

最近傍識別器を用いた背景差分と色検出の統合

加藤 丈和[†] 柴田 智行[†] 和田 俊和[†]

[†]和歌山大学 システム工学部 情報通信システム学科
〒 640-8510 和歌山県 和歌山市 栄谷 930
E-mail : t.kato@ieee.org

あらまし

本稿では、最近傍識別器のコンピュータビジョンへの応用として、背景差分と色検出を統合したターゲット検出システムについて述べる。我々は現在までに、画像の各画素を YUV の 3 次元ベクトルとみなし、最近傍識別器によって識別を行うことによって、色ターゲットの検出を行う方法を提案した。本研究では、その拡張として、あらかじめ用意した背景画像の画素の色 3 次元 (YUV) と、入力画像の対応する画素の色 3 次元 (YUV) を合せた 6 次元のベクトルを識別することにより、背景差分によって得られる効果と、色検出によって得られる効果を両方兼ね備えたターゲット検出システムを構築する。実験では、半透明物体と背景が変動する場合に対するターゲット検出実験を示し、その有効性を実証する。

キーワード 最近傍識別器, 色検出, 背景差分

Integration between Background Subtraction and Color Detection based on Nearest Neighbor Classifier

Takekazu Kato[†], Tomoyuki Shibata[†], Toshikazu Wada[†]

[†]Department of Computer and Communication Science, Wakayama University,
930 Sakaedani, Wakayama, 640-8510, Japan

E-mail : t.kato@ieee.org

Abstract This paper describes a target-detection system integrating a color-detection and a background subtraction based on nearest neighbor classifier. We proposed a color-target detection system that regarded each pixel of images as 3-d vector (YUV) and classified it by nearest neighbor classifier. In this paper, we expand the color-target detection into target-detection integrating a color-detection and a background subtraction. It regards each pixel as 6-d vector that consists of 3-d vector in a pixel in input images and 3-d vector in a background image. Experimental results demonstrate the effectiveness of our method against semitransparent target and variable background.

Keywords nearest neighbor classifier, color target detection, background subtraction

1 まえがき

多くのコンピュータビジョンのアルゴリズムでは、様々な物理モデルや統計モデルを用いてシーンの解析を行う方法がとられてきた。このようなアプローチの欠点としては、モデルに関する事前知識を必要とする点、事前に与えたモデルで表現できない複雑なシーンや、例外的な現象に対処することが困難な点などが挙げられる。このような問題に対し、我々は、物理モデルや統計モデルを用いず、事

例によって事象を説明する Case-Based Reasoning (CBR) [1] のアプローチに基づくコンピュータビジョンアルゴリズムの研究を行っている。

CBR の中核となりうる基本的な技術として、最近傍識別器 [3] があげられる。最近傍識別器は、トレーニングパターンをプロトタイプとして記憶し、入力パターンと最も近いプロトタイプが属するクラスによって対象を識別する手法であり、1) 十分なトレーニングパターンが与えられれば、ベイズ

誤り確率の二倍以下の誤り確率を達成できる, 2) 分布モデルなどの事前知識を必要としない, 3). そのままで多クラスの識別が可能である, などの優れた性質を持っている. 一方で, 識別速度が遅い, メモリを大量に消費するなどの欠点も持っているため, いままで最近傍識別器がコンピュータビジョンの実用的なタスクに応用された例は少ない.

我々は, このような問題点を解決するために, 最近傍識別器の高効率化, 高速化 [4, 7, 8, 9] を行うとともに, 最近傍識別器のコンピュータビジョンへの応用 [5, 6] についての研究を行ってきた. 文献 [6] では, 画像の各画素の色を YUV 色空間の 3 次元ベクトルとして表現し, 最近傍識別によって識別することで, 精度の良い色ターゲット検出を実現する手法を提案した. また, Look Up Table (LUT) を利用した高速化によって, ビデオレートでの検出を実現した. また, 文献 [5] では, 画像中の領域を表わす二値画像を最近傍識別によって識別することで, 領域の認識を行った. この手法では, 領域画像の 4 分木表現を用いることによって, 高次元の二値画像を高速に識別した.

