

2008年PRMUアルゴリズムコンテスト

「^{ナイト}騎士を数えナイト ~ 画像に含まれる特定物体の計数 ~ 」

実施報告と受賞アルゴリズム紹介

岩村 雅一[†] 吉村 宗浩^{††} 河野 修久^{†††} 楠 卓也^{†††} 平原 一帆^{†††}
高田 征吾^{††} 山崎 真也^{††††} 古畑俊一郎^{††††} 佐藤 秀昭^{††††} 渡辺 大介^{††††}
正井 秀俊^{††††}

[†] 大阪府立大学大学院工学研究科 〒599-8531 堺市中区学園町 1-1

^{††} 立命館大学情報理工学部メディア情報学科 〒525-8577 滋賀県草津市野路東 1-1-1

^{†††} 広島市立大学大学院情報科学研究科 〒731-3194 広島県広島市安佐南区大塚東 3-4-1

^{††††} 筑波大学大学院 システム情報工学研究科 〒305-8577 茨城県つくば市天王台 1-1-1

^{†††††} 筑波大学第三学群工学システム学類 〒305-8577 茨城県つくば市天王台 1-1-1

E-mail: [†]masa@cs.osakafu-u.ac.jp

あらまし 電子情報通信学会 情報・システムソサイエティ パターン認識・メディア理解 (PRMU) 研究専門委員会が今年度実施したアルゴリズムコンテスト「^{ナイト}騎士を数えナイト ~ 画像に含まれる特定物体の計数 ~ 」の実施内容について報告すると共に、受賞者によるアルゴリズム紹介を掲載する。応募総数 33 件のアルゴリズムのうち、7 件が入賞した。キーワード アルゴリズムコンテスト、特定物体、計数

PRMU Algorithm Contest 2008

“Let’s Count the Knights —Counting a Specific Object—”

Summary Report and Prize Winning Algorithms

Masakazu IWAMURA[†], Kazuhiro YOSHIMURA^{††}, Nobuhisa KONO^{†††}, Takuya KUSU^{†††},
Kazuho HIRAHARA^{†††}, Seigo TAKADA^{††}, Shinya YAMAZAKI^{††††}, Shuniichiro
FUURUHATA^{††††}, Hideaki SATO^{††††}, Daisuke WATANABE^{†††††}, and Hidetoshi MASAI^{†††††}

[†] Graduate School of Engineering, Osaka Prefecture University, 1-1 Gakuencho, Naka, Sakai, 599-8531 Japan

^{††} Dept. of Media Technology, Ritsumeikan University, 1-1-1 Noji-Higashi, Kusatsu, Shiga, 525-8577, Japan

^{†††} Graduate School of Information Sciences, Hiroshima City University

3-4-1 Ozuka-Higashi, Asa-Minami-Ku, Hiroshima, 731-3194, Japan

^{††††} Graduate School of Systems and Information Engineering

University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8577 Japan

^{†††††} College of Engineering Systems, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8577 Japan

E-mail: [†]masa@cs.osakafu-u.ac.jp

Abstract This article presents a summary report on the algorithm contest 2008 entitled “Let’s Count the Knights —Count of a Specific Objects—,” which is sponsored by PRMU Technical Group of IEICE-ISS, with brief overviews of prize-winning algorithms provided by the winners. 7 algorithms out of 33 have won the prizes.

Key words algorithm contest, specific object, count

1. まえがき

電子情報通信学会 情報・システムサイエティパターン認識・メディア理解 (PRMU) 研究会では、1997 年度より毎年、パターン認識・メディア理解分野における代表的・基本的な研究課題をテーマとしたアルゴリズムコンテストを実施している。このコンテストは、当該研究分野における若手研究者の育成と研究会活動の活発化を目的としており、応募対象者として高等専門学校、大学学部および大学院の学生を中心とする若手研究者を想定している。このため、課題設定は初歩的かつ汎用的なテーマを選び、また、応募時期の設定や広報活動においても若手研究者が応募しやすいように配慮している。また、コンテストの審査にあたっては、アルゴリズムの完璧さや複雑さよりも素朴なアイデアを積極的に評価すべく、優秀賞のほかにも審査員特別賞を設けて積極的に評価する方針を採っている。

本コンテストでは、電子情報通信学会の秋の大会併催事業の 1 つとして表彰式を開催した。本年度も情報科学技術フォーラム (FIT2008) のイベント企画として、同フォーラムの援助のもと、9 月 4 日 (木) 13:00 ~ 15:50 に FIT 第 2 イベント会場 (慶應義塾大学 湘南藤沢キャンパス 棟 1F 11 教室) にて表彰式を開催した。この表彰式では例年、審査結果発表と入賞者表彰に加え、入賞者によるアルゴリズム紹介とコンテストに関連した研究分野に関する招待講演を実施している。本年は、「画像照合から始まる画像認識への道程」と題して、国立情報学研究所の佐藤真一氏に御講演いただいた。内容は、画像解析におけるもっとも基本的な処理の一つで、本年の課題に関連する画像照合に焦点を当て、これをさらに一般化していくことにより、顔検出に代表される物体検出技術、顔識別や一般物体認識などの画像識別技術、さらには巨大な画像・映像データベースから有用な情報を発見するためのマイニング技術にまで発展可能であることを解説していただいた。

本稿では、入賞者に執筆いただいた原稿により入賞アルゴリズムを紹介するとともに、今年度のコンテスト概要と実施内容について報告する。

2. コンテスト概要

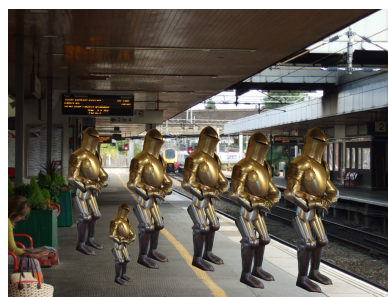
2.1 課題設定

今年度のアルゴリズムコンテストでは、「特定物体の計数」をテーマとして、画像中に特定の物体がいくつ含まれるか答えてもらうことにした。具体的には、図 1 に示すテスト画像と参照画像を提示し、テスト画像中に参照画像がいくつ含まれるかを数えるというものである。これをいかに早く、正確に行えるかを競ってもらった。しかし、早い方が良くからといって当てずっぽうで答えられるとコンテストの主旨に反するため、個数だけではなく、「大まかな位置」も同時に答えてもらい、「大まかな位置」が明らかにおかしい場合は間違いとみなすことにした。

計数の正確さを表す「スコア」は次式のように定義した。

$$\frac{100(y-z)}{x} \quad (1)$$

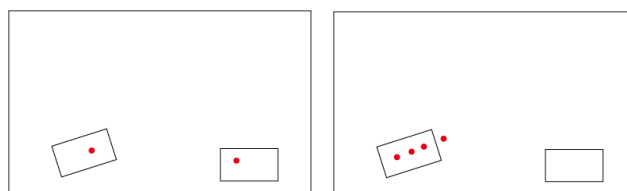
ここで x は正解の数、 y は回答のうち、画像の位置が正しいも



(a) テスト画像 .

