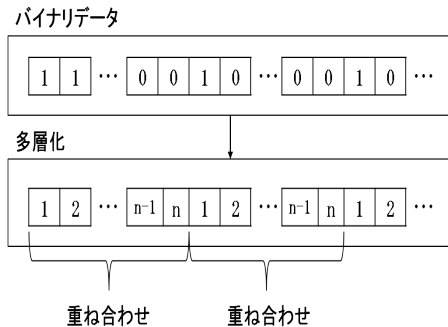


図 7 多層化と重ね合わせの概要図

図 7 のように、初めに最適化したデータをバイナリデータ列として扱う。次に、多層化するために、バイナリデータを分割する。更に、分割したデータを各層のデータと見立てる。最後に、各層の各セルを重ね合わせる。重ね合わせの際の組み合わせによって各セルの配色が決定する。配色の決定には、式 (1) を用いる [8]。

$$\begin{pmatrix} r_1 & r_2 & \dots & r_n \\ g_1 & g_2 & \dots & g_n \\ b_1 & b_2 & \dots & b_n \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} = \begin{pmatrix} r \\ g \\ b \end{pmatrix} \quad (1)$$

(r_i, g_i, b_i) ... 第 i 層の固有の色 (r, g, b)
 $f_i \in [0, 1]$... 各セルの各層毎におけるビット
 $r_1 + \dots + r_n = 255$
 $g_1 + \dots + g_n = 255$
 $b_1 + \dots + b_n = 255$

式 (1) は n 層に多層化した際の 1 セルの RGB 値を示している。導出されたベクトル $(r, g, b)^t$ によって配色が決定する。各セルの各層毎におけるビットは $(f_1, f_2, \dots, f_n)^t$ で表される。式 (1) を用いることで、可逆な復元が可能になる。

3.3. 生成機能

本節では 3 次元カラーバーコード生成機能について説明する。本機能の流れは以下の通りである。

ステップ 1 ユーザが任意のデータを選択する。

ステップ 2 最適化処理を行う。

ステップ 3 最適化処理したデータをバイナリデータとして扱う。

ステップ 4 データの多層化を行う。

ステップ 5 多層化したデータを重ね合わせる。

ステップ 6 各セルの配色を決定する。

ステップ 7 3 次元カラーバーコードを生成する。

ステップ 1 において、ユーザが画像ファイルやテキストファイル等を選択する。ステップ 2 において、選択されたデータを最適化処理をする。本アプリケーションは最適化処理に ZIP 圧縮を用いる。ステップ 3 において、最適化処理されたデータをバイナリデータとして扱う。バイナリデータとして扱うことにより、元データの形式に左右されない仕様となる。ステップ 4 からステップ 6 によって、配色が決定する。これらの手順を踏まえ、iPad2 の画像管理システムに 3 次元カラーバーコードが生成される。以上の手順を組み込んだ、本アプリケーションの動作チャートを図 8 に示す。図 8 の各項目の番号は、ステップ番号に対応している。

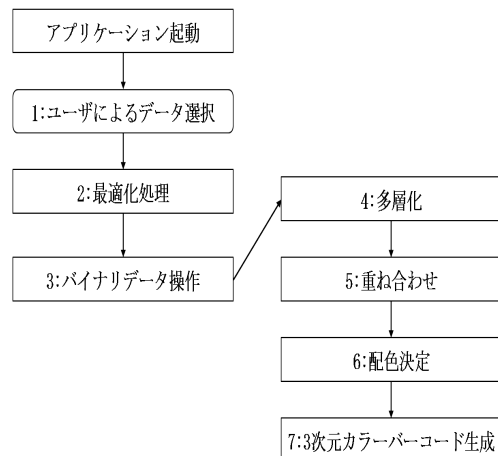


図 8 アプリケーション動作チャート (生成)

