

2010年度PRMUアルゴリズムコンテスト 「ターゲットをロックオンせよ！～移動物体の追跡～」

実施報告と受賞アルゴリズム紹介

出口 大輔[†] 村瀬 洋[†] 梅田 一彰^{††} 斎藤 正孝^{†††} 藤 賢一朗^{††††}
山下 真吾^{††††} 末次 祐樹^{†††††} 高井 翔太^{††††††} 佐藤 竜太^{††††††} 丸山 拓馬^{†††††††}

† 名古屋大学大学院情報科学研究科
†† 岡山県立大学大学院情報系工学研究科
††† 中京大学情報理工学部
†††† 宮崎大学大学院工学研究科
††††† 筑波大学大学院システム情報工学研究科
†††††† 立命館大学情報理工学部
††††††† 電気通信大学大学院情報理工学研究科
E-mail: †ddeguchi@is.nagoya-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「ターゲットをロックオンせよ！～移動物体の追跡～」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。今回の課題は「動画中の物体追跡」をテーマとし、高速かつ精度良い物体追跡の実現を課題とした。また、応募総数は歴代2位の73件であり、そのうち7件が入賞した。
キーワード アルゴリズムコンテスト, 物体追跡, 物体検出

PRMU Algorithm Contest 2010 “Lock on targets! – object tracking –”

Summary Report and Prize Winning Algorithms

Daisuke DEGUCHI[†], Hiroshi MURASE[†], Kazuaki UMEDA^{††}, Masataka SAITO^{†††},
Kenichiro FUJI^{††††}, Shingo YAMASHITA^{††††}, Yuki SUETSUGU^{†††††},
Syota TAKAI^{††††††}, Ryuta SATO^{††††††}, and Takuma MARUYAMA^{†††††††}

† Graduate School of Information Science, Nagoya University
†† Graduate School of Systems Engineering, Okayama Prefectural University
††† School of Information Science and Technology, Chukyo University
†††† Graduate School of Engineering, Miyazaki University
††††† Graduate School of Systems and Information Engineering, Tsukuba University
†††††† College of Information Science and Engineering, Ritsumeikan University
††††††† Graduate School of Infomatics and Engineering, The University of Electro-Communications
E-mail: †ddeguchi@is.nagoya-u.ac.jp

Abstract This paper reports a summary of the 14th algorithm contest entitled “Lock on targets! – object tracking –”. This contest is supported by PRMU (Pattern Recognition and Media Understanding) technical group of IEICE-ISS. In this paper, brief overviews of prize-winning algorithms are presented by the winners. The contest in this year asks fast and accurate algorithms to track objects from videos. 73 algorithms were submitted to this contest (2nd largest submissions among the past contests), and 7 algorithms had won the prizes.

Key words algorithm contest, object tracking, object detection

1. はじめに

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) 研究会が主催するアルゴリズムコンテストでは、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題材として取り上げ、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行っている。今回で 14 回目を迎える本コンテストは、1997 年度より毎年開催されている PRMU 研究会の一大イベントでもある。若手研究者およびこれから研究者を目指す学生 (高等専門学校、大学学部、大学院の学生) を主たる対象に、具体的な課題を解決することの楽しさを通して研究の面白さを体験してもらうことを目指している。そのため、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し、優秀賞のほかにも審査員特別賞などを設けている。

本年度も情報科学技術フォーラム (FIT2010) のイベント企画として、同フォーラムの援助のもと、9 月 7 日 (火) 13:30~16:30 に FIT 第 2 イベント会場 (九州大学 伊都キャンパス 西講義棟 2F 工学部 第 1 講義室) にて表彰式を開催した。この表彰式では例年、審査結果発表と受賞者表彰に加え、受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特別講演を実施している。本年度は、「時空間 MRF モデルによるトラッキング -ITS への応用-」と題し、東京大学の土上俊介氏に、同氏の研究グループがこれまでに開発してきた画像トラッキング技術、およびその ITS (高度交通システム) への応用例について紹介しながら、ITS における画像センシングの役割に関して非常に興味深い内容をご講演いただいた。

本稿では、2. で本年度のコンテスト概要と実施内容を報告するとともに、応募アルゴリズムの種類と傾向をまとめる。そして、3. 以降に受賞者による詳細なアルゴリズム紹介を掲載する。

2. コンテスト概要

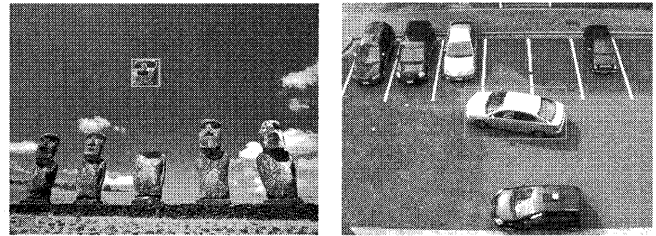
2.1 課題設定

今年度のアルゴリズムコンテストでは、「移動物体の追跡」を題材として取り上げ、動き回る物体を撮影した動画画像から移動物体を発見し、追跡するアルゴリズムを募集した。具体的には、図 1(a) のように移動する物体を撮影した動画画像を入力とし、図 1(b) のように動画画像中の各フレームにおける移動物体を囲む矩形を出力するアルゴリズムの作成を課題とした。ただし、プログラム作成負荷の軽減、および評価手法の統一を目的として、画像の入出力関数や出力結果の評価関数などをサンプルプログラムに含め、コンテストのホームページ [1] 上で配布した。応募者には動画画像中の各フレームから移動物体を検出・追跡する関数の提出を求めた。

今回のアルゴリズムコンテストでは、CG 画像や実画像といった入力動画画像の違い、照明変化やオクルージョンの程度の違い、によって以下の 3 つのレベルを設定した。

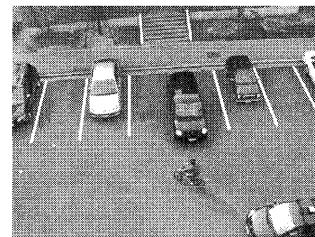


図 1 コンテスト概要. (a) 入力画像系列, (b) 出力結果.



(a) レベル 1

(b) レベル 2



(c) レベル 3

図 2 各レベルのサンプル画像

レベル 1 図 2(a) のように、複数の物体 (移動物体を含む) を任意の背景に合成した動画画像を入力として使用する。また、初期フレームにおける移動物体の位置と大きさを表す矩形が与えられ、初期フレームにおける移動物体の位置と大きさの情報を追跡に利用できる。ただし、移動物体の大きさの変化、オクルージョン (移動物体の部分的な隠れ)、見えの変化、は含まれない。

レベル 2 図 2(b) のように、ビデオカメラを使用して撮影した動画画像を入力として使用する。レベル 1 とは違い、初期フレームにおける移動物体の位置と大きさを表す矩形は与えられない。そのため、動画画像中からの移動物体の検出・追跡の両方を実現する必要がある。ただし、オクルージョン (移動物体の部分的な隠れ) は含まれておらず、移動物体に大きさの変化や見えの変化が少し含まれる。

レベル 3 レベル 2 と同じビデオカメラで撮影した動画画像を入力として使用する。また、レベル 2 と同様に初期フレームにおける移動物体の位置と大きさを表す矩形は与えられない。さらに、オクルージョン (移動物体の部分的な隠れ) が含まれており、移動物体の大きさの変化や見えの変化はレベル 2 と比べて大きい (図 2(c))。

これら全てのレベルにおいて、動画画像の撮影はカメラを固定した状態で行い、移動物体の数は 1 つとした。また、各レベルで使用する動画はフレーム番号 0~100 までの 101 枚の画像から構成されている。そして、プログラムの性能は、フレーム番号 1~100 に対してプログラムが出力する検出枠と正解枠の重なり率により評価した。

2.2 スケジュール

本年度のアルゴリズムコンテストは以下のスケジュールで実施した。コンテストホームページの公開およびアルゴリズムの応募は、例年と同様に大学の講義開始時期にあわせて行い、講義の演習課題として取り上げていただけるよう配慮した。