(b) 参照画像 .

図 1 コンテスト概要 .



(a) 100 点 ($x=2, y=2, z=0$) .

(b) -100 点 ($x=2, y=1, z=3$) .

図 2 スコアの計算例 .

の数の、 z は回答のうち、画像の位置が正しくないものの数である。スコアは最高 100 点であり、誤りに応じて減少し、負の値も取る。

スコアについて、例を用いて具体的に説明する。図 2(a) は、スコアが満点の 100 点の場合である。図はテスト画像を表しており、図中の矩形はテスト画像に配された参照画像の外接矩形の正しい位置である。図中の点はプログラムが出力した参照画像の位置である。この点が参照画像の外接矩形の内側に入っていれば正しく数えているとみなし、正解とした。図 2(b) は、スコアが -100 点の場合である。この図では、左側の図中に点が 3 点あるが、これは 1 つの物体を 3 回数えたことを意味するため、正しい点は 1 つのみである。

参照画像に加える変形を変化させることで、難易度の異なる 3 つのレベルを下記のように設定した。

レベル 1 : 平行移動 + 拡大縮小

レベル 2 : 平行移動 + 拡大縮小 + 回転

レベル 3 : 平行移動 + 拡大縮小 + 回転 + オクルージョン

ホームページでは各レベル 5 枚ずつのサンプル画像を公開した。その一部を図 3 ~ 図 5 に示す。実際に配布した画像は図 1 とは異なり、背景のある参照画像とマスク画像の 2 枚の画像で図 1(b) のような背景のない参照画像を表している。

2.2 応募方法と応募状況

応募の際に応募者に要求したものは、実装した画像の数を数える関数 (ANSI-C 準拠の C 言語で書かれたもの) と、アルゴリズム説明書 (A4、2 ページ以内) の 2 点である。実装する関数はコンテストホームページ [1] からダウンロードできるサン



(a) テスト画像 .



(b) 参照画像 .



(c) マスク画像 .

図 3 レベル 1 のサンプル画像の例 .



(a) テスト画像 .



(b) 参照画像 .



(c) マスク画像 .

図 4 レベル 2 のサンプル画像の例 .



(a) テスト画像 .



(b) 参照画像 .



(c) マスク画像 .

図 5 レベル 3 のサンプル画像の例 .

ブルプログラムの関数の 1 つを書き換えてもらう形式とした。

応募されたアルゴリズムの件数と内訳は以下の通りであった。なお、同一の応募者が複数レベルに応募することが可能であることため、応募者数と応募件数は同一ではない。複数人がグループとして応募した場合は 1 グループを 1 名として数えた。応募アルゴリズム総数：33 件

[レベル別] レベル 1：16 件，レベル 2：12 件，レベル 3：5 件

応募者数：20 名 (グループ)

[種類別] 学部 2 年生:1 名，学部 4 年:8 名，修士 1 年生:8 名，修士 2 年生:2 名，高専生:1 名^(注1)

[所属別] 立命館大学：5 名，奈良先端科学技術大学院大学：4 名，広島市立大学：3 名，筑波大学：2 名，電気通信大学：2 名，大阪大学：1 名，公立はこだて未来大学：1 名，岡山県立大学：1 名，新居浜工業高等専門学校：1 名

2.3 審査方法

入賞アルゴリズムは、まず客観評価と主観評価を行い、実行委員全員による議論を経て、決定した。公平を期すため、審査は個人を特定できる全ての情報を伏せて行なった。審査の過程を下記にまとめる。

客観評価：応募されたプログラムをホームページに掲載している Linux 環境で実行し、サンプル画像 5 枚と非公開の評価用画像 5 枚 (画像の例を図 6～図 8 に示す) の計 10 枚に対するスコアと処理時間を計測した。一部のプログラムは Linux 環境で正しく動作しなかったため、Windows 環境 (Visual Studio .net) で同様に実行し、参考結果とした。どちらの環境においても正しく動作しないプログラムについては、「結果なし」とした。

主観評価：応募アルゴリズムの説明書と客観評価の結果を審査委員に提示し、それらに基づき 10 段階の主観評価 (1～10 点) とコメントを提出していただいた。

議論：客観評価と主観評価の結果に基づき、入賞アルゴリズム案を 2 案作成し、実行委員全員による議論を経て最終的な審査結果とした。本コンテストの主旨は「アルゴリズムの完璧さや実装の工夫よりも、若手研究者や学生の皆さんの独自の着眼点や素朴なアイデアを評価する」であるため、入賞アルゴリズム案を作成する際には主観評価の結果や審査委員のコメントを優先して評価したが、結果的には客観評価結果と大きな隔たりはなかった。

2.4 審査結果

最優秀賞 (1 件)

- 「直線上の色の变化を比較したマッチング」
吉村 宗浩 (立命館大学 B4) レベル 1

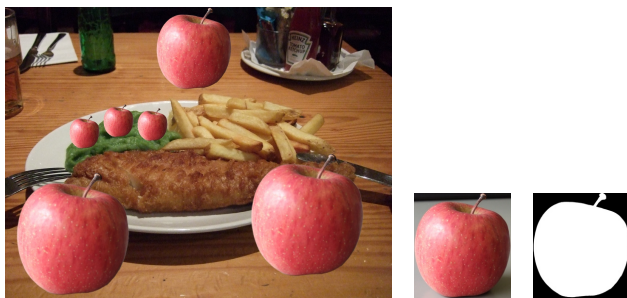
優秀賞 (2 件)