4. 読み取り処理

4.1. EM アルゴリズム

本節では読み取りの際に用いる EM (Expectation-Maximization) アルゴリズムについて説明する。EM アルゴリズムは統計学では古くから知られている手法であり、Dempster らによって一般的に定式化された [9]。音声認識や、近年では画像に隠れマルコフモデルが適用されている。EM アルゴリズムは解を逐次改良していく繰返し探索のアルゴリズムであり、ある種の確率モデルの学習において大域的最適解への良好な収束が知られている [9]。今、 N 個のデータ Z が観測データ X 、非観測データ Y 、未知パラメータ θ からなる時、確率密度関数族は $p(X, Y; \theta)$ である。この最尤推定値は、観測データの対数尤度関数族

$$L(\theta; X) := \log p(X; \theta) = \log \int p(X, Y; \theta) dY \quad (2)$$

を最大化する である [10] . 一方 EM アルゴリズムの場合 , 完全データ対数尤度関数

$$L_c(\theta; Z) := \log p(X, Y; \theta) \quad (3)$$

の条件付き期待値 (Q 関数) の逐次最大化によって , 式 (2) を実現する .

第 t 回目のパラメータ推定値を $\theta^{(t)}$ とする . 第 $1+t$ 回目の反復時は , 最初に E(Expectation) ステップで Q 関数

$$Q(\theta | \theta^{(t)}) := E\{L_c(\theta; Z) | X; \theta^{(t)}\} \quad (4)$$

を計算する . 次に , M(Maximization) ステップで Q 関数を最大にする $\theta^{(t+1)}$ を求める . 最後に $\theta^{(t+1)}$ とし , E ステップに戻る . 以上の EM アルゴリズムを整理し , 以下に示す .

ステップ 1 初期値 $\theta^{(0)}$ を設定 . $t = 0$ とする .

ステップ 2 以下を収束するまで繰り返す .

E ステップ $Q(\theta | \theta^{(t)})$ を計算する .

M ステップ $\theta^{(t+1)} = \operatorname{argmax}_{\theta} Q(\theta | \theta^{(t)})$ とする .

4.2. 混合正規分布への適用

前節で述べた , EM アルゴリズムを混合正規分布モデルへ適用させる [9] . E ステップでは , 未知パラメータを用いて Q 関数を求める . 観測データを x , データ数を N とした時 , a 次元正規分布は

$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{a/2} |\Sigma|^{1/2}} \exp\left\{-\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2}\right\} \quad (5)$$

と表される . 式 (5) を m 混合正規分布に適用させた式は

$$\sum_{M=1}^m \pi_M N(x|\mu_M, \Sigma_M) \quad \left(\sum_{M=1}^m \pi_M = 1\right) \quad (6)$$

となる . 式 (6) において , 未知パラメータ集合となる重みを π , 平均を μ , 共分散行列を Σ とする . 未知パラメータ集合を用いて , Q 関数

$$\sum_{n=1}^N \sum_{k=1}^m \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^m N(x_n|\mu_j, \Sigma_j)} \{\log \pi_k + \log N(x_n|\mu_k, \Sigma_k)\} \quad (7)$$

を求める .

次に M ステップでは , 未知パラメータ集合の最適化を行う . それぞれ , 未知パラメータ集合を最適化すると

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^N x_n \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^m N(x_n|\mu_j, \Sigma_j)} \quad (8)$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^N \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^m N(x_n|\mu_j, \Sigma_j)} (x_n - \mu_k)(x_n - \mu_k)^T \quad (9)$$

$$\pi_k^{new} = \frac{N_k}{N} \quad (10)$$

$$N = \sum_{n=1}^N \sum_{k=1}^m x_n \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^m N(x_n|\mu_j, \Sigma_j)}$$

$$N_k = \sum_{n=1}^N x_n \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^m N(x_n|\mu_j, \Sigma_j)}$$

となる . 最適化した値を新しい未知パラメータ集合として , E ステップへ戻る . ステップを繰り返し , 最終的に求めた μ_k が , 推定配色の RGB 成分となる .