- 公開開始：2010年4月27日（火）
- 応募開始：2010年5月31日（月）
- 応募締切：2010年7月29日（木）
- 客観審査：2010年7月30日（金）～8月3日（火）
- 主観審査：2010年8月4日（水）～8月17日（火）
- 最終審査：2010年8月19日（木）～8月24日（火）
- 結果発表：2010年9月7日（火）^(注1)

2.3 応募方法と応募状況

今年度のアルゴリズムコンテストに応募する際は、2.1で述べた動画像中の移動物体を囲む矩形を各フレーム単位で出力する関数と、アルゴリズムのアイデアや処理内容を説明する説明書（A4、2～4ページ）、の2点の提出を求めた。今年度は、新しい試みとしてC++言語（ANSI/ISO C++準拠）で作成したプログラムの応募にも対応した。これまでのアルゴリズムコンテストでは、C言語（ANSI-C準拠）による実装のみを受け付けていたが、C++言語を利用したいという多数の要望のもと、新たにC++言語への対応を行った^(注2)。また、昨年のアルゴリズムコンテストで利用可能であったOpenCV [2]に加え、MIST（Media Integration Standard Toolkit）[3]ライブラリも利用可能とした。

今年度の課題である移動物体の追跡および検出に関する技術は、パターン認識・メディア理解分野で広く研究されているものであり、多くの若手研究者や学生に興味を持ってもらえるテーマであると思われる。そのため、多数の応募者を募ることができ、応募件数は歴代のアルゴリズムコンテスト [4] の中で2位であった。応募アルゴリズムの件数と内訳は以下の通りである。ただし、同一の応募者が複数レベルに応募することも可能なため、応募者数と応募件数は一致しない。また、複数人のグループで応募した場合は1グループを1名として数えた。

【応募総数（アルゴリズム別）】：73件

レベル1：33件、レベル2：20件、レベル3：20件

【応募者数】：59名^(注3)

学部2年（3名）、学部3年（6名）、学部4年（19名）、修士1年（17名）、修士2年（8名）、博士1年（1名）、博士2年（1名）、博士3年（2名）、高専5年（1名）、高専専攻科1年（1名）

【応募者所属】：19施設^(注4)

立命館大学（14）、奈良先端科学技術大学院大学（9）、筑波大学（5）、大阪大学（5）、広島市立大学（4）、宮崎大学（4）、岡山県立大学（3）、中京大学（2）、岐阜大学（2）、北海道大学（2）、法政大学（1）、東京大学（1）、東京工業大学（1）、舞鶴

(注1)：FIT2010 イベント企画にて表彰式と受賞アルゴリズム紹介を実施。

(注2)：C++言語を利用した応募総数は21件。

(注3)：グループでの応募は代表者のみを集計

(注4)：グループでの応募は1件として集計

工業高等専門学校（1）、佐世保工業高等専門学校（1）、電気通信大学（1）、大阪工業大学（1）、千葉大学（1）、愛媛大学（1）

2.4 応募アルゴリズムの種類と傾向

本年度のアルゴリズムコンテストは物体追跡をテーマとし、動画像中からの移動物体の検出と追跡を実現するアルゴリズムを募集した。今回の問題設定では、どのレベルにおいても動画像中の移動物体は1つのみであり、移動物体が映っていないフレームは存在しない。そのため、背景差分を採用するアルゴリズムが多数を占めた。また、高速化を目的とした動き予測などの工夫を行っているアルゴリズムも多く見られた。以下では各レベルにおける応募アルゴリズムの種類と傾向をまとめる。

まずレベル1では、固定背景上に移動物体が合成されたCG画像系列を入力動画像とし、初期フレームにおける移動物体を囲む矩形情報も与えられる。そこで、サンプルプログラムでは探索範囲に制限を設けたテンプレートマッチングのソースコードを提供した。そのため、応募アルゴリズムの多くはこのテンプレートマッチングをベースとしており、フレーム間差分を利用した動き予測により高速化の工夫をするもの、テンプレートマッチングの探索順を工夫するもの、などが見られた。一方、入力動画像の全フレームから背景画像を作成し、その背景を利用した背景差分により移動物体を検出するアルゴリズムも多数見られた。このアプローチは、レベル2やレベル3でも多く採用されており、本年度の課題設定では比較的簡単に実現できる精度の良い手法であったと思われる。またわずかではあるが、部分テンプレートを照合に利用することで高速化を図るものも見られ、その中でも最も高速なアルゴリズムを提案した中京大学のグループが優秀賞に選ばれた。

レベル2とレベル3は、実環境で撮影した動画像を入力とするため、照明変化や影の影響への対処が大きな課題であった。残念ながら、レベル3のオクルージョンの問題に積極的に取り組んだ応募はわずかであった。これらのレベルに応募されたアルゴリズムのほとんどは背景差分を利用しており、どのように影領域の除去を行うかが手法の違いとして見られた。影領域の除去には、HSV色空間を利用するもの、エッジ領域の形状を利用するもの、移動物体と影を分離するしきい値を動的に決定するもの、ポスタリゼーションの手法を利用するもの、などさまざまな工夫が見られた。また高速化に関しては、フレーム間差分の結果を利用するものがほとんどであり、物体の移動方向を予測して探索範囲を制限するものが多数を占めた。応募されたアルゴリズムのほとんどはオーソドックスな手法を採用しており、比較的新しい物体追跡技術に挑戦した応募はパーティクルフィルタを用いたもの1件だけであった。

上述のように、レベル2とレベル3に応募されたアルゴリズムには、おおむね似た傾向が見られた。これは、昨年度（2009年度）のレベル3の課題が非常に難しかったため（応募はわずか1件のみ）、本年度はレベル3の難易度を下げ、比較的取り組みやすい問題設定にしたことが原因だと考えられる。結果として、レベル3への応募は20件と大幅に増加したが、レベル2とレベル3の差が少なくなってしまった。レベル2とレベル3の問題設定の工夫に関しては、次年度以降の課題としたい。

2.5 審査方法

応募アルゴリズムの審査は、アルゴリズムの追跡性能と処理速度を定量的に評価する客観評価、アルゴリズムの独自性やおもしろさなどを評価する主観評価、実行委員全員による最終評価、の3段階で行った。

客観評価： アルゴリズムコンテストのホームページで学習用として配布している動画像（各レベルにつき5種類）とは別に、評価用の動画像（各レベルにつき2種類）を用意し、プログラムが出力する移動物体を囲む検出枠と正解枠との重なり率を計測した。また、すべての応募プログラムはコンテストのホームページに掲載しているLinux環境で実行し、プログラムの実行にかかる処理時間を計測した。これら、重なり率と処理時間の2つを客観評価の指標とした。

主観評価： 応募時に提出されたアルゴリズムの説明書、ソースコード、プログラムの出力結果、客観評価の結果、を審査委員に提示し、10段階の主観評価（1～10点）とコメントをいただいた。

最終評価： 客観評価と主観評価の結果に基づき、受賞アルゴリズム案を作成し、審査員全員による議論を経て入賞者を決定した。

2.6 審査結果

前節に示した審査方法により、最優秀賞（1件）、優秀賞（2件）、入賞（3件）、審査員特別賞（1件）、を決定した。最終審査結果は以下の通りである。

- a) 最優秀賞（1件）
 - 梅田一彰（岡山県立大学, M1, レベル3）
- b) 優秀賞（2件）
 - 斎藤正孝, 秋月秀一, 岡明也, 柴田悠太郎（中京大学, B3, レベル1）
 - 山下真吾, 藤賢一郎（宮崎大学, M2・M1, レベル2）
- c) 入賞（3件）
 - 糟谷望, 末次祐樹, 水流弘達, 君島直城, 田邊健（筑波大学, D2・M2・M1・B4・B4, レベル2）
 - 高井翔太（立命館大学, B4, レベル3）
 - 佐藤竜太, 剣持星二, 長島正典, 王ジン, 柏熊淳也（筑波大学, M2・M2・M1・M1・B4, レベル3）
- d) 審査員特別賞（1件）
 - 丸山拓馬（電気通信大学, M1, LV3）