- 「画像に出現する色のプロットを用いた目的画像探索」
河野 修久 (広島市立大学 M1) レベル 1
- 「円形テンプレートを用いたアクティブ探索法による物体の位置検出」
楠 卓也 (広島市立大学 M2) レベル 2

入賞 (3 件)

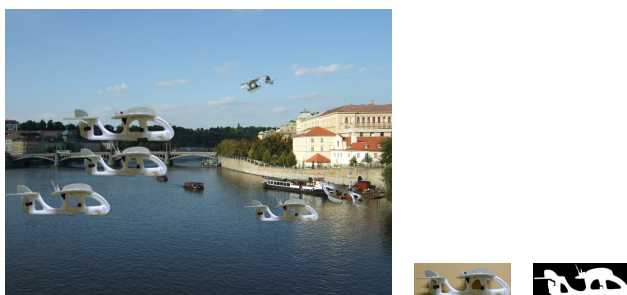
- 「頻色三色による三角形の出現を用いたアルゴリズム」
平原 一帆 (広島市立大学 M1) レベル 2
- 「色の分布と形状の特徴を用いたアルゴリズム」
高田 征吾 (立命館大学 B4) レベル 2

(注1): グループの場合は代表者の情報



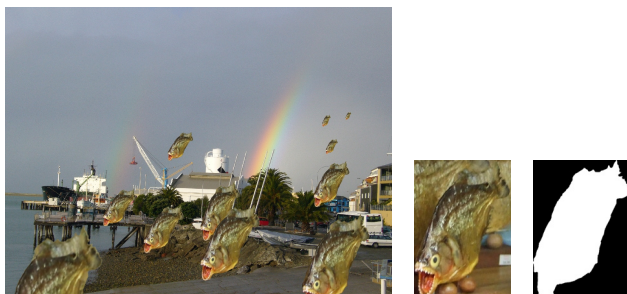
(a) テスト画像 . (b) 参照画像 . (c) マスク画像 .

図 6 レベル 1 の評価用画像の例 .



(a) テスト画像 . (b) 参照画像 . (c) マスク画像 .

図 7 レベル 2 の評価用画像の例 .



(a) テスト画像 . (b) 参照画像 . (c) マスク画像 .

図 8 レベル 3 の評価用画像の例 .

- 「円形テンプレートの色情報を用いた類似画像探索」
山崎 真也 (筑波大学 M2)、古畑 俊一郎 (筑波大学 M2)
佐藤 秀昭 (筑波大学 M1)、渡辺 大介 (筑波大学 B4)
レベル 3

審査員特別賞 (1 件)

- 「飛相似による特定画像の検出」
正井 秀俊 (筑波大学 B4)、レベル 1

2.5 組 織

本コンテストの一切は、本コンテストを主催する PRMU 研

究専門委員会、ならびに下記のメンバーからなる実行委員会が担当した。

実行委員長

馬場口 登 (大阪大学)

実行委員

幹事 岩村 雅一 (大阪府立大学)

石井 育規 (松下電器産業株式会社)

内橋 真吾 (富士ゼロックス株式会社)

川崎 洋 (埼玉大学)

島田 伸敬 (立命館大学)

鳶田 聡 (NTT サイバーソリューション研究所)

藤吉 弘亘 (中部大学)

帆足 啓一郎 ((株)KDDI 研究所)

柳井 啓司 (電気通信大学)

山口 晃一郎 (豊田中央研究所)

天野 敏之 (奈良先端科学技術大学院大学)

玉木 徹 (広島大学)

斎藤 英雄 (慶応義塾大学)

前田 英作 (NTT)

大町 真一郎 (東北大学)

角所 考 (京都大学)

内田 誠一 (九州大学)

亀田 能成 (筑波大学)

黄瀬 浩一 (大阪府立大学)

中居 友弘 (日本学術振興会、大阪府立大学)

また、本コンテストでは大阪府立大学の本道 貴行氏にホームページやポスターの作成等に助力して頂いた。ここに記して感謝の意を表する。

2.6 スケジュール

- 公開開始：2008 年 4 月 22 日 (火)
- 応募開始：2008 年 5 月 23 日 (金)
- 応募締切：2008 年 7 月 25 日 (金)
- 客観審査：2008 年 7 月 26 日 (土) ~ 8 月 5 日 (火)
- 主観審査：2008 年 8 月 6 日 (水) ~ 8 月 21 日 (木)
- 議論：2008 年 8 月 22 日 (金) ~ 8 月 25 日 (月)
- 結果発表：2008 年 9 月 4 日 (木) (FIT2008 イベント企画にて)

3. 最優秀賞：吉村 宗浩 (応募レベル 1)

3.1 アルゴリズムの概要

本アルゴリズムは、画像中の縦軸および横軸に平行な直線上の色の変化に注目したものである。レベル 1 では回転もオクルージョンも含まれないため、テンプレート画像内での縦もしくは横の 1 本の直線上の色の変化は、そのままテスト画像内でも 1 本の直線上の変化として現れるはずである。よって本手法は、テンプレート画像内で最も特徴的な直線 (以下特徴線) を抽出し、その特徴線に対応する直線をテスト画像内で探索するものである。なお、探索には抽出した特徴線上の特徴的な色の变化箇所を利用する。

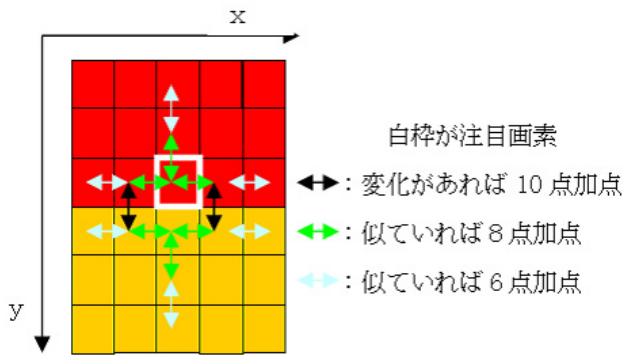


図 9 計算方法

3.2 アルゴリズムの詳細

今回は画像中の縦軸に平行な直線に注目したときを例に説明する。

3.2.1 テンプレート画像内で特徴線抽出

テンプレート画像内で画像の縦軸に平行な直線で最も特徴的な直線を抽出する。手順は以下の 5 点。

- 縦直線を上中下の 3 領域に等分する。
- 縦直線ごとに上から下へ画素を走査し、隣接する上下の画素間で輝度値の差が閾値を超える変化箇所を見つける。
- 変化箇所を図 9 のような計算方法で、周囲の状況に応じて得点を与える。