本稿では, 画素の識別に基づくターゲット検出をより柔軟かつ安定に行うために, 色検出と背景差分を統合したターゲット検出手法を提案する. 画像中の各画素を背景画像の色 (YUV) と入力画像の色 (YUV) を合せた 6 次元ベクトルとして識別することで, 色検出と背景差分の特色を持ち合せたターゲット検出手法を実現する. また, ハッシュ表による識別結果のキャッシュを用いた最近傍識別の高速化手法を示す.

2 最近傍識別によるターゲット検出

画像中からターゲットを検出する問題は, コンピュータビジョンの中でも重要な課題の一つとして挙げられる. その中でも画像中の特定の色を検出する色検出と, あらかじめ用意した背景画像から変化した領域を検出する背景差分は, ターゲット検出の基本的な技術として様々な場面で用いられている.

これら技術はどちらも, 画像中の各画素を, その画素値に基づいて, ターゲットと非ターゲットに識別する問題と見なすことができる. このような問題に対して, これまでに様々な研究がなされてきたが, 従来の研究の多くは, ターゲットの画素値のパターンを物理的, 統計的なモデルで表現し, モデルとの一致度を「ターゲットらしさ」を表す確率や尤度, 類似度などによって判断している. しかし, このような手法では, 対象のモデルを事前知識として与える必要があり, 対象とするシーンが複雑に変化する場合など, 事前に与えたモデルでは表現するのが困難であるような場合や, 例外的な現象に対して精度がよくないという問題点があった (図 1(a)).

このような問題点を解決するためには, 図 1(b) に示すように, ターゲットをモデルによって表現することなく, パターン空間中に直接識別面を構成すれば良い. 我々はこのような考えに基づき, 最近傍識別器 [3] を用いた画素の識別に基づく色ターゲット検出手法 [6] を提案した. 最近傍識別器は, 与えられたトレーニングパターンに対して, 最大マージン基準を与える区分的識別面を持ち, 図 1(b) に示すような識別面を容易に得ることが可能である.

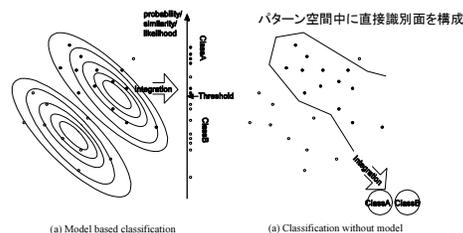


図 1: モデルに基づく識別とモデルを使わない識別

2.1 色検出

文献 [6] では, 画像の各画素を YUV の 3 次元ベクトルと見なし, 3 次元パターン空間内での識別を行うことで色ターゲット検出を実現する手法を提案した. この手法では, トレーニングパターンをユーザがインタラクティブに指示することによって, 照明変動によって画素値が変化する場合や, カラー LED の検出など, 従来安定な検出が困難であったシーンに対して, 安定な検出を実現した.

2.2 背景差分

背景差分の問題は, あらかじめ与えた背景画像と入力画像との画素値の差を用いて背景領域かターゲット領域かを決定する問題と捉えることができ, 色ターゲット検出の問題と同様に考えることができる. 濃淡画像に対する背景差分では, 単純に濃度の差分値に対する閾値の決定問題と見なすことができるが, カラー画像に対する背景差分では, 色の違いをどのように評価するかということが重要な問題となる.

従来では, 色を表わすベクトル間にユークリッド距離などの距離を定義し, その距離に対して閾値を設定する方法や, 色ベクトルの各要素に閾値を設定する方法, 色ベクトル空間での差分値の分布に対して統計的なモデルをあてはめ, モデルとの類似度を評価する方法などが用いられてきた. このような手法では, どのような色空間を用いるか, また, 色ベクトルの各要素をどのように評価するかが重要な問題となっていた.

このような問題に対して我々のアプローチでは, 単純に色ベクトル間の差分ベクトルの識別問題と考え, 差分ベクトルを直接識別することで, 色空間や色ベクトルの評価法に影響されにくい安定な背景差分を実現することができる.