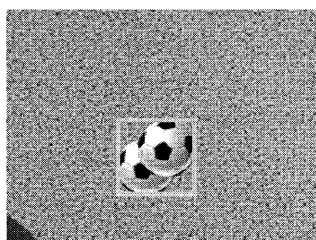


図3 前フレームとのフレーム間差分矩形

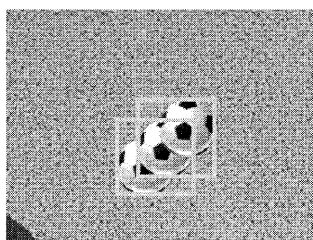


図4 フレーム間差分矩形の重ね合わせ

2.7 組織

これまでに述べた、課題内容の決定、各種サンプルの準備、評価方法の策定、実際の審査、はすべて本コンテストを主催するPRMU研究専門委員会、ならびにその中から選ばれた下記メンバから構成される実行委員会が担当した。

実行委員長

美濃 導彦（京都大学）

実行委員

幹事 出口 大輔（名古屋大学）

村瀬 洋（名古屋大学）

石井 育規（パナソニック株式会社）

佐藤 清秀（キヤノン株式会社）

帆足 啓一郎（KDDI研究所）

山口 晃一郎（豊田中央研究所）

喜多 泰代（産業技術総合研究所）

清水 郁子（東京農工大学）

杉山 将（東京工業大学）

高橋 隆史（龍谷大学）

新田 直子（大阪大学）

羽下 哲司（三菱電機株式会社）

岩井 俊雄（大阪大学）

田中 宏（富士通研究所）

佐藤 洋一（東京大学）

柴藤 稔（NTTドコモ）

山口 修（東芝）

神原 誠之（奈良先端科学技術大学院大学）

玉木 徹（広島大学）

藤吉 弘亘（中部大学）

3. 最優秀賞：梅田一彰（応募レベル3）

3.1 はじめに

移動物体を追跡する際、有効な手法として背景差分とフレーム間差分が挙げられる。しかし本課題では、すべてのフレームに移動物体が存在するため、背景画像としていずれかのフレーム画像を選択することは不適切である。また、自然画像に対応するためには効果的なノイズ処理が必要であり、全画素の差分処理では正確性、計算量の両面で非効率である。そこで提案手法は、前フレームとのフレーム間差分矩形、後フレームとのフレーム間差分矩形を前フレーム間差分矩形を中心に探索し、両差分矩形の共有部分を出力矩形とする効率的なアルゴリズムを提案する。

3.2 アルゴリズム

3.2.1 フレーム間差分

図3、4はフレーム間差分矩形の重ね合わせにより算出する出力矩形の例である。フレーム間差分矩形は2フレームのRGBそれぞれの差をとり、RGBのいずれかがしきい値（しきい値A）より大きな差分をとる画素を検出し、ノイズ処理を行って矩形で囲んだものである。

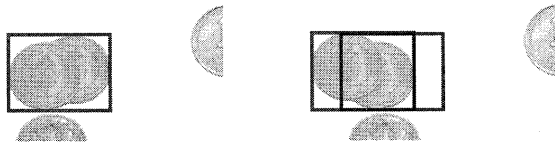


図5 前フレームとのフレーム間差分矩形
図6 フレーム間差分矩形の重ね合わせ誤差

3.2.2 例外処理

a) 最終フレームの矩形

提案手法は後フレームが存在しない最終フレームには適用できない。このため、物体の大きさは前フレームと大きく変化はしないと仮定し、前フレームと同様の幅と高さとする。また、差分矩形の2辺は前フレームの移動物体の矩形、他の2辺は現フレームの移動物体の矩形であることを利用し算出する。

b) 差分矩形を抽出できない場合

移動物体の大きさ、移動速度により差分矩形を算出できない場合が存在する。これは、

- 探索範囲 (3.2.3 (b) 参照) を越える移動である場合
- 動きが少なく、移動物体と認識できない場合
- 差分に対するしきい値 (しきい値 A) が適切でない場合

これらに対しては差分矩形が抽出できなかった場合に、まず移動物体が探索範囲を越える移動をした場合を考え、画像全体で差分をとり、差分矩形の作成を行う。これにより抽出できない場合は動きが少ないと考え、前フレームと同様な矩形を出力矩形とし、さらに次フレームでも認識ができない場合はRGBの差分に対するしきい値 (しきい値 A) を変更する。

c) 一定の動きをする場合

図5, 6は矩形の重なりだけでは誤差を生じる例である。このように凸となる動きをした際には凹側の1辺だけ大きく矩形をとってしまう。このような場合には、現フレームの画像の凹側の矩形部分から内側に1列 (又は行) ごとに、列 (又は行) 中の画素のRGB値にしきい値より大きな変化が無ければ矩形を縮小することで誤差を軽減する。ここでのしきい値は誤差を軽減できればよいため、しきい値 A より小さなしきい値とする。

3.2.3 処理量の削減

a) 矩形の探索法

処理量の削減には比較する画素数の削減が有効である。まず本手法は前フレームとの差分矩形と後フレームとの差分矩形の重なりを検出するものであるため、前フレームとの差分矩形位置を中心として後フレームとの差分矩形探索を行う。まず、前フレームとの差分矩形から外側に1列 (又は行) に対して現フレームと後フレームとの差分をとる。このときしきい値 A 以上のRGB値の変化が3画素連続で検出された場合、移動物体によって変化が起こったとし、しきい値 A 以上のRGB値の変化が3画素連続で検出されなくなるまで矩形を拡大させる差分矩形拡大処理を適用する (図7参照)。その後、1列 (又は行) も移動物体によってRGB値に変化が起こっていないとした辺

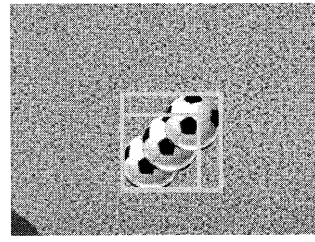


図7 差分矩形拡大処理

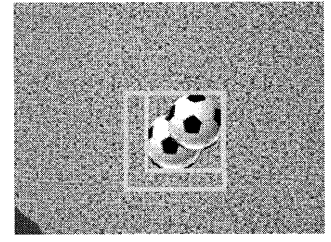


図8 差分矩形収縮処理

の内側1列 (又は行) に対して、現フレームと後フレームとの差分をとる。このときしきい値 A 以上のRGB値の変化が3画素連続で検出されない場合、移動物体によって変化は起こっていないとし、しきい値 A 以上のRGB値の変化が3画素連続で検出されるまで矩形を縮小させる差分矩形収縮処理を適用し (図8参照)、後フレームとの差分矩形探索を行う。本手法により、差分をとる計算量の大幅な削減が行えるとともに移動物体周辺のノイズ処理だけで済むため、ノイズによる精度の低下を防ぐことが可能となる。

b) 移動速度を考慮した参照画像の更新

さらなる処理量の削減を図るために、参照画像 (前フレーム画像) の更新の際に、移動速度と物体の大きさにより部分的に更新を行う。具体的には、前フレームとの差分矩形周辺に前フレーム矩形の長辺 $+ \alpha$ の領域を探索範囲とし、その範囲に対し更新を行う。このとき移動物体が大きい場合は探索範囲が大きくなる。ただし、速度が遅い場合には探索範囲を縮小することが可能である。そこで、移動方向と速度に応じて更新領域を縮小する。

3.3 まとめ

フレーム間差分矩形を重ね合わせ、両差分矩形の共有部分を出力矩形とする手法とその処理の削減手法を提案した。提案手法を提供されたサンプル画像に適用した結果、いずれの画像についても安定した検出が行え、前フレームとのフレーム間差分矩形を中心に探索を行うことで、処理の大幅な削減の実現を確認した。追跡精度を下げた要因である物体の影への対策が今後の課題である。

4. 優秀賞：斎藤 正孝， 秋月 秀一， 岡 明也， 柴田 悠太郎 (応募レベル1)

4.1 はじめに

本コンテストのレベル1の課題は照明変化がないなど、一見、比較的容易に見える。そこで我々は「信頼性を落とさず極限まで高速化する」ことに挑戦することにした。

初期フレームにおける移動物体の位置と大きさがあらかじめ与えられていることから、追跡にはテンプレートマッチングが有効であると考えられる。しかし、提供されたサンプルプログラムでは、探索範囲が前フレームにおける結果をもとに固定的に選択されており、またマッチングにおいてもテンプレートの全画素を用いていることから計算コストが増加するという問題があった。

