

第 19 回 パターン認識・メディア理解研究会 (PRMU)

アルゴリズムコンテスト

「あの靴、どこの？」ロゴマークからメーカー当て！

三重大学大学院 工学研究科 情報工学専攻

〒514-8507 三重県 津市 栗真町屋町 1577

ヒューマンインターフェース研究室 修士 1 年

氏名：岩田 聖也

連絡先(電話番号・メールアドレス)

岩田聖也(080-1553-6784・iwata@hi.info.mie-u.ac.jp)

応募レベル：Level1

1. プログラム開発環境

使用言語 : C++

OS : CentOS 6.5

エディタ : Emacs

OpenCV : version 2.4.9, 2.4.11

・コンパイル方法と注意事項

サンプルファイルの Makefile を用いてコンパイルを行う。

```
$ make
```

```
$ ./demo
```

で実行開始。

時間計測をする場合、Makefile の CFLAGS に以下の下線部を追加する。

```
CFLAGS= -g -O2 `pkg-config --cflags
```

```
opencv` -std=gnu++11
```

gcc 4.4.7 では時間計測が出来なかった

今回のアルゴリズムでは特徴量の取得に

MSER を用いており、OpenCV3.0.0 以上の環境では正しく動作しないことを確認した。

OpenCV2.4.9, 2.4.11 では動作確認を行った。

2. アルゴリズムの概要

MSER と RGB 特徴量を用いたクラスタリングによるロゴ(検出対象)候補領域を検出した後に、テンプレート画像と候補領域画像を正規化(100×100 pixel)し、テンプレートマッチングによりロゴ認識を行った。

処理の流れを図 1 に示す。また以下にそれぞれの処理の説明を行う。

2.1 前処理

今回の検出・認識対象であるロゴは人間の目では 1 色で構成されていると判断できるも

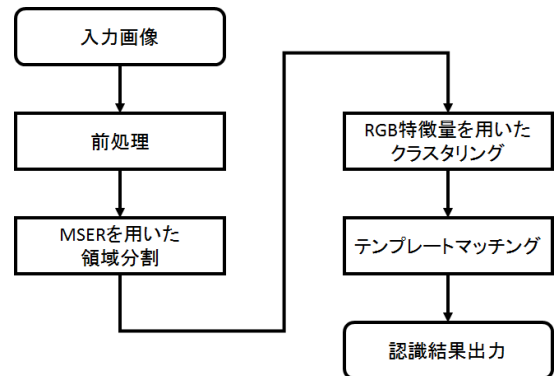


図 1. アルゴリズムの処理の流れ

のが多く、また、ロゴとロゴの印字されている周辺では色の違いが明確であると判断できるものが多い。よって、ロゴを構成する RGB 情報の削減のために減色処理を行った。今回は RGB 階調の 256 色から各階調を 16 色まで減色処理を行い、グレースケール画像に変更した。次に画像内のノイズ除去・画像の鮮明化のために平滑化とヒストグラム平坦化を行った。平滑化処理後の処理で用いる MSER は隣接画素の画素値等を用いて領域分割を行うため、メディアンフィルタやガウシアンフィルタを用いる場合、境界(エッジ)付近の画素値が混合してしまい、正しく領域を分割することが出来ない。このためエッジ付近の画素値を保持するためにバイラテラルフィルタを用いて平滑化を行った。

2.2 ロゴ候補領域の検出

図 2, 3, 4, 5 に以下で説明する各処理の処理結果を示す。

初めに、MSER を用いて画像から領域分割を行い、局所領域を検出する(図 3)。次に検出された局所領域内の RGB 特徴量を取得し、R 値・B 値・G 値それぞれの平均を算出する。その後、上記で求めた平均 RGB 特徴量を用い

てユークリッド距離を求める。ユークリッド距離と局所領域を包含する各矩形(図 4)間の中心座標からの距離が任意の閾値未満であればクラスタリングを行い、ロゴの候補領域として検出する(図 5)。クラスタリングを行う矩形が多数存在する場合はユークリッド距離が小さい上位 3 つを対象としクラスタリングを行う。

また、クラスタリングは、課題対象である「YONEX」や「瞬足」の分離しているロゴの検出を対象としている。

2.3 テンプレートマッチングによるロゴ認識

学習画像より各ロゴのテンプレートを一枚ずつ作成する。テンプレート画像には対象とするロゴが鮮明に映っている画像を用い、大津の二値化と膨張・収縮処理を行い作成する。またロゴの姿勢変化に対応するために、90 度ずつ回転、左右反転を行ってから 90 度ずつ回転を行い、各ロゴのテンプレート画像を計 7 枚ずつ追加作成した。これらのテンプレート画像を用いてマッチング処理を行う。