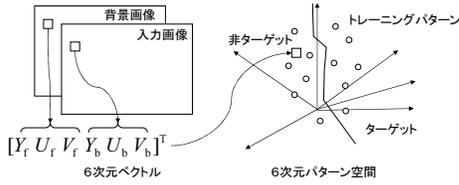


図 2: 6次元ベクトルと6次元パターン空間

例えば、画素の色をYUV色空間で表現する場合、入力画像のある画素の色を Y_f, U_f, V_f 、背景画像の対応する画素の色を Y_b, U_b, V_b としたとき、色ベクトル間の差 d を次式のように与える。

$$d = [|Y_f - Y_b| \quad |U_f - U_b| \quad |V_f - V_b|]^T \quad (1)$$

このようにして得られた差分ベクトル d を、3次元パターン空間で最近傍識別によって識別することで、背景領域かターゲット領域かを決定することができる。

2.3 色検出と背景差分の統合

色検出では、背景にターゲットと似た色が存在する場合に検出が不安定になるという問題がある。また、背景差分では、半透明物体のようにターゲット色が背景に影響されて変化する場合や、物体の影のように逆にターゲットによって背景が変化する場合、また、背景が変動する場合などに安定な検出を行うことができないという問題がある。このように、これらの手法にはそれぞれ得手、不得手が存在する。そこで本研究では、これらの手法を統合したターゲット検出手法を提案する。

従来、このように異なる手法の統合を実現するためには、それぞれの手法の評価値をどのように統合するかが大きな問題点となっていたが、我々のアプローチでは、このような評価値の統合の問題を考える必要がなく、それぞれの入力パターンを合せたパターンに対して直接識別するだけで、それぞれの手法と統合したのと同じ結果を得ることが可能である。ここでは、色検出で用いる入力画像の各画素の色ベクトルと、背景差分で用いる背景画像の対応する画素の色ベクトル¹を合せたベクトルを構成し、最近傍識別器によって識別することで、色検出と背景差分との統合を実現する。

例えば、画素の色をYUV色空間で表現する場合、入力画像のある画素の色を Y_f, U_f, V_f 、背景画像の対応する画素の色を Y_b, U_b, V_b としたとき、次式のような6次元ベクトル c を構成する。

$$c = [Y_f \quad U_f \quad V_f \quad Y_b \quad U_b \quad V_b]^T \quad (2)$$

このようにして得られた c を6次元パターン空間で最近傍識別によって識別することで、色検出と背景差分を統合したターゲット検出を実現することができる。

¹実際には、背景差分では背景画像と入力画像との差分ベクトルを用いるが、入力画像の色ベクトルと合せた場合、本質的に同じ情報になるため背景画像の色ベクトルをそのまま用いた。

3 ターゲット検出のための最近傍識別器の高速化

画像の各画素の識別によるターゲット検出では、画像中の全ての画素を識別する必要があるため、非常に高速な識別が必要となる。例えば 640×480 のサイズの画像からターゲット検出を行うためには、約30万回の識別を行う必要があり、これをビデオレート(30fps)で動作させるためには、1個のパターンに対して約 $0.1\mu s$ 以内に識別が終了しなければならない。また、結果を見ながらトレーニングデータを追加できるインタラクティブなシステムの構築のためには、学習時間も高速である必要がある。

我々は、既存の最近傍識別アルゴリズムに比べて最大で約400倍の高速化が実現可能なアルゴリズム、K-D Decision tree(KDDT) [9, 4]を提案しているが、この手法でも1個のパターンに対して、3次元で $0.9\mu s$ 、6次元で $14.1\mu s$ の識別時間が必要であり十分な識別速度とは言えない。また、KDDTは、次元数やサンプル数が増加すると学習に時間がかかり、6次元では数十秒から数分の時間がかかるという問題点もある。

また、色ターゲット検出 [6]では、Look Up Table (LUT)を用いた高速化により、ビデオレートの色ターゲット検出を実現している。しかし、この手法では、色空間全体のクラス情報をLUT上に記録するため、6次元の識別に用いることはできない。また、LUTの更新アルゴリズムより、パターン間の距離尺度が市街地距離に限定され、ユークリッド距離等の他の距離定義に適用することができないという問題点もある。