そこで、テンプレート画像の中から濃度共起発生確率の低い

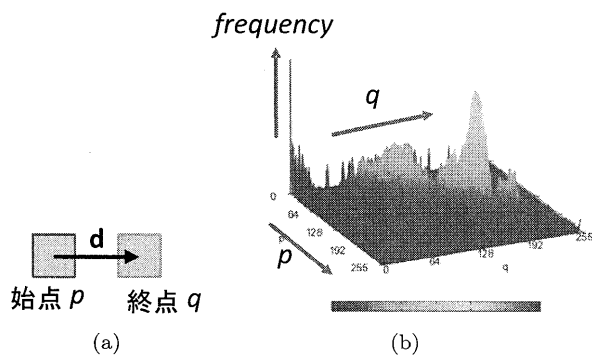


図9 画素ペアと共起ヒストグラムの例

画素パターンを自動的に抽出することによって、照合に有効な必要最小限のテンプレート点を選択し、さらにマッチングに最適な色プレーンの自動選択、探索範囲の動的決定、テンプレート内のアクセス順序の考慮、という計4つのアイデアを盛り込むことにより、移動物体を高速にトラッキングする手法を提案する。

4.2 提案アルゴリズム

4.2.1 追跡対象を代表するごく少数の画素のみの選択

与えられたテンプレート画像から、実際の照合に使用する少数の画素を選択することにより、テンプレート点の大幅な削減をおこなう。

テンプレート点は、テンプレート画像内における画素ペアの共起ヒストグラムを生成し、この共起ヒストグラムを基に発生頻度の低い画素パターンを選択する。図9に画素ペアと共起ヒストグラムの例を示す。画素ペアは図9(a)のように画素 p から変位ベクトル d 離れた画素 q によって構成される。この p 、 q の濃度値をインデクスとした図9(b)の濃度共起ヒストグラムにおいて、頻度値の低い画素パターンをテンプレート点として選択する。

このアルゴリズムによりテンプレート点を30画素（データセット2のテンプレート画像における約0.3%）まで削減することに成功した。

4.2.2 自己相関類似度マップ分析に基づく最適な色プレーンの選択

毎フレームRGB全ての色プレーンを相違度計算に使用するのには計算コストがかかるため、最適な色プレーンを自動選択することにより計算コストを1/3に削減する。

最適な色プレーンは、初期フレームにおける正解座標周りの8画素近傍領域に着目し、各色プレーンのみを用いたテンプレートマッチングの相違度に基づいて決定する。これは初期フレームの情報のみを用いた自己相関であるが、この類似度マップを分析し、正解座標周りの相違度が一番大きくなる色プレーンが誤対応を起こしにくい有効なプレーンであると考え、これを選択することにした。

4.2.3 探索範囲の動的決定

入力画像の全範囲をテンプレートマッチングにより探索することは、計算コストの面から考えて適切ではない。そこで、各フレーム間の物体の移動距離が微小であることに着目し、探索

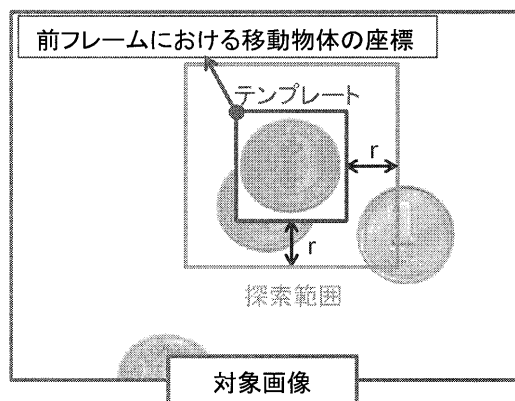


図10 探索範囲の決定方法

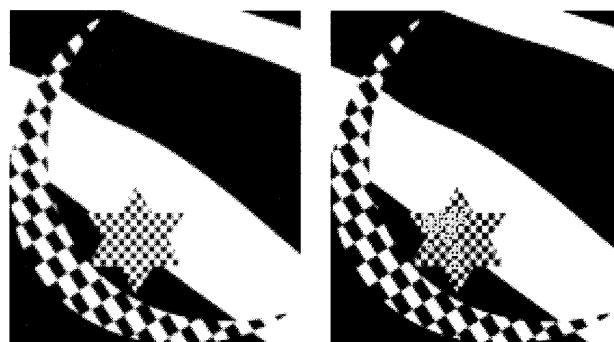


図11 評価用データセット7

範囲を動的に決定する。

まず、対象フレームの1フレーム前と2フレーム前のフレームにおける移動物体の座標から移動距離 r を算出する。次に、図10のように移動距離 r だけ各フレームにおいて物体の座標から探索範囲を広げる。これにより、探索範囲の無駄を削減し、高速なマッチングをおこなう。

4.2.4 物体の移動方向を考慮したテンプレート内のアクセス順序の決定

今回我々はテンプレートマッチングの類似度にSSDAを用いているため、アクセス開始位置と方向は高速化において非常に重要である。

4.2.3節より、絞り込まれた探索範囲を効率よくアクセスするために、対象フレームに対して1フレーム前と2フレーム前のフレームから求められる移動方向から、アクセス開始位置と方向を決定する。つまり、物体が右上に移動したと予想されるときは右上から左下方向へアクセスするなど、予測方向からアクセスを行うことで、SSDAの打ち切りのしきい値を早期に低くしマッチングを高速化する。

4.3 考察

本アルゴリズムによって、評価用データセット1~6においてはトラッキングすることに成功したが、評価用データセット7においては失敗してしまった。図11に評価用データセット7におけるテンプレート画像と選択されたテンプレート点を示す。

テンプレート画像中の追跡対象である月領域よりも背景である星領域に発生頻度の低い画素があり、背景領域をテンプレ



図 12 離れたフレーム間での差分による問題

ト点 (図 11(b) の黄点) として選択してしまっている。この問題に対しては、あらかじめテンプレート画像中の前景領域を推定し、テンプレート点を前景領域から優先的に選択することが有効であると考えられるが、これについては今後の課題と考えている。

5. 優秀賞：山下 真吾，藤 賢一郎 (応募レベル 2)

5.1 プログラムの概要

本プログラムは、背景差分処理とエッジ検出処理によって特徴量を生成し、その特徴量をもとに移動物体を探索し、追跡を行うアルゴリズムである。

5.2 特徴量の生成

本プログラムの特徴量は背景差分処理とエッジ検出処理により画像の各画素で生成され、「移動物体の存在を示す」情報となる。ここで、本節と 5.3 節の説明では、 N 番目のフレームにて物体追跡を行う場合について述べる。特徴量の生成手順を以下に示す。

- (1) 画像のグレイスケール化処理
- (2) N 番目と 0 番目のフレームの背景差分処理を行い、値が得られた画素を「差分あり」画素とする。
- (3) 「差分あり」画素領域についてエッジ検出処理を行う。エッジが検出された画素を「特徴あり」画素とする。本プログラムでは、この「特徴あり」画素の集合体を移動物体としている。

ここで手順 2 について具体的に述べる。一般的な背景差分処理では前フレームとの処理を行うことが多い。しかし、手順 2 では背景差分処理の基準として前フレームではなく、0 番目フレームを使用している。それは、前フレームからの物体の移動が小さい場合、物体を動かないもの、つまり、背景として認識してしまう恐れがあるためである。

離れたフレーム間での差分をすることで欠点も生じる。例として、図 12 に問題となったサンプル画像 (配布データ 2-2, $N = 55$) を示す。ここで図 12 に示された 2 つの領域は手順 2 で得られた「差分あり」領域を表している。この領域に対して手順 3 ではエッジ検出処理を行うが、基準となる 0 番目側の領域 (図 12 左側) にエッジが発生してしまう。これは 0 番目フ