- 上中下 3 つの領域でそれぞれ最高得点をもつ変化箇所を求め、3 つの得点を合計し、その合計得点を直線の得点とする。
- 全ての直線で得点を求め、最高得点である直線を特徴線とし、その直線上の 3 領域での変化箇所を特徴変化箇所とする。

3.2.2 テスト画像内で特徴変化箇所に対応する箇所を探索

テスト画像内で 3 つの特徴変化箇所に対応する箇所を検出し、濃淡画像に記憶。手順は以下の 2 点。

- テンプレート画像内で 3 つの特徴変化箇所の変化の上部と下部の計 6 つの輝度値を構造体に記憶させておく。
- テスト画像内で 3 つの特徴変化箇所の下部の輝度値を探索し、見つかった点の上にそれぞれの特徴変化箇所の上部の輝度値があれば、テスト画像と同サイズである濃淡画像の対応する座標に、3 つの内どの特徴変化箇所かを記憶させる。

3.2.3 テスト画像内で特徴線に対応する直線を探索

作成した濃淡画像をもとに、テンプレート画像の特徴線と同じ特徴変化箇所の順で、同じ特徴変化箇所間の長さの比率を持つ直線を検出する。手順は以下の 2 点。

- 高速化のため、3 つの中で検出された数が少ない特徴変化箇所を基準にする。
- 特徴変化箇所の検出数の大きさが仮に 1 つ目 < 2 つ目 < 3 つ目だとすると、まず 1 つ目を見つけ、次に 2 つ目を見つける。その後長さの比率より 3 つ目の位置を推測し、そこで 3 つ目を探索する。

3.2.4 検出した直線が特徴線と似ているかチェック

検出した直線が正しいかどうか確かめるため、テンプレート画像の特徴線と次の 2 点で比較し、特徴線に対応する直線とし

て相応しくない直線を解答候補から除外する。

- 直線周辺の画素の輝度値
- 直線に含まれる画素の輝度値の平均

3.2.5 解答を限定する

同一物体内に何本も直線を検出してしまっている場合、1 つの物体につき直線が 1 つになるように重複を消去し、解答を限定する。

3.3 まとめ

与えられた画像の多くで、高精度かつ高速な結果を得ることが出来た。しかし、本アルゴリズムはテンプレート物体内の色の変化を利用しているので、変化のない物体を検出することは難しいと考えられる。また、テンプレート画像から特徴変化箇所を抽出する方法では、上下ではっきりと変化している箇所を抽出するため、今回用いた得点計算方法を考えた。ある程度納得出来る位置を特徴変化箇所とすることが出来たので満足している。この段階ではっきりとした特徴変化箇所を抽出出来れば、高精度の結果を得ることが可能となる。

4. 優秀賞：河野 修久（応募レベル 1）

4.1 はじめに

テスト画像内に出現する参照画像を数えようとする場合、参照画像が現れる付近には、参照画像内に出現する色が密集して現れると考えられる。そこで、参照画像に出現する色情報から作成したプロット画像を用いて、目的画像を探し出すアルゴリズムを提案する。

4.2 アルゴリズム

4.2.1 プロット画像の作成

参照画像に出現する全ての色を記憶し、テスト画像内の記憶した色が現れた場所にプロットすることで、プロット画像を作成する。しかし、記憶した色全てを用いてプロットすると、図 10(b) のように不要なプロットも多く出てきてしまう。そこで、それぞれの色の、参照画像内での出現回数 c_1 とテスト画像内での出現回数 c_2 をカウントし、 $c_1 \times 15 < c_2$ となる色は参照画像に特徴的な色ではないとみなし、プロットしないようにした (図 10(c))。またこの時、アルゴリズム後半の連結成分の取捨選択処理で使用するために、参照画像での出現回数が上位のもので、かつ互いの色が以下の式を満たす頻出色 5 色を記憶させるようにしている。ここでは、色 i の RGB 値をそれぞれ R_i, G_i, B_i としている。

$$(R_i - R_j)^2 + (G_i - G_j)^2 + (B_i - B_j)^2 > 2000 \quad (2)$$

4.2.2 プロットの連結

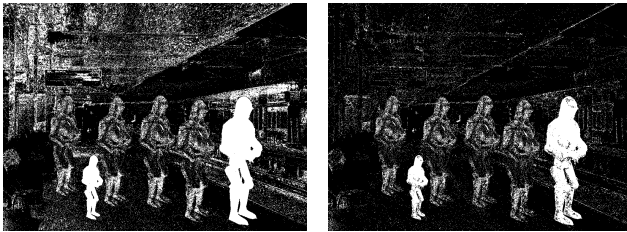
プロット画像から目的画像を探し出すために、ある程度プロット同士を連結させ、いくつかの点集合を作成するプロットの連結作業を行う。ここでは順に以下の 3 つの処理を行っている。

(1) 孤立点の消去、空白の補完

全てのピクセルを順に調べていき、注目するピクセルにプロットがある場合は 9×9 マスクを用いて孤立点の消去、プロットがない場合は 5×5 マスクを用いて空白の補完を行う。孤立点の消去、空白の補完共にマスク内のプロットの



(a) テスト画像 .



(b) 全色使用プロット .

(c) 不要色を削除した場合 .

図 10 プロット画像の作成 .