3.1 キャッシュを用いた識別の高速化

本研究では、実際の画像には、画素値の偏りがあり、同じ色が繰り返し出現することが多いという点に着目し、キャッシュを用いた高速化を行う。つまり、一度識別を行った識別結果をメモリ上に記憶しておき、次に同じ色が出現した場合には、記憶した結果を用いて識別を行う。

キャッシュアルゴリズムには、検索速度と、メモリ使用量をコントロールしやすい点から入力パターンをキーとしたハッシュ表を用いた。ここで、入力パターンに対応するハッシュ値は、似た色が多く出現するという画像の特性を考慮し、次式のように計算した。

$$\begin{aligned} \text{hash}(Y_f, U_f, V_f, Y_b, U_b, V_b) = & (Y_f \bmod \hat{Y}_f) * \hat{U}_f * \hat{V}_f * \hat{Y}_b * \hat{U}_b * \hat{V}_b \\ & + (U_f \bmod \hat{U}_f) * \hat{V}_f * \hat{Y}_b * \hat{U}_b * \hat{V}_b \\ & + (V_f \bmod \hat{V}_f) * \hat{Y}_b * \hat{U}_b * \hat{V}_b \\ & + (Y_b \bmod \hat{Y}_b) * \hat{U}_b * \hat{V}_b \\ & + (U_b \bmod \hat{U}_b) * \hat{V}_b \\ & + (V_b \bmod \hat{V}_b) \end{aligned} \quad (3)$$

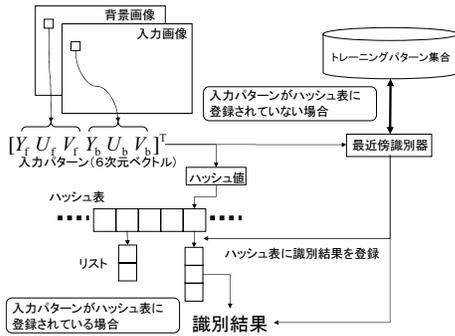


図 3: ハッシュ表による識別結果のキャッシュ

ただし、 Y_f, U_f, V_f は入力画像の画素の色、 Y_b, U_b, V_b は背景画像の画素の色である。また、 $\hat{Y}_f, \hat{U}_f, \hat{V}_f, \hat{Y}_b, \hat{U}_b, \hat{V}_b$ は定数であり、 mod は剰余演算を表す。

式 (3) によって、似た色に対して異なるハッシュ値が得られるようにすることで、似た色が多く出現する場合に高速な検索が可能である。また、この計算は、各画素値の法が 2 のべき乗のときには、各画素値に対する論理積、論理和、ビットシフトのビット演算のみで計算することが可能であり、計算機で高速に計算することができる。また、同じハッシュ値を持つ結果に対しては、ハッシュ値の再計算は行わず、ハッシュ表の各エントリをリストで管理する。

図 3 にアルゴリズムの概要を示す。あるパターンについてハッシュ表を検索したとき、ハッシュ表に同じパターンの結果が存在しなければ、最近傍識別によってターゲットのクラスを決定し、その結果をハッシュ表のエントリのリストの先頭に挿入する。また、そのパターンの識別結果がハッシュ表に存在する場合は、その結果を用いる。このとき、検索したパターンをリストの先頭に移動することで、同じパターンを連続して検索したときに高速に検索可能にする。

4 実験結果

4.1 ターゲット検出システム

以下の 3 種類の検出アルゴリズムを用いて文献 [6] と同様のターゲット検出システムを構築した。

1. 入力画像の YUV を 3 次元ベクトルとして識別 (色検出)
2. 入力画像と背景画像の YUV それぞれの差分値を 3 次元ベクトルとして識別 (背景差分)
3. 入力画像の YUV 背景画像の YUV を合せた 6 次元ベクトルとして識別 (統合手法)

本システムではユーザが画像上の領域を指定して、ターゲット 1 からターゲット 255、及び非ターゲットのクラスを対応づけることで、インタラクティブに学習を行うことができる。