ロゴ候補領域として検出されている矩形内では、MSER の上記処理によりラベリングが行われている(図 6)。しかし、MSER は設定するパラメータが多く、全画像に対応した正確なロゴ領域のラベリングを行うことは困難である。ここで、上記で使用した平均 RGB 特徴量を使用する。検出処理の段階では検出した矩形毎に平均 RGB 特徴量が求められる。検出された矩形内を画素単位で走査し、対象画素の RGB 値と平均 RGB 特徴量のユークリッド距離を求め、閾値未満であれば対象画素に対しラベリング処理を行う。この処理を行うことで検出できていなかった領域もラベリングを行うことが出来る(図 7)。また、ラベリングを行うと同時に二値化処理を行うことが出来る。以上の処理で作成された画像に対しテン

プレートマッチングを行う。

類似度に対し、各ロゴの複雑さに応じ閾値を決定し、閾値以上で類似度が最大となるテンプレート画像のラベルを付ける。また、両画像の画像サイズを 100×100(幅×高さ)の同サイズに変更することでテンプレートマッチングの探索回数の削減を図り、処理速度の高速化を行った。



図 2. 元画像(lv1_YONEX_0002)

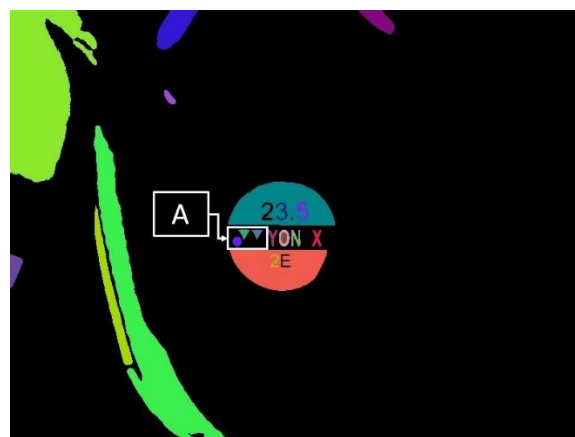


図 3. MSER による局所領域検出結果



図 4. 矩形検出(図 3 の A 矢印領域内)



図 5. クラスタリングによる矩形検出



図 6. MSER による矩形内のラベリング(2 値化)



図 7. 平均 RGB 特徴量を用いた
矩形内ラベリング(2 値化)

2.4 「FILA」の学習画像の変更

「FILA」のロゴ画像には、図 8(a)のようなロゴを構成する色が二色となる画像がある。これらは RGB 特徴量を用いたクラスタリングでの矩形検出が困難である。ここで学習画像に図 8(b)に示す画像を追加し、テンプレートマッチングを行った後に、回転・左右反転を考慮して「FILA」のラベルがついた検出矩形を広げる。検出矩形は学習に用いた図 8(b)を図 8(a)から切り出した時の比を用いて広げる。



(a) 元学習画像 (b)切り出し画像

図 8. 学習画像の変更

3. 本アルゴリズムの特徴

ロゴ候補領域検出では、MSER で取得できない領域も平均 RGB 特徴量を用いたクラスタリングとラベリングによりロゴの形状をテンプレート画像に近い形でラベリングした画像領域を検出することが出来る。また、バイラテラルフィルタを用いることでエッジ付近の画素値を保持した平滑が行え、より正確に MSER による局所領域の検出が可能となる。

認識に用いたテンプレートマッチングでは、同サイズの画像に対し処理を行うことで、探索回数を減少させ、処理速度の高速化を図った。また、テンプレート画像作成のため、各ロゴから学習画像を一枚だけしか用いない (FILA は 2 枚)。このため学習にかかる時間がほとんど無い。

4. 結果

配布されたサンプルファイルの評価を行うプログラムを用いて、提案したアルゴリズムの評価を Recall(再現度)、Precision(適合度)より行った。表 1. に Level1 での評価結果を示す。

表 1 の Precision の結果より、検出されたロゴの識別は高い精度を誇ることがわかる。一方、Recall の結果より、検出できなかったロゴの数が多いことが課題に挙げられる。

表 1. Level 1 での評価結果

Recall(再現度)	39.0565
Precision(適合度)	80.2828