個数で処理を決定する．孤立点の消去では， 9×9 マスク内にプロット 10 個未満ならば注目点のプロットを孤立点とみなし消去する．空白の補完では， 5×5 マスク内にプロット 13 個以上ならば注目点にプロットを追加する．

(2) プロットの膨張・収縮処理

連結成分を 1 画素分大きくする膨張処理，1 画素分小さくする収縮処理をいずれも 8 近傍で複数回行う．膨張と収縮を組み合わせて行うことで，孤立した点や小さな孔を検出，消滅させる．

(3) メディアン・フィルタ

膨張・収縮処理を終えたプロット画像に対しメディアンフィルタをかける．これにより，上記 2 つの処理と同様に，孤立点や小さな孔を消滅させる．

4.2.3 連結成分の取捨選択

前節で作成した連結成分の全てが目的画像の位置を示しているとは限らないため，連結成分の取捨選択を行う．

各連結成分に対し，4.2.1 節で記憶させておいた頻出色 5 色の出現の有無を調べる．頻出色 5 色のうち，3 色以上出現する連結成分は目的画像が存在すると判断し，3 色未満の場合目的画像が存在する場所でないとは判断してその連結成分を削除する．これにより，不要な連結成分をほぼ削除することができる．最終的に残った各連結成分に対し距離変換を行い，最も大きい値をとる場所の 1 つを目的画像が存在する位置の座標として出力する．

4.3 考 察

参照画像に出現する色情報から作成したプロット画像を用いて目的画像を探し出すアルゴリズムを提案した．プロット画像を作成するために不要となる色情報の削除や作成した連結成分

の取捨選択をうまく行うことで，正答率の高い目的画像探索手法にすることができた．ただし，参照画像によって，出現する色の関係でうまく目的画像が見つけれない場合もあったので，改良の余地があると思われる．

5. 優秀賞：楠 卓也 (応募レベル 2)

5.1 はじめに

円形のテンプレートをを用いた高速な探索により物体の位置検出を行う手法について述べる．本手法は主にレベル 2 を対象としており，物体の回転に対してロバスト性を担保した位置検出を行うことを目的とする．

5.2 アルゴリズムの説明

本アルゴリズムは，検出対象の参照画像から円形のテンプレートを作成し，画像中からテンプレート内のヒストグラムの類似度が高い位置を探索することで物体の位置検出を行う．

5.2.1 概 要

本アルゴリズムの処理手順を以下に示す．

step1(参照画像からの円形テンプレートの切り出し)

参照画像から背景を含まない最大の矩形領域を抽出し，この矩形領域内の内接円をマスク画像とする．例として，図 11 の参照画像から切り出した円形テンプレートを図 12 に示す．

step2(円形テンプレートのヒストグラムの作成)

step1 で作成した円形テンプレートの色ヒストグラムを作成する．このとき作成するヒストグラムは，RGB 値をそれぞれ 8 階調に落とした $8 \times 8 \times 8 = 512$ 次元の正規化ヒストグラムである．

step3(アクティブ探索による位置検出)

step2 で求めた円形テンプレートのヒストグラムを用いてアクティブ探索を行う．まず，テンプレートの拡大率を 10% ~ 200% の間で 10% 間隔で変化させる．そして，各拡大率においてアクティブ探索を行い，テスト画像中の探索位置でテンプレートと同一サイズの円形領域の正規化ヒストグラムを作成し，2 つのヒストグラムの類似度を計算する．このとき，類似度に応じて，値の更新かあるいは計算不要な画素のスキップを行う．類似度が閾値 (85%) よりも大きい位置を出力位置とし，閾値よりも大きい類似度をもつ位置が検出されなくなるまで探索を行う．なお，テンプレートの拡大率を 200% から順に下げて探索を行っており，探索中に検出した位置とその周辺はそのときの円形テンプレートのマスク画像により以後の探索範囲から除外している．

5.2.2 工 夫 点

本アルゴリズムの工夫点は，参照画像から作成した円形のテンプレートを探査に用いている点とアクティブ探索により探索を行っている点である．

円形テンプレートを探査に用いることで，検出対象物体の回転に不変な特徴量となる色ヒストグラムの算出が可能となる．それに加えて，テンプレートを参照画像の背景を含まない最大の矩形領域から作成することで，参照画像中から回転に不変な特徴量を最も多く取り出すことを可能にしている．また，アクティブ探索を用いることで，検出対象物体がありそうなところ



図 11 参照画像



(a) 矩形領域 (b) マスク画像

図 12 円形テンプレート

表 1 サンプル画像への適用結果

レベル	物体の検出率 [%]	平均スコア [点]
レベル 1	100	69.4
レベル 2	100	100.0
レベル 3	73	46.8

は細かく、そうでない場所は粗く間引いて照合を行うことが可能となるため、高速な探索が実現できる。

5.3 実験

本アルゴリズムを与えられたサンプル画像(各レベル5枚)に適用した。結果を表1に示す。

表1より、本手法が主に対象とするレベル2が最もよい結果となっており、物体の回転に対してロバスト性を担保できていることがわかる。しかし、レベル3の結果は悪く、レベル3のみ物体の検出率が100%となっていない。このことから、本アルゴリズムが対象物体のオクルージョンには対応できていないこともわかる。レベル1の平均スコアが低いのは、1つの検出対象物体に対して異なる位置で複数回の検出をしてしまったサンプル画像があったためである。また、筆者の計算機環境での平均処理時間は1.0秒となり、アクティブ探索法を用いることで高速な探索が実現できている。

5.4 まとめ

今回、特定物体の位置検出を行う手法として、円形テンプレートを用いてアクティブ探索を行う方法を述べた。テンプレートを円形とすることで物体の回転に対してロバスト性を担保できている、なおかつアクティブ探索による高速な探索も実現できた。

6. 入賞：平原 一帆 (応募レベル2)

6.1 アルゴリズム概要

本アルゴリズムは、参照画像の色情報を用いている。テスト画像中の参照画像が現れる部分には、参照画像で使われている色が密集して出現すると考えられる。これを効率よく特定するために、最頻色(参照画像に最も多く出現する色)と、最頻色と色合いの異なる頻色二色の計三色を用いる。これらの三色を各頂点とした最小の三角形を描き、その重心・頂点の出現点を、参照画像の現れる候補とする。

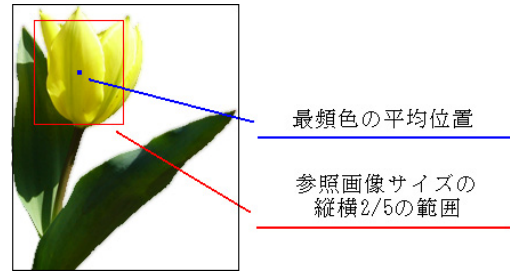


図 13 頻色の出現範囲。

6.2 アルゴリズム詳細

以下、アルゴリズムについて詳述する。

6.2.1 頻色の決定

参照画像に出現する全ての色を記憶し、その中から頻色として用いる三色を決定する。最頻色は、参照画像に出現する画素のうち、出現回数が最も多いものとする。次頻色は、「1. 最頻色と色合いがある程度異なる」「2. 最頻色のある程度近傍に出現する」、二点を満たす色を選択する。具体的には、「1. RGB値のR値G値B値のいずれかが最頻色より10階調以上異なる(上限255階調)」2「最頻色の出現点の平均位置である中心から、参照画像のサイズの縦横2/5以内の範囲に出現する」ことが条件となる。なお、三色目の選択時には、最頻色と次頻色それぞれから、ある程度色合いが異なるものを選択している。