4.3. 統計モデル化

本節では統計モデル化について説明する . 本研究では , 配色の劣化の原因は , 配色の劣化による画像の RGB 値の誤差にあると捉えている . 誤差を確認するために劣化前と劣化後の RGB ヒストグラムを比較する . 3 次元カラーバーコードの劣化前における RGB ヒストグラムを図 9 に示す .

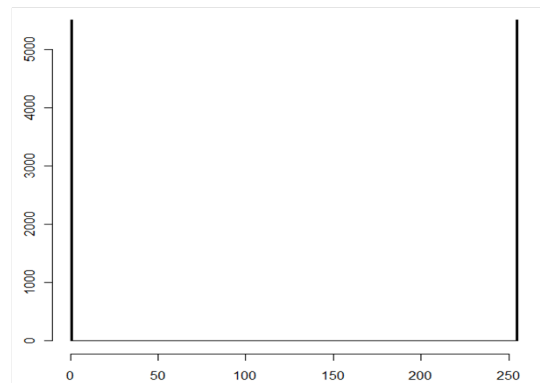


図 9 劣化前の RGB ヒストグラム

図 9 より , RGB 値が 0 と 255 のみで構成されていることが確認できる . 続いて劣化後における RGB ヒストグラムを図 10 に示す .

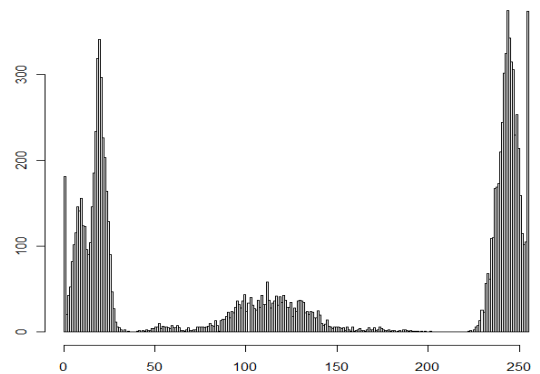


図 10 劣化後の RGB ヒストグラム

両ヒストグラムを比較すると、確かに配色の劣化による RGB 値の誤差を確認できる。カラーバーコードの RGB ヒストグラムを統計モデルである混合正規分布に見立てる。RGB ヒストグラムは各階調における画素数の粗密を表したグラフである。混合正規分布は画素の確率密度を表したグラフである。統計モデル化の概要図を図 11 に示す。

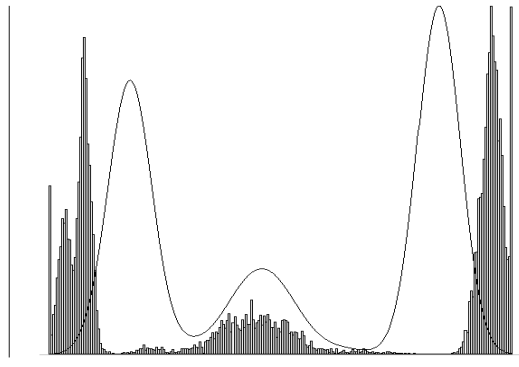


図 11 統計モデル化の概要図

図 11 の実線部は上記の RGB ヒストグラムをデータとして求めた混合正規分布のグラフである。RGB ヒストグラムの起伏は画素数の粗密を表す。混合正規分布の起伏は、サンプルデータの各ピクセルを確率変数と見なした際の確率密度の粗密を表す。図 11 より、両グラフの起伏のパターンが類似しており、統計モデル化した際にも画素数の粗密の特徴を維持していることが分かる。更に、混合正規分布はどちらの分布にも属する可能性のあるデータを、分布間で共有する特徴がある。

