

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提案

寺田遼平[†] 藤本敬介[†] 中山泰一[†]

[†]電気通信大学 情報工学科

1 はじめに

近年, QR コードに代表される二次元コードが携帯電話に対する情報提供の手段として注目を集めている. QR コードは携帯電話に付属するカメラを用いて手軽に情報を取得でき, 対応機種も多いことから一般の目に触れることも多い. 最近ではカラー版二次元コードであるカラー ID が発表され注目を集めている.

本研究では解像度の高いカラー版二次元コードを提唱し, それを認識するアルゴリズムの手法を提案する.

2 関連サービス

2.1 QR コード

現在日本で最も普及している二次元コードである. 高速に読み取りが行えるように開発され, 多くの携帯電話機種で読み込みがサポートされている. 特許権者のデンソーウェーブは, 規格化された技術に対し特許権を行使しないとしている.

2.2 カラー ID

GMG カラーテクノロジーズと JGS が商品化を発表した二次元コード. 二次元コードのカラー版であり, QR コードを遙かに凌ぐ容量を得る. URL を読み込んでそこにアクセスするという QR コードによく取られる利用方法ではなく, データそのものをコードに埋め込むため通信を行うことなく画像や音声などの容量の大きなコンテンツを読み込むことができる.

現在は 4 色のみだが, 将来は 8 色, 16 色, 256 色などに増やすことも検討しているとされる.

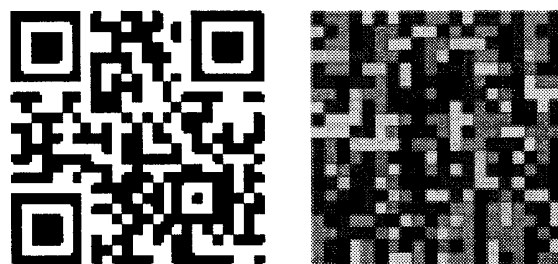


図 1: QR コードの例

図 2: カラー ID の一例

3 画像の閾値化

白黒 2 色で生成された二次元コードの読み取りには閾値化技術が用いられている. 閾値化とはグレースケールの画像から純粋な (灰色の混ざっていない) 白黒 2 色の画像を作り出す技術である. 今回は主にカメラで撮影した画像が対象であるため, 光や影の影響を認識し, 処理を加えた上での閾値化が必要となる¹.

ただし閾値化はあくまで白黒 2 色の画像を出力する技術であるため, カラー画像の色識別には使用できない. また閾値化には多くの手法が存在し研究されているが, その大部分は紙に印刷された文書をテキストデータに落とすことを目的としており, 閾値化の対象が文書であることを前提として考案された手法も多い [1].

4 提案手法

4.1 RGB 要素に分解しての閾値化

前述のとおり閾値化はグレースケールの画像から純粋な白黒 2 色の画像を作り出す技術である.

本研究ではカラー画像を読み取るので, この画像を直接閾値化することは出来ない.

しかし, 計算機上で画像を処理する場合, 1 ピクセルに対応する色とは赤要素 (R) の濃さが r , 緑要素 (G) の濃さが g , 青要素 (B) の濃さが b という表現を用いるのが普通である. このことに着目し, 特定の色要素のみを抜き出して濃さが r といった表現に書き換えられ

¹この一連の処理を含めて“閾値化”と呼ぶこともある.

Recognition algorithm for high resolution colored 2D code

Terada TERADA[†], Keisuke FUJIMOTO[†] and Yasuichi NAKAYAMA[†]

[†]Department of Computer Science

The University of Electro-Communications

ばそれぞれの色要素に対応したグレイスケールの画像が生成できる。

そして、そのそれぞれのグレイスケール画像を閾値化すれば $2^3 = 8$ 色の二次元コードの読み込みに対応できることになる。

4.2 閾値を動かすことによる色認識

8より多くの色数を用いてコードを生成する場合、単にRGB分解して閾値化するだけでは認識できる色数が足りず、正確に認識できない。

そこで、閾値化を行う際のパラメータ（閾値）を動かしながら、複数回閾値化を行うことで色認識を行う。簡単な例を挙げると、閾値を変えて2回閾値化を行い2回とも白と判定されれば白、2回とも黒と判定されれば黒、1回白1回黒と判定されれば灰色、という判別方法である。ただし、このとき使用する閾値は二次元コード上で使用されている色や影の影響を考え、工夫して設定する必要がある。

この方法と上記のRGB要素に分解する方法を使えば、多くの色数で生成された二次元コードを正しく判別することが出来る。

5 二次元コードの設計

今回提案する二次元コードのフォーマットを元に生成したコードの例が図3である。この図では12色のコードを生成している。

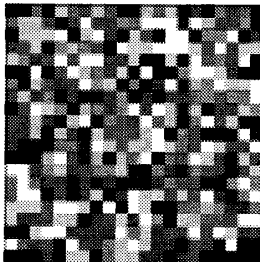


図3: 提案する二次元コードの例

二次元コード上で使用されている色をパレット情報として、二次元コードの左下から上に向かい1色につき1マスずつ塗りつぶしている。これを情報源とし、これら12色が別々の色として認識されるように複数の適切な閾値を設定する。

このようにして任意の色、任意の色数を使えるようにすることで、埋め込みたいデータの大きさから色数を決めたり、デザイン上の都合などで例えば暖色系の色を使いたくない、といったことにも対応できる。

6 実験

図3を携帯電話のカメラで撮影したものを計算機に転送し、計算機上で色認識を行った。

ただし、現時点では設計で提案したパレット情報は無視し、決め打ちで閾値を設定している。また、コードの輪郭の抽出には大津の閾値化を用いた[2]。

撮影された画像が図4、その認識を行い再現した画像が図5である。

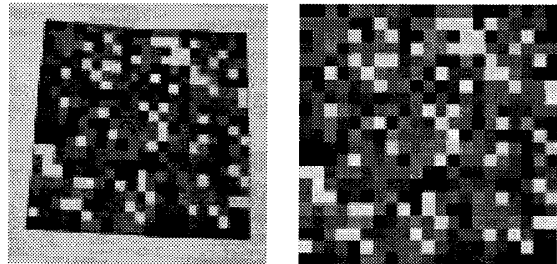


図4: 撮影されたコード 図5: 色認識の結果

左端の下から6番目のマスと7番目のマスが同じ色として扱われているなど認識に失敗しているところがあるが、おおよそ9割程度のマスが正常に認識できている。

7 まとめ

本研究では解像度の高いカラー版二次元コードを提唱し、それを認識するアルゴリズムの手法を提案した。

しかし現時点では色の認識が正確に行われていないため、パレット情報を認識しそれを加味して閾値を設定する手法の実装や、光や影の影響に対策を施す必要がある。これらの処理を加えた上でデータが正しいかどうかの誤り検出および訂正を行い、極力多い色数で、極力マスの細かな二次元コードの認識を目指す。

参考文献

- [1] Mehmet, S. and Bülent, S.: Survey over image thresholding techniques and quantitative performance evaluation (2004).
- [2] Cheriet, M. Said, J.N. and Suen, C.Y.: A Recursive Thresholding Technique for Image Segmentation (1998).