6.2.2 テスト画像の走査

テスト画像を走査し、頻色が出現する点にプロットする。テスト画像に現れる参照画像部分は、殆どの場合、拡大・縮小・回転が行われているため、元の参照画像のRGB値と完全一致しない場合が多い。そこで、選定した頻色のRGB値と完全一致する点だけではなく、RGB値にある程度の幅を持たせて走査を行うことで、参照画像部分の検出を行う。

この時、RGB値に持たせる幅は、参照画像に現れる頻色の多寡で動的に決定している。これは、参照画像に現れる頻色の数が多ければ、テスト画像走査時の参照画像部分に頻色が現れやすく、参照画像に現れる頻色の数が少なければ、テスト画像走査時の参照画像部分に頻色が現れにくいという傾向に基づく。

6.2.3 三角形の形成

プロットしたテスト画像を元に、三角形の形成を行う。この三角形は、最頻色・頻色の三色それぞれを頂点とするものである。頻色三色のうち最もプロットが少ない色を起点として、近距離にある点を探し、最小の三角形を形成・重心点の計算を行う。この時同時に全ての三角形の起点と重心点との距離から、全体の起点・重心点の平均距離を計算する。

全ての三角形を用いるとノイズとなる候補点が多く出現する。これを軽減するために、以下の操作で候補点を候補から外す。

(1) 重心点との距離が平均距離以上の場合は、ノイズを含むと判断してプロットしない。

(2) 頻色三色の一色が、他の二色に比べて出現割合が高すぎる場合はノイズと判断してプロットしない。

(3) 頻色三色のうち、参照画像に出現する画素の300倍以上出現している場合はノイズと判断してプロットしない。

以上の補正を加味した上で、テスト画像に候補点となる顔色三色と重心点をプロットする。

6.2.4 候補の絞込み

候補点のうち、孤立点を消去し近傍点同士をクラスタリングすることで、参照画像部分の絞込みを行う。孤立点の消去では、候補点の周囲にある程度以上の点が存在しない場合、その候補点を孤立しているとして消去する。この時、孤立点の判定に用いる周囲の点の数は、テスト画像全体に対するプロットの数・プロット間の平均距離、連結成分の数を考慮して決定する。すなわち、全体のプロット数や平均距離が小さい場合は、孤立点と判定するための値も小さいものとする。

孤立点の消去の後、近傍点のクラスタリングを行う。近距離に存在する点同士にラベル付けを行い同じクラスとし、同じクラス内での平均位置を参照画像部分の候補点として計数する。

6.3 まとめ

参照画像の特徴点として顔色三色を使用し、それを三角形の各頂点として用いることで候補点を決定するアルゴリズムを提案した。色情報のみを用いて参照画像を特定する際に障害となるノイズを取り除く手法として、三角形の一頂点と重心点との平均距離を利用した。実行速度を速めるために、近傍点を求める走査時にテスト画像全体を操作するのではなく、ある点から同心円ごとに走査を行い最初に見つかった点を近傍点とするなどの工夫を行っている。本アルゴリズムでは、種々の閾値について、動的ではあるがヒューリスティックに設定していたため、より良い閾値が存在する可能性がある。また、三角形ではなく、顔色四色の四角形を用いることで、精度が向上する可能性もある。

7. 入賞：高田 征吾（応募レベル2）

7.1 アルゴリズム概要

今回提案するアルゴリズムは、テンプレート画像の形状特徴、色特徴を用いたものである。テンプレート画像内の輝度値は、テスト画像内の解が存在する箇所によく分布しているはずであるので、その箇所を探し、形状特徴、色特徴を比較する事で解かどうかを判断する。テンプレートが拡大縮小や回転をしているも分布は同様である事を利用している。また、アルゴリズムを単純にする事で処理の高速化を実現している。

7.2 アルゴリズム

7.2.1 背景除去

テンプレート画像に存在しない画素をテスト画像から除去し、図 14(a) の背景除去画像を作成する。手順は以下。

- テンプレートに存在する全ての輝度値を色記憶用の配列に記録する。

- テスト画像を走査し、色記憶用配列を参照して同じ値が見つかれば作成した濃淡画像にその座標を記憶させ、座標値を 255(白色)にする。

7.2.2 マスク画像作成

背景除去した二値画像を用いて図 14(b) のマスク画像を作成する。手順は以下。

- 二値画像 $8\text{pixel} \times 8\text{pixel}$ を一つの領域として考え、各領



(a) 背景除去画像 .

(b) マスク画像 .

図 14 背景除去画像とマスク画像

域内の白色の点の数を数える。

- 数えた白色の点の数が閾値を超える場合、その領域内全ての値を 255(白領域)にする。また閾値以下の時は領域内全ての値を 0(黒領域)にする。

以下、隣接する同じ色の領域を一つの領域として考える。

7.2.3 マスク画像内各領域の解判定

マスク画像内白領域を順番に注目し、テンプレートと異なる形状特徴、色特徴をもとに解領域を求める。手順は以下。

- 孤立点除去...マスク画像全体の孤立領域 ($8\text{pixel} \times 8\text{pixel}$ 領域) の上下左右 4 近傍領域を参照し、4 近傍全て黒領域である場合対象領域を黒領域にする。

- テンプレート画像がもつ以下の処理で用いる特徴量 (外接四角形の頂点座標、面積、複雑度、RGB ヒストグラム) を求める。また、マスク画像内各白領域がもつ以下の処理で用いる特徴量を求める。以下、各白領域特徴とテンプレート特徴を用いて処理を進める。

- 線分の判定...対象領域が x 軸または y 軸に一繋がりである時、解領域でない判定する。

- 面積による判定...対象領域の面積がテンプレート画像の最大拡大率 2.0 倍 (面積 4.0 倍) 以上、もしくは最小縮小率 0.1 倍 (面積 0.01 倍) 以下の時、解領域でない判定する。