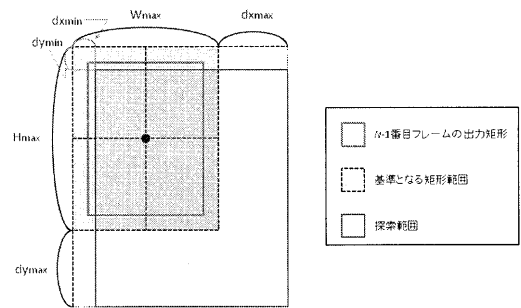


図 13 探索範囲設定の様子

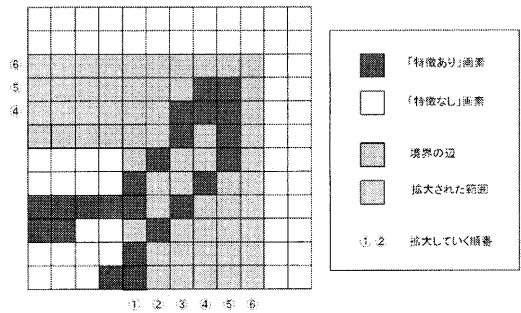


図 14 探索範囲が広がる様子

レームで移動物体の背後にエッジを検出してしまうような背景が隠れている場合があるためである。そこで、基準となる 0 番目フレーム画像の 0 番目側領域について補正を行う。具体的には、この領域の画素情報を毎フレーム書き込んでいく。この補正によって N 番目と 0 番目フレームの 0 番目側領域の画素情報は同じになるため、問題となるエッジ情報を検出させなくできる。

5.3 探索範囲の設定

本章では特徴量生成における探索範囲の設定を行う。図 13 に探索範囲の絞り込みの様子を示す。まず $N - 1$ 番目までで最大となる出力矩形の幅 W_{max} 、高さ H_{max} となる基準矩形を設定する。このとき、基準矩形の中心位置は $N - 1$ 番目の出力矩形の中心に合うように設定する。次に、 $N - 2$ 番目から $N - 1$ 番目までの移動距離 x_{dire} , y_{dire} 、また、 $N - 1$ 番目までで最大、最小となる移動距離 dx_{max} , dy_{max} , dx_{min} , dy_{min} を算出し、この x_{dire} , y_{dire} の正負に応じて基準矩形を伸縮させる。例えば、 $x_{dire} \geq 0$ ならば矩形右端から dx_{max} だけ伸長させ、矩形左端から dx_{min} だけ縮小させる。 $y_{dire} \geq 0$ ならば矩形下端から dy_{max} だけ伸長させ、矩形上端から dy_{min} だけ縮小させる。なお、符号が逆の場合は伸縮方向が逆になる。以上の処理によって探索範囲の絞り込みを行う。しかし、 $N - 1$ 番目までの経験を超えるような大きさや移動距離が発生した場合、探索範囲の外に移動物体が存在してしまう危険性がある。そこで、図 14 のように探索範囲を広げる処理を行う。探索範囲の境界 4 辺に属する画素で「特徴あり」が検出されたとき、境界の辺から 1 画素分外側の辺を新たな探索範囲として探索を行う。この処理を繰り返し、「特徴あり」が検出されなくなるまで続けることで取りこぼしを防いでいる。

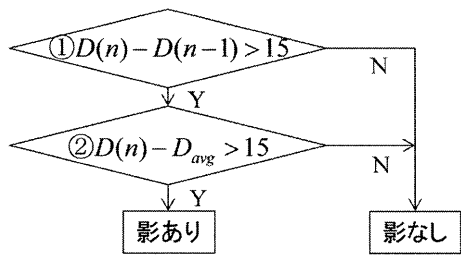


図 15 影の有無識別のフローチャート

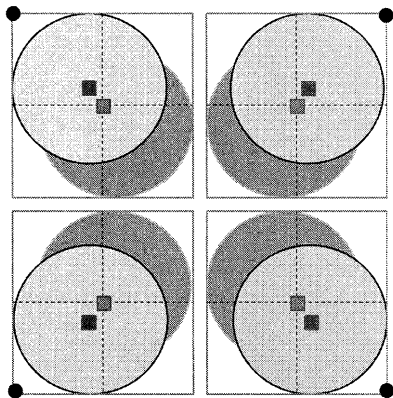


図 16 影が左上, 左下, 右上, 右下にある場合

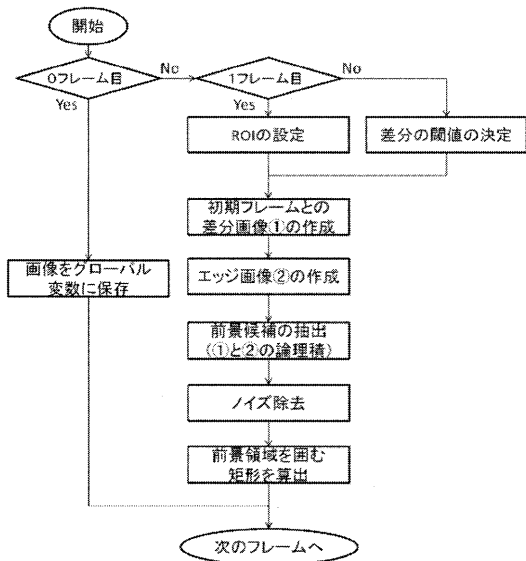


図 17 アルゴリズムのフローチャート

5.4 影による追跡精度低下への対処

物体に影が発生した場合、影は移動物体とともに変化していくため、影を物体として判定してしまうことがある。そこで本章では、以下の2点に注目し、物体のみを抽出するアルゴリズムについて述べる。なお、ここで述べる矩形とは、前章までの手法で抽出した物体を囲った矩形である。

- (1) 矩形内の影の有無
- (2) 影がある場合の物体の大まかな位置検出

(1)は、図 15 の流れに従い、影の有無を識別する。ここで用いる特徴量 D は出力矩形の斜辺距離である。条件 1 は前フレームとの比較で、物体が大きく変化したかを判別する。条件

1 を満たした場合、(A) 影が発生している、(B) 物体そのものが大きく変化している、場合が考えられる。しかし、Lv2 では物体が急激に変化するケースは少ないため、条件 2 のように前フレームまでの D の平均値 D_{avg} との比較を行うことで (B) を除外できる。

(2) は影の位置を仮定することで物体の位置を決定する。図 16 のように、影が左上、左下、右上、右下にある場合を考えると物体の中心と矩形の中心との位置関係がそれぞれ違っていくことがわかる。この違いから●点を基準に、前フレームの幅、高さで矩形を設定する。

5.5 考 察

本プログラムは背景差分処理とエッジ検出処理によって特徴量を生成し、移動物体追跡を行った。その際、離れたフレーム間差分や影の問題への対応、また、探索範囲を絞り込むことで、処理時間を短く、かつ、精度を向上させることができた。

6. 入賞：糟谷 望, 末次 祐樹, 水流 弘達, 君島 直城, 田邊 健 (応募レベル 2)

6.1 概 要

本アルゴリズムは、様々なシーンで利用できる事を目指し、未来の情報を用いずにシーンに対して頑健で高精度かつビデオレートでの処理を目指す。前景抽出のしきい値を適応的に決定し、2種類のノイズ除去法を用いることでシーンに頑健にし、初期フレーム画像との差分とエッジを組み合わせることにより、精度の高い移動物体追跡を実現し、探索領域を絞り込むことで、処理の高速化を行う。

処理の流れを図 17 に示す。初期フレームでは、後に差分画像を作成するために入力画像を保存する。 t ($t \geq 1$) フレーム目では、移動物体の位置を求めるため、0 フレーム目と t フレーム目の画像の差分をとり、二値化処理を行う。この際のしきい値は画像毎に動的に変化させる。ここで、得られた差分画像では、 t フレーム目の移動物体だけでなく 0 フレーム目の移動物体も検出されてしまうため、 t フレーム目のエッジ画像との論理積をとることで、 t フレーム目の移動物体のみを抽出する。さらに、光源環境の変化などの影響によるノイズ除去を行なうことで追跡対象物体を検出する。全ての処理は、高速化のため、グレースケールで計算を行い、 $t-1$ フレーム目での移動物体の位置と速度を用いて ROI 領域を設定し、計算コストを削減する。