4.4. 読み取り機能

本節では、3次元カラーバーコード読み取り機能について説明する。本機能の流れは以下の通りである。

ステップ 1 元データである、劣化する前のカラーバーコードをプリンタで印刷用紙に印刷する。

ステップ 2 印刷されたカラーバーコードを本アプリケーションで読み取る。

ステップ 3 コード部を抽出する。

ステップ 4 方向補正を行う。

ステップ 5 コードサイズを正規化する。

ステップ 6 プロトタイプ部を抽出する。

ステップ 7 確率分布を用いたカラーバーコード配色推定をする。

ステップ 8 推定した配色を用いてカラーバーコードの各セルを読み取る。

ステップ 9 重ね合わせを復号化する。

ステップ 10 多層化を復号化する。

ステップ 11 最適化処理を復号化する。

ステップ 12 データを復元する。

以上の手順を組み込んだ、本アプリケーションの動作チャートを図 12 に示す。図 12 の各項目の番号は、ステップ番号に対応している。

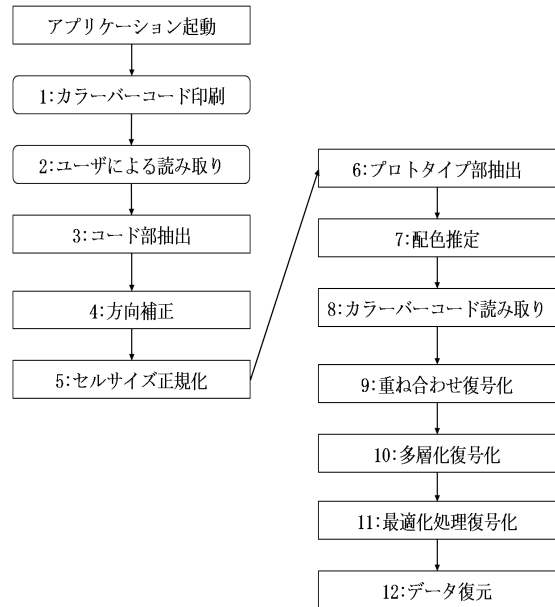


図 12 アプリケーション動作チャート (読み取り)

従来はステップ 1 からステップ 6 までは手動で行っていた。更にステップ 9 以降は存在しなかった。本アプリケーションは、ステップ 3 からステップ 13 までを自動で行う。手動を必要最低限に抑えることで、検証の信頼性を高める仕様となっている。ステップ 5 まで行ったカラーコードの各ピクセルを観測データとする。ステップ 6 でプロトタイプ部、予備のプロトタイプ部がある下 1 行と右 1 列を抽出する。プロトタイプ部を用いて、EM アルゴリズムの初期値を定める。ステップ 7 では EM アルゴリズムによって観測データが属している分布を推定する。推定した各分布の平均値が、提案手法で推定した各配色の RGB 成分である。ステップ 8 では推定した配色を用いてプロトタイプ部、予備のプロトタイプ部を含めたセルの読み取りを行う。ステップ 9 から 12 ではカラーバーコード生成の際の各符号化処理を復号化していく。

4.5. ユーザによる読み取り

本アプリケーションは、ユーザによるカラーバーコードの読み取り作業が存在する。読み取りには iPad2 のカメラ撮影機能を用いる。読み取りの際、本来のカメラ撮影機能のみだと、読み取り精度が著しく落ちる。原因として、想定外の範囲を抽出、補正の限界を超える歪曲等が考えられる。ユーザの補助となる機能をカメラ撮影機能に搭載する必要がある。本アプリケーションのユーザへの補助を図 13 に示す。

補助機能として、読み取り画面で 3次元カラーバーコードの外枠を表示する。外枠を印刷した 3次元カラー

バーコードに合わせる．ユーザに合わせてもらうことにより，的外な画面を抽出することを防ぐ．読み取りの際，上下，及び左の外枠付近に位置するピクセルデータを抽出する．枠毎に抽出したピクセルデータの RGB 成分の平均値を求める．平均値の最も大きい 3 原色の成分と，枠毎の色が一致しない限り，読み取りは行われない．