- 複雑度による判定...対象領域の複雑度とテンプレート画像の複雑度の差の二乗値を計算する。値が閾値以上の時、解領域でない判定する。複雑度 = $(\text{周囲長})^2 / (\text{面積})$ 。

- 色による判定...正規化したテンプレート画像 RGB ヒストグラムと各領域 RGB ヒストグラムとの差の絶対値の総和を求める。総和が閾値を越える時、解領域でない判定する。

- 以上の処理が通った領域を解領域とし、解座標は対象領域外接四角形の中心座標とする。

- 全ての白領域を判定するまで繰り返す。

7.3 結果と考察

用意された Lv2 画像までの結果を見ると、得点が良い画像が 5 枚存在しているが、極端に得点が悪い画像も存在している事がわかる。得点の悪い理由は背景除去の時に上手く解と背景が区別されなかったからだと思われる。また、特徴量を比べる際に閾値は二つ用いており、複雑度を比較する際の閾値とヒストグラム比較の際の閾値である。これらの値は全て統計的な値を用いているが、全ての画像に完全に適用できるとは限らない。画像によってはヒストグラム比較の際の適切な閾値が現在設定

表 2 画像ごとの点数と実行時間 (VisualStudio2005 : メモリ 512MB : CPU:Pentium4 2.02GHz)

画像	得点 (正解/間違い)	実行時間 (s)
Lv1/1	0 点 (0,0)	1.230000
Lv1/2	33 点 (2,1)	0.562000
Lv1/3	100 点 (10,0)	0.250000
Lv1/4	100 点 (5,0)	0.562000
Lv1/5	100 点 (7,0)	0.438000
Lv2/1	100 点 (6,0)	0.344000
Lv2/2	100 点 (10,0)	0.360000
Lv2/3	60 点 (3,0)	0.250000
Lv2/4	71 点 (5,0)	0.343000
Lv2/5	63 点 (7,2)	0.297000

の値を超える場合があるので画像によった調整が必要であると考える。

7.4 まとめ

背景除去が上手くいかない画像に対しての得点は良くはないが、誤検出が少ないので「正解数 < 間違い数」になる画像は存在しなかった。背景除去が上手くいく画像は非常に得点が高かったため、どれだけ上手く解と背景を分離できるかがこのアルゴリズムの得点向上において鍵となる。また、時間のかかる背景除去は効率的に構造体リストを用い、計算量において浮動小数演算を用いない事でアルゴリズムの高速化を実現できた。

8. 入賞 : 山崎 真也・古畑 俊一郎・佐藤 秀昭・渡辺 大介 (応募レベル 3)

8.1 アルゴリズムの概要

本アルゴリズムでは、拡大や縮小、回転に対して普遍な特徴として、円形テンプレートの色情報を利用した。対象画像と参照画像から円形に切り出した領域の色のユークリッド距離を比較することで、教示画像との間でテンプレートマッチングを行い、類似画像を発見することを基本方針とする。なお、対象とするレベルはレベル 1,2,3 とする。

8.2 アルゴリズムの詳細

8.2.1 マスク画像からテンプレート領域を算出

最初に、マスク画像に対して白画素が 1 画素になるまで収縮処理を行い、それを中心画素と定義する。次に中心画素を中心に、前景領域内で最大の円形領域を取得し、その領域をテンプレート領域とする (図 15)。

8.2.2 テンプレートの作成と検索範囲の絞り込み

8.2.1 で取得されたテンプレート領域の RGB 各色の色ヒストグラムを参照画像から算出し、これをテスト画像とのマッチングに使用するテンプレートとする。

しかし、テンプレートをテスト画像全体に当てはめると計算範囲が膨大になってしまう。そこで、8.2.1 で求めた中心座標の画素を使用し、その画素との画素値の差が ± 5 である画素をテスト画像から算出し、注目画素とする。なお、誤差の値は経験的に決定した。そして、注目画素を中心に、テンプレートマッチングを行う。

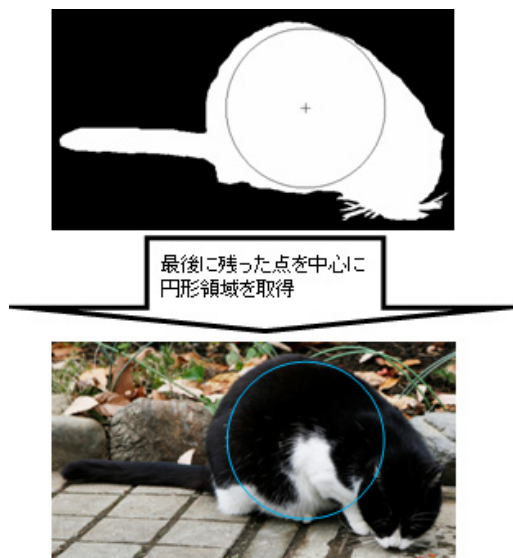


図 15 テンプレート領域の決定

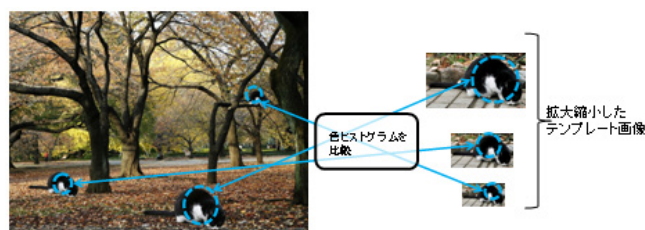


図 16 テンプレートとの比較

8.2.3 テンプレートとのマッチング

8.2.2 で求めた注目画素を中心に 8.2.1 で求めたテンプレート領域とのマッチングを行う。マッチングの指標には、テスト画像とテンプレート領域との間の RGB 各色の色ヒストグラムのユークリッド距離を使用する。実験的に求めた閾値よりユークリッド距離が小さい注目画素を、テンプレートとマッチしたものと判断する。

なお、本課題では対象の拡大縮小も考慮しなければならない。それに対しては、テンプレートを 0.1~2.0 倍まで 0.1 刻みで大きさを变化させたテンプレート領域を用意し比較を行うことで対応する (図 16)。