最近傍識別アルゴリズムには Arya らが提案した Approximate Nearest Neighbor (ANN) [2] を用いた。ANN では許容誤差を設定できるが、ここでは許容誤差は 0 (誤差なし) とした。

また、手法 1., 2. では Y, U, V それぞれを 128 段階に量子化した YUV 色空間を用い、手法 3. では 64 段階に量子化した YUV 色空間を用いた。手法 3. の場合に 64 段階の量子化を行った理由は、最近傍識別ではパターン空間が広くなると、より多くのトレーニングパターンが必要になり、インタラクティブにトレーニングパターンを指示する場合に時間がかかるためである。なお、式 (3) のハッシュ値の計算に用いる定数 $\hat{Y}_f, \hat{U}_f, \hat{V}_f, \hat{Y}_b, \hat{U}_b, \hat{V}_b$ は全て 32 とした。

使用機材は、計算機: Intel Xeon 2.4GHz Dual CPU, カメラ: SONY DFW VL500 を用い、入力画像は 640x480 の YUV カラー画像を用いた。

4.2 半透明物体の検出

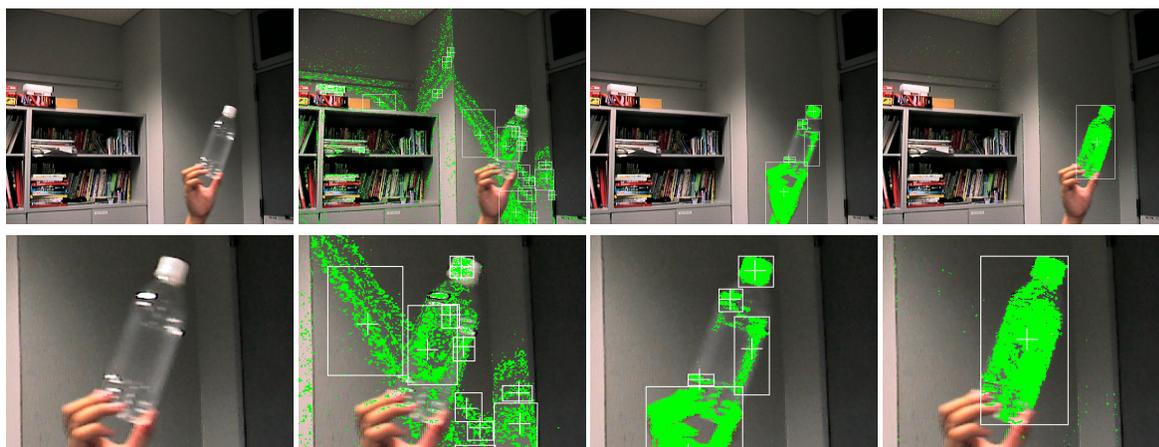
まず、半透明物体の検出結果を図 4 に示す。この実験では、半透明物体であるペットボトルをターゲットとし、これを左から右に移動させながら撮影したシーケンスを用いた。全てのフレームの画像を使って、できるだけペットボトルだけが検出されるようにトレーニングを行った。

色検出 (a) では、ターゲットの部分で背景色が透けて見えているため、背景の色と似た色となるため、背景との切り分けができていないことが分る。また、キャップの白い部分についても、背景に非常に明い部分があったため、切り分けが困難となっている。背景差分 (b) では、キャップや手などの不透明な部分に関しては、検出ができていたが、半透明な部分では、背景色との違いが小さいため、検出が困難であることがわかる。

これに対して統合手法 (c) では、ほぼ完全に背景色が透けている部分で若干の抜けはあるものの、安定に検出できていることがわかる。また、背景差分では、手の部分もターゲットとして検出しているが、統合手法では手の部分は非ターゲットとして切り分けができていたことがわかる。これは、統合手法は色検出の機能も兼ね備えているため、ターゲットとして検出したい部分だけをターゲットとしてトレーニングし、それ以外を非ターゲットとしてトレーニングすることで、このような画像の変化があるが、非ターゲットであるような領域を切り分けることができる。