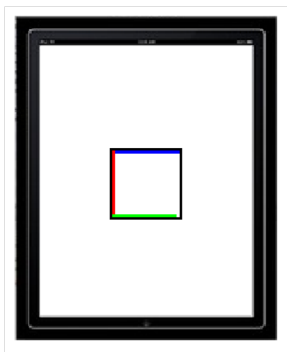


図 13 読み取り時の補助

本アプリケーションは，iOS 上で動作する仕様である．iOS 対応携帯端末を所有しているユーザならば，容易に導入することが可能である．一方，携帯端末は，本来バーコード読み取り用のデバイスではない．結果，安定した読み取り精度を得られにくい．3 次元カラーバーコード読み取り専用機器を開発した方が，読み取り精度は飛躍的に安定すると想定される．一方，読み取り専用機器はユーザへの普及率の低迷が懸念される．コスト，導入行程等に問題がある．本研究では，携帯端末上で動作するアプリケーションとして開発を進めていく．

5. まとめ

本研究では，3 次元カラーバーコード生成，読み取りアプリケーションの開発を提案した．カラーバーコード生成の際，Deflate Algorithm を用いて最適化を行った．最適化を行ったデータをバイナリデータとして扱うことで，データの形式に左右されない仕様となった．カラーバーコード読み取りの際，EM アルゴリズムを用いた．RGB ヒストグラムを統計モデルである混合正規分布に見立てた．EM アルゴリズムを用いて混合正規分布の各分布を推定することで，配色の劣化を考慮した．本研究によって，従来手動作業であった処理を自動化することが可能となった．

一方，3 次元カラーバーコード読み取り時に，本仕様の補助のみでは不十分といえる．今後，3 次元カラーバーコードの認識の仕様を改善する必要がある．更に，本アプリケーションは内部システムの実装が主となっている．従って，ユーザにとって使い勝手がよいインターフェースとは必ずしもいえない．実際，ユーザに使用してもらいアンケート調査を行う必要がある．今後，アンケート調査を元にインターフェースの改善を行う必要がある．

参考文献

- [1] HU Donghong, CHEN Xinmeng, YU Dan, HU Donghong, LI Dehua, " Algorithm for Detecting the Rows Boundary of the PDF417 Barcode ", Proc SPIE Int Soc Opt Eng, Vol.6043 No.Pt.2 Page.60431Z.1-60431Z.12 (2005)
- [2] 長屋 隆之, 山崎 知彦, 原 昌宏, 野尻 忠雄, " 高速読み取り対応 2 次元コード [QR コード] の開発 ", 情報処理学会第 52 回 (平成 8 年前期 (2)) 全国大会講演論文集, 253-254 (1996)
- [3] Hiroko Kato, Keng T.Tan " Pervasive 2D Barcodes for Camera Phone Applications ", IEEE Pervasive Computing, pp. 76-85 (2007)
- [4] Devi Parikh, Gavin Jancke, " Localization and Segmentation of A 2D High Capacity Color Barcode, Application of Computer Vision ", IEEE Workshop (2008)
- [5] 奥田 真也, 六井 淳, " 確率分布を用いたカラーバーコード配色推定 ", 情報科学技術フォーラム講演論文集 (2011)
- [6] Apple, <http://www.apple.com/> (2012/3/28 アクセス)
- [7] Android, <http://www.android.com/> (2012/3/28 アクセス)
- [8] 本多 功, 六井 淳, " 多層化を用いた 2 次元カラーコード ", 画像の認識・理解 (MIRU) (2009)
- [9] 赤穂 昭太郎, " EM アルゴリズムの幾何学 ", 情報処理 37(1), 43-51 (1996)
- [10] 上田 修功, 中野 良平, " 混合モデルのための併合分割操作付き EM アルゴリズム ", 電子情報通信学会論文誌. D-II, 情報・システム, II-パターン処理 J82-D-II(5), 930-940 (1999)