8.3 結果と考察

以上の処理により、Level1、2、3 の画像から参照画像の探索を行った。各 Level で指定された 5 枚の画像に対して適用したところ、Level1 では平均スコア 36、Level2 では平均スコア 85.6、Level3 では平均スコア 42 を獲得することができた。個別の結果を観察すると、参照画像と似た色の分布が存在する画像や参照画像の縮小率が大きい画像では、誤答が多かった。これは、マッチングの際の閾値を経験的に決定しているため、その閾値の値により結果が左右されてしまい、画像により得手不得手が大きく分かれてしまったためと考えられる。

8.4 まとめ

参照画像と対象画像の色情報に注目し、テンプレートマッチ

ングを行うことで、対象画像内に含まれる教示画像の領域を指示するアルゴリズムを提案した。拡大縮小には、テンプレートの大きさを変化させることで対応した。その結果、テスト画像中に参照画像と似た色の領域がない場合には、高い正答率が得られた。また、オクルージョンのある画像に対しても、テンプレート領域の大部分が見えていれば、正しいマッチングを行うことができた。

9. 審査員特別賞：正井 秀俊（応募レベル1）

9.1 アルゴリズムの概要

Level1 の対象は拡大、縮小のみの処理が施されているので、その不変量である相似関係に着目する。もし、画像がユークリッド空間上で連続的なものとして定義されているならば、どのような拡大縮小に対しても対応する点が存在する。しかし、実際の画像は離散的な空間上のものであり、拡大によって、画素が増え、縮小によって画素が失われてしまう。このような画素の増減に対応するため、飛相似（とびそうじ）という関係を提案する。

9.2 飛相似

テスト画像上と参照画像上のそれぞれ同じ画素値を持つ4組の点を探索し、それぞれが作る長方形が相似関係にある場合、飛相似が成立する、と定義する。飛相似の成立を判別する具体的な処理を以下に述べる。

(1) 参照画像とテスト画像で同じ画素値を持つ点を1組探索する。参照画像上の点を (x, y) 、これに対応するテスト画像上の点を (X, Y) とする。

(2) 点 $(X, Y + 1)$ の画素値を記憶し、参照画像の点 (x, y) の下方向にある点 $(x, y + j) (j = 1, 2, \dots)$ の画素値が記憶した値と同じであるかを調べる。

(3) (2) で同じ値を持つ点を見つけれなかった場合、同様の処理を $(X, Y + i) (i = 2, 3, \dots)$ について行い、同じ値を持つ点が見つかるまで繰り返す。見つからなければ、飛相似は成立しない、とする。

(4) 同じ画素値を持つ点 $(X, Y + k)$ と $(x, y + l)$ が見つかった場合、同様の処理を右方向に行う。ただし、今回は参照画像上での探索範囲を、得られた比率 (k/l) を用いて限定する。具体的には、 $(X + m, Y)$ に対して $(x + [m * k/l], y)$ (注2) の周辺のみ調べる。同じ画素値を持つ点を見つけれれば、(5)へ進み、見つからなければ飛相似は成立しない、とする。

(5) (4) で見つかった点を $(X + m, Y)$ と $(x + n, y)$ とする。このとき $(X + m, Y + k)$ と $(x + n, y + l)$ の画素値が同じであれば、テスト画像上と参照画像上のそれぞれに同じ画素を持つ4組の点が見つかったことになり、飛相似が成り立つ、とする。

9.3 飛相似による特定画像検出の流れ

飛相似による特定画像検出の流れを図17に示す。この手法では飛相似成立後、参照画像を相似比倍し、テスト画像上の記憶した点のまわりに重ね、参照画像上にあると思われる点を探

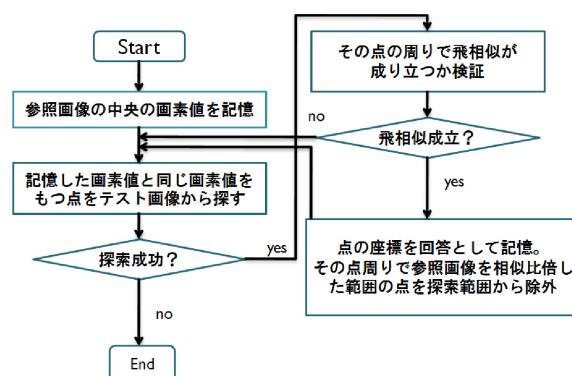


図17 アルゴリズムの流れ

索範囲から除外している。この時、除外する範囲がテスト画像からはみ出てしまったら、回答としない、という処理を行っている。

また、この手法では参照画像の中央の画素値は拡大縮小で消えないという経験則を用いている。

9.4 利点と問題点

利点：処理が単純であるため、非常に高速である。(平均0.04秒)

問題点：

- 参照画像の中央の画素が存在しない(マスク画像が黒)場合、正しい結果は得られない。
- 参照画像の中央周辺の画素がテスト画像上に大量に存在する場合、正しい結果は得られない。

9.5 考察とこれから

単純だが効果的なものを。これが今回のコンテストの私のテーマであり、それを上手く実現できた。今後は中央値のみでなく他の点も探索する等、飛相似成立の条件を緩和し、飛相似が成立した点の周辺で他の手法を組合せることで、速さと精度を実現できるのではないかとと思われる。

10. むすび

本コンテストは1997年から継続して行われ、若手研究者に様々な「きっかけ」を提供してきた。この分野の魅力を感じるきっかけ、理解を深めるきっかけ、新しいことを考えるきっかけ、自分の考えを人に伝えるきっかけ、同年代の研究者と知り合うきっかけ。数え上げれば切りがない。今回も皆さんにそのようなきっかけを提供することができたならば幸いである。

最後に、本コンテストを実施するにあたり、多くの方々から大変親身なご助言とご協力をいただいた。ここにその旨を述べ感謝の気持ちを表したい。

文 献

- 2008年PRMUアルゴリズムコンテストホームページ, <http://alcon.imlab.jp/>.
- 天野敏之, 玉木徹, 村上剛基, 青木康昌, 尾藤幸司, 秋間雄太, 荒木孝介, 佐古祐, 花岡昇平, “2007年PRMUアルゴリズムコンテスト「ジグソーパズルを完成させよう! マッチングによる画像復元」実施報告と受賞アルゴリズム紹介,” 信学技報PRMU2007-157, Dec. 2007.

(注2): $[x]$ は x を越えない最大の整数を表す。