6.2 しきい値の適応的決定

初期 2 フレームの画像を用いて、前景抽出の際のしきい値を決定する。2 枚の画像から差分画像を作成し、しきい値を 0 から 1 ずつ増加させながら二値化を行う。二値化で得られた前景領域の面積、重心、前景画素の分散の変動が十分に小さくなった際の値をしきい値とする。

6.3 前景候補の抽出

t フレーム目における移動物体を検出するため、0 フレーム目との差分をとる。ここから、 t フレーム目の移動物体のみを抽出するために、 t フレーム目の画像について、ゼロ交差法を用いてエッジを抽出し、差分画像との論理積をとることで、前景候補を抽出する。

6.4 ノイズ除去

前節で記述した手法で前景候補を検出すると、光源の変化などに起因するノイズが存在するため、それらを取り除く必要がある。そこで2つのノイズ除去法を用いる。

6.4.1 小領域の類似度を用いたノイズ除去

1つはごま塩ノイズのような小さなノイズを除去するために、小領域の類似度を用いたノイズ除去を行う。注目点近傍の小領域をテンプレートとし、前フレームとの類似度を求め、類似度が高い場合には前フレームからの変化が少ないため背景とする。ただし、移動物体の位置変化が小さい場合には、この手法では前景を背景に分類してしまう。そこで、前フレームでの移動物体を囲む矩形内のみ初期フレームとの類似度を用いる。

6.4.2 移動物体の重心からの距離を用いたノイズ除去

前景領域の背景領域への映りこみなどに対処するため、重心からの距離と分散を用いてこれらを除去する。移動物体の重心を算出し、重心からの距離の分散を x 方向、 y 方向それぞれについて求める。これらの分散に対して重心から一定以上離れている画素をノイズと見なし、背景に分類する。

6.5 探索領域の限定

差分画像およびエッジ画像を作成する際、ROI 領域を設定し、探索領域を限定することで処理の高速化を行う。

物体の移動を等速直線運動であると仮定し、 $t-1$ フレームの移動物体を囲む矩形を B_{t-1} 、 t フレーム目の位置へ移動したものを矩形 B_t とする。移動物体が停止した場合、向きを変えた場合に対応するため、探索領域はこの B_t と B_{t-1} を含む矩形を1.5倍に拡大したものとする。

6.6 まとめと結果

本アルゴリズムは、未来の情報を用いないこと、シーンによらない頑健性をもつこと、ビデオレートでの処理が可能であることを目指し、しきい値の適応的決定、2種類のノイズ除去法の組み合わせによりそれらを実現した。その結果として、追跡精度は86.6と高精度であり、処理時間は100フレームで1.346秒とビデオレートを実現できた。

7. 入賞：高井 翔太（応募レベル3）

7.1 アルゴリズムの概要

本アルゴリズムは、正規化距離を用いることで影や照明変化に頑健な移動物体検出を行う。算出した正規化距離値から移動物体ごとに適応的な変動しきい値を決定する方法も提案する。また、正規化距離による検出を行うには背景画像が必要なため、メディアンを考えを基にした背景画像の作成をおこなう。

7.2 アルゴリズムの詳細

7.2.1 背景画像の作成

100枚の画像から背景画像を作成する。手順を以下に示す。

(1) 100枚の画像をすべて輝度値のグレースケール画像に変換する。

(2) 背景画像の各座標における輝度値を決定するために、各座標において100枚の画像から輝度値のヒストグラムを作成する(図18)。そして、メディアンを考えを基に輝度値の出現回数を順にカウントしていき、出現回数が50回を超えたとき

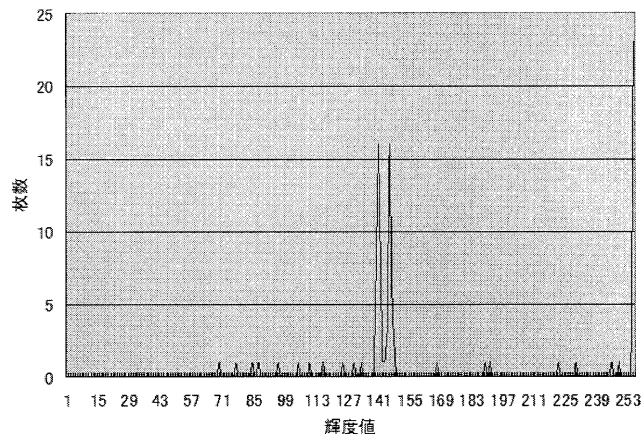


図18 ある座標における輝度ヒストグラム

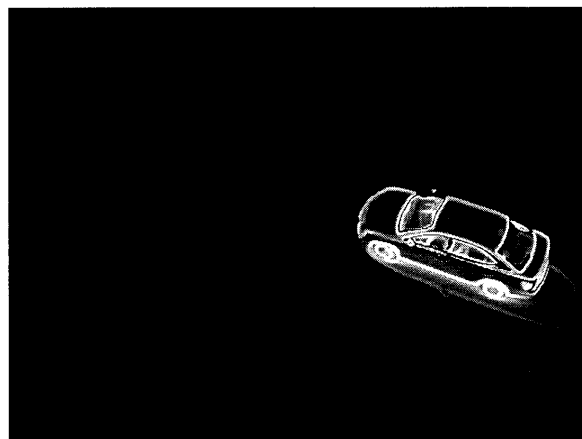


図19 正規化距離画像

の輝度値を背景画像の輝度値とする。

7.2.2 移動物体検出のため正規化距離の算出

作成した背景画像と各フレーム画像で正規化距離を求める。手順を以下に示す。正規化距離を計算するために、まず、各座標を中心とした 5×5 画素の領域を考え、その領域を25次元のベクトルで表現する。ベクトルの各要素は領域中の輝度値であり、フレーム画像と背景画像の領域に対応するベクトルを $\mathbf{i}_{(u,v)}$ 、 $\mathbf{b}_{(u,v)}$ とすると、正規化距離 $ND(\mathbf{i}_{(u,v)})$ は次式で表される。

$$ND(\mathbf{i}_{(u,v)}) = \left\| \frac{\mathbf{i}_{(u,v)}}{\|\mathbf{i}_{(u,v)}\|} - \frac{\mathbf{b}_{(u,v)}}{\|\mathbf{b}_{(u,v)}\|} \right\| \quad (1)$$

正規化距離のとり値の範囲は0.0~1.0である。正規化距離により求めた値を画像として表示したものを図19に示す。図19は正規化距離の値が大きいほど白(255)となる。

7.2.3 変動しきい値を用いた移動物体の領域の検出

図19に示すように、正規化距離を用いることで移動物体を検出することができる。しかし、正規化距離は移動物体だけではなく影領域も同時に検出してしまふ。そのため、適切なしきい値を画像ごとに決めることで影の領域と移動物体の領域を分離する必要がある。影の大きさは移動物体の大きさと環境光によって変化する。移動物体が小さければ、環境光の影響により小さな影が現れる(図20)。室内にて画像を撮影した場合にお

表 1 出現数に応じたしきい値

出現数	しきい値	影領域
0~500	0.120	大 ↑
500~1000	0.140	
1000~1500	0.160	
1500~2000	0.180	
2000~3500	0.200	
3500~5000	0.215	
5000~7000	0.230	
7000~9000	0.245	↓ 小
9000~	0.205	

いては光源が多数存在するため環境光の影響が大きく小さな影が現れる (図 21)。以上二つの場合はどちらも影の領域の正規化距離の値が小さい。移動物体が大きければ、環境光の影響を受けにくいので大きな影が現れる。この場合は正規化距離の大きい値が多く含まれる (図 22)。これらのことから、小さな影のときは、しきい値を低くし、大きな影のときはしきい値を高くすることで移動物体と影を分けることが出来る。しきい値の決定方法を以下に示す。

(1) 画像毎に影の領域の大きさを求める。正規化距離の値が 0.02~0.10 の範囲であれば、影の領域であると仮定し、その範囲の値を持つ正規化距離の出現数の総計値を求める。

(2) 影の領域が小さく数が少なければしきい値を低くし、影の領域が大きければしきい値が高くなるようにしきい値を 0.12~0.25 の範囲で変動させる。影の大きさごとのしきい値を表 1 に示す。

7.3 まとめ

本稿では、照明変化に頑健な移動体検出手法を提案した。始めに、メディアンを考えを基にした背景画像の作成により、ノイズのない背景画像を作成する。次に正規化距離を用いた移動物体の候補領域を求める。候補領域から移動物体だけの領域を検出する際に、移動物体の大きさと影の明るさの影響を考慮した適切なしきい値の決定を行う事により移動物体の領域を決定した。提案手法は、レベル 2, 3 のサンプル動画画像の全てにおいて正しい移動物体検出結果を得ることが可能であった。

8. 入賞：佐藤竜太, 剣持星二, 長島正典, 王ジン, 柏熊淳也 (応募レベル 3)

8.1 概要

本アルゴリズムの基本的なポリシーとして、

- (1) レベルに関わらず同じアルゴリズムで動作すること。
- (2) 時系列データを全て使用して計算を行うこと。
- (3) 自然画像にも対応できること。

を目標に挙げた。それぞれ以下を対応策としている。

(1) オクルージョンを考慮する。レベル 1 での初期領域は使用しない。

(2) 最初に全フレームから背景モデルを構築し、モデルから

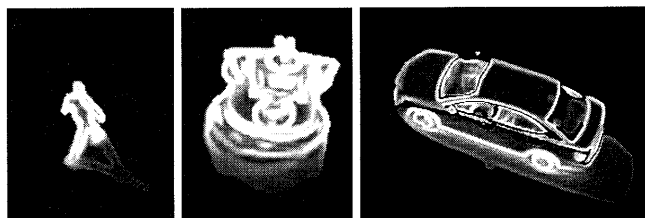


図 20 人の影 図 21 室内の影 図 22 車の陰

の計算を行う。

(3) 影の中でも特に影響が大きいものを除去するなどの領域の改善を行う。

以下それぞれの対応策について順に説明する。最初に 8.2 節において全体の構成を説明した後、8.3 節で背景モデルの構築方法について述べる。8.4 節にて領域改善方法を説明して、8.5 節において結言を述べまとめとする。

8.2 全体の流れ

本アルゴリズムの入力は、時系列に並ぶ 101 フレームの画像列である。最初に、これら全ての画像における動物体以外の領域を利用することで、動物体が映り込んでいない背景画像を用意する。その上で、各フレームにおける背景画像との差分を利用し、動物体領域候補を算出する。

動物体領域候補には、影などの不要な部分が存在したり、削れてしまっている部分が発生するため、領域の改善を行いこれの外接矩形を結果とする。

8.3 背景画像の構築

0 フレーム目と 100 フレーム目を除く入力画像について、フレーム間差分により得られる動物体領域を 0、それ以外を 1 とするマスク画像 M_n (n : フレーム番号) を求める (式 (2))。

$$M_n(i, j) = \begin{cases} 1 & (|I_{n-1}(i, j) - I_n(i, j)| \leq \tau_I) \\ 0 & (|I_{n-1}(i, j) - I_n(i, j)| > \tau_I) \end{cases} \quad (2)$$

ここで I_n は n フレーム目の入力画像である。また、 $\tau_I = 15$ とした。さらにこのマスク画像中で、面積が 50 ピクセル未満の領域はノイズとして除去する。以上の処理によって、各フレームのマスク画像 M_n が得られる。

このマスク画像 M_n を用いて、各フレームにおける背景領域の平均画像を作成し、背景画像 BG とする。

$$BG(i, j) = \frac{\sum_{n=1}^{99} M_n(i, j) I_n(i, j)}{\sum_{n=1}^{99} M_n(i, j)} \quad (3)$$

8.4 領域の改善

本節では、領域の改善方法について説明する。先に述べたように、8.3 節で構築された背景画像 BG から差分を行うと、(i) しきい値の低い場合は影やノイズが発生し、(ii) しきい値の高い場合は欠損が発生する。そのため、しきい値の低い差分画像に存在しながらしきい値の高い差分画像に存在しない領域 (以下大小しきい値間差分画像と呼ぶ) が、影と欠損であることに着目し、影の影響が大きい場合に効率的に取り除く手法を用いる。

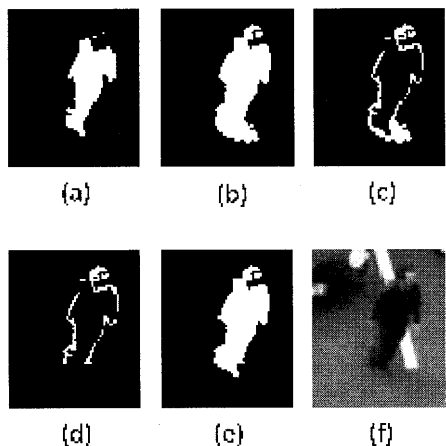


図 23 領域改善の流れ (a) しきい値高 (b) しきい値低 (c) 大小しきい値間差分 (d) 影除去 (e) 結果 (f) 原画像

まず、大小しきい値間差分画像 I_{dr} の算出方法について述べる。背景画像と入力画像との差分を低いしきい値 $\tau_{lo} = 25$ により二値化した画像 I_{lo} と、高いしきい値 $\tau_{hi} = 60$ により二値化した画像 I_{hi} に対して、各画素 p_{lo} , p_{hi} を用いて、大小しきい値間差分画像の各画素 p_{dr} は論理積により次式で計算される。

$$p_{dr} = p_{lo} \& \bar{p}_{hi} \quad (4)$$

大小しきい値間差分画像が影を含んでいる場合、その多くは重なり率の低下を招くが、手前に伸びる影が発生する場合において大きく低下する。そこで本アルゴリズムにおいては、影の除去は手前に伸びる影のみを対象とすることにした。

この発生状況においては、大小しきい値間画像は正解領域下部に大きな領域を含むことになる。そのため、本アルゴリズムでは、以下のような手法で対応することとした。

まず、大小しきい値間差分画像 $I_{dr}(x, y)$ における縦方向のプロジェクトン $P_y(y) = \sum_i I_{dr}(i, y)$ を計算する。ここで、 P_y の傾き $P'_y = \frac{d}{dy} P_y$ を用いて、以下の二条件を満たす最大の y 座標を影領域の y 座標 y_s とする。

$$P'_y(y_s) > 0 \quad (5)$$

$$P'_y(y_s - 1) \leq 0 \quad (6)$$

このようにして求められた y_s より下部に存在する領域を影領域としてその領域を削除し、この領域が削除された画像としきい値が高い画像との間で論理和をとる。これにより、性能に大きく影響する影領域を削除することができる。

各段階における画像の例を図 23 に示す。

8.5 まとめと結果

本アルゴリズムは、同一のアルゴリズムで全てのレベルに対応することを狙った。また、自然画像にも対応するために、大きくノイズを受ける部分に対して、プロジェクトンを利用した改善手法を考案した。この結果として、サンプル画像セットにおいて、すべてのレベルの平均精度が 88.8% となった。

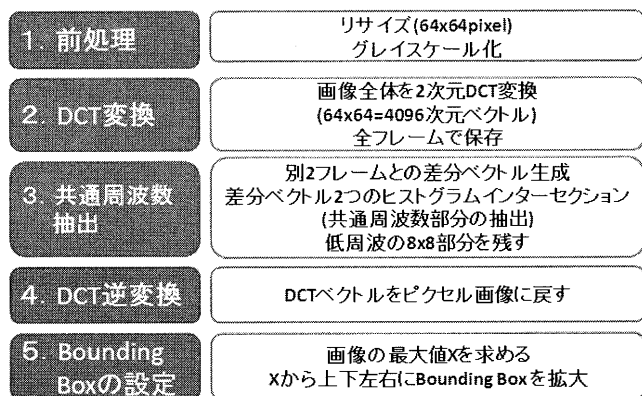


図 24 アルゴリズムの流れ

9. 審査員特別賞：丸山 拓馬 (応募レベル 3)

9.1 アルゴリズムの概要

物体を輝度値の集合と考えたとき、この輝度の波の移動を補足するのが物体追跡である。本アルゴリズムは、この物体の波という着眼点に基づいて考えた。具体的にはフーリエコサイン変換 (DCT) を施して、その低周波情報を利用することで物体のたまかな位置の判定に成功している。

9.2 アルゴリズムの詳細

図 24 にアルゴリズムの流れを示す。

基本的な構想は低周波情報を見ることで物体の位置を割り出すことにある。図 25 に基底画像の例 (特定基底のみを 1, 他を 0 として逆 DCT 変換したもの) を示す。このように基底は対応する領域を持っており、この基底を組み合わせることで周波数成分から画像を表現できる。低周波ほど対象領域 (白い部分) が広いため、これら周波数の低い基底をいくつか組み合わせれば物体の位置が分かるというアイデアである。これにより、基底の調べる数と精度がトレードオフになるアルゴリズムも構築可能だと考えられるが、本アルゴリズムではそこにまでは至っていない。実際には DCT 変換の後に低周波要素の共通成分を残し、逆 DCT 変換でピクセル画像を復元してこのピクセル画像で現れた物体領域を囲むという流れである。

手順について説明する。まずアルゴリズムは前処理を行い、 64×64 の全画像領域を対象に DCT 変換 (DCT-2) する。これを全フレームで行って DCT ベクトルを保存する。その後、判定フレームから十分に離れた別フレームを 2 つ選び出す。この選び出したフレームと対象フレームとの差分をとり、2 つ出来た差分画像の共通周波数部分を抽出する。具体的にはヒストグラムインターセクションを行うことでこれを達成する。次にローパスフィルタで低周波の 8×8 以下の DCT 基底のみを残す。残った周波数成分を逆 DCT 変換 (DCT-3) することで、対象物体部分だけ浮き彫りになる画像が出来上がる。低周波で表現すれば物体中心ほど輝度が大きくなるはずである。よって最大輝度値は対象物体の矩形に含まれているという想定のもと、上下左右に Bounding Box の拡大を行う。拡大処理はピクセル値がしきい値以下になるまで行う単純なものである。最後に得られた Bounding Box が検出物体の矩形として登録される。

9.3 結果と考察

追跡精度は平均 50.6% であった。実際の検出結果を図 26, 27 に示す。画像に対して相対的に小さな物体は追跡ができていたが、特に車など対象領域が広い物体は正確に囲めない場合が多い。領域の小さな物体でも細かい部分まで正確に囲めていない。これらの理由としては、次のようなものが考えられる。

a) 差分方法の問題

特に検出領域が大きい対象では選択したフレーム間で領域が重なることが多くなる。これにより、重なってしまった領域が安定せず、くっきりとした領域が抽出できていない(図 27)。また差分をとってインターセクションを求める手法では、白線などの背景成分が残ってしまい、これも周波数を乱す一因となっている。

b) 解像度、再現度の問題

低周波情報では大まかな領域抽出しかできず、ノイズが多い。さらに前処理の段階で画像をリサイズしていること(64 × 64 pixel)が問題となり、物体領域を正確に再現することができず、自転車の尖端など細かい部分が抽出できていない(図 26)。

本手法で作成したアルゴリズムでは精度と速度ともに優れたものには到達しなかった。だが低周波を使うことでエッジに作用されずに物体の中心を求めることが可能になっているところは他のアルゴリズムにない点である。

9.4 まとめ

高速フーリエコサイン変換を施し、その低周波成分を利用することで物体の大まかな位置を割り出すアルゴリズムを考案した。精度のよい認識には至らなかったが、物体追跡に周波数情報を利用するという新しい試みを行えた。低周波を用いれば物体のエッジに影響されにくく、物体中心がわかりやすいという利点がある。

10. むすび

今年度のアルゴリズムコンテストでは、パターン認識・メディア理解分野において活発に研究されている「移動物体の追跡」を題材として取り上げ、動き回る物体を撮影した動画像から移動物体を発見し、追跡するアルゴリズムを募集した。歴代のテーマを振り返ってみると、ステレオグラム、領域分割、物体抽出、物体識別、物体計数、画像モザイクング、ビデオのシーン分割、画像復元、教師なし分類、といった画像もしくは動画像処理の基本的なテーマは網羅されたように思われる。しかしながら、これまでのテーマは静止画像を対象としたものがほとんどであり、本年度のように動画像を対象としたテーマは少ない。近年、動画を対象とした研究も広く行われており、また、それらを処理するための計算機の性能も大幅に向上している。このようなことから、本年度のアルゴリズムコンテストでは動画像処理を題材とし、動画像処理の中でも多くの人に興味を持っていただけの「物体追跡」をテーマとして選んだ。その結果、応募総数は歴代 2 位の 73 件であり、学部 2 年生から博士課程の学生まで幅広い応募をいただいた。

1997 年に第 1 回アルゴリズムコンテストを実施してから、本年度で 14 回目を迎えた。年々アルゴリズムコンテストへの応

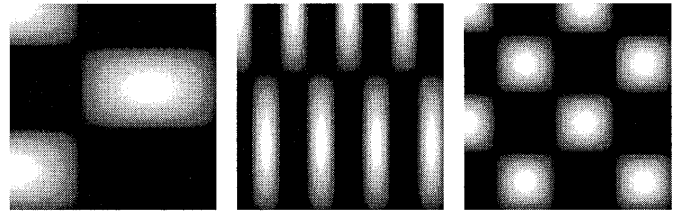


図 25 DCT 基底画像の例. 左から右図へ、基底 (3, 2), (2, 8), (4, 4) のみが 1 で他を 0 とした場合。

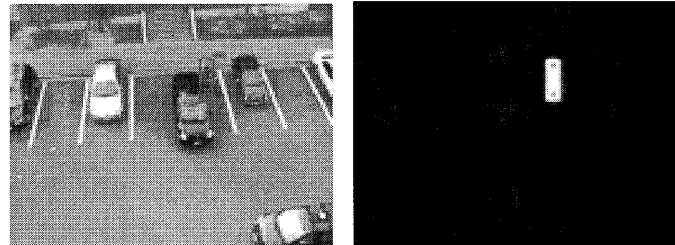


図 26 狭い物体の検出 (自転車)

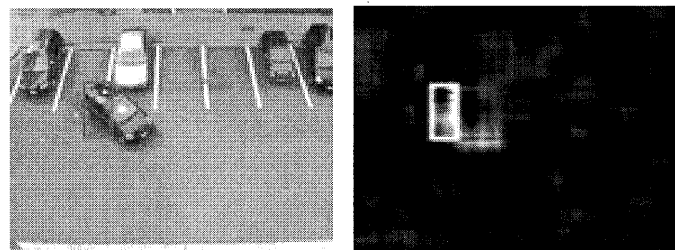


図 27 広い物体の検出 (車)

募者数も増加してきており、本コンテストも広く認知されるようになったと思われる。これもひとえに、本コンテストを授業の演習課題や研究室での研修に活用いただき、学生諸氏に広く本コンテストを紹介していただくなど、ご支援いただいている皆様のおかげである。今後もご支援いただき、本コンテストがパターン認識・メディア理解分野の活性化と若手育成に貢献できれば幸いである。また、若手研究者や学生に研究の楽しさを知ってもらう場として、今後も本コンテストが継続して発展していくことを期待する。

謝辞 本コンテストを実施するにあたり、多くの方々からご助言をいただくとともに、運営面でも多大なご協力をいただいた。ここに深く感謝する。特にコンテスト準備等に多大な協力をいただいた、愛知工業大学の北坂孝幸氏、名古屋大学村瀬研究室の道満恵介君、野田雅文君、内山寛之君、加藤祐二君に深く感謝する。

文 献

- [1] 2010 年度アルゴリズムコンテストホームページ,
<http://www.murase.m.is.nagoya-u.ac.jp/alcon2010/>
- [2] OpenCV,
<http://sourceforge.net/projects/opencvlibrary/>
- [3] MIST (Media Integration Standard Toolkit),
<http://mist.murase.m.is.nagoya-u.ac.jp/>
- [4] 新田直子, 岡田行史, 糟谷望, 宇津野雄亮, 藤垣真人, 徳本晋之介, 藤賢一朗, 小川文夫, 川崎亨, 丸山拓馬, 大倉史生, “2009 年 PRMU アルゴリズムコンテスト「クローンを探せ！」実施報告と受賞アルゴリズム紹介,” 電子情報通信学会技術報告, PRMU2009-155, pp.95-105, December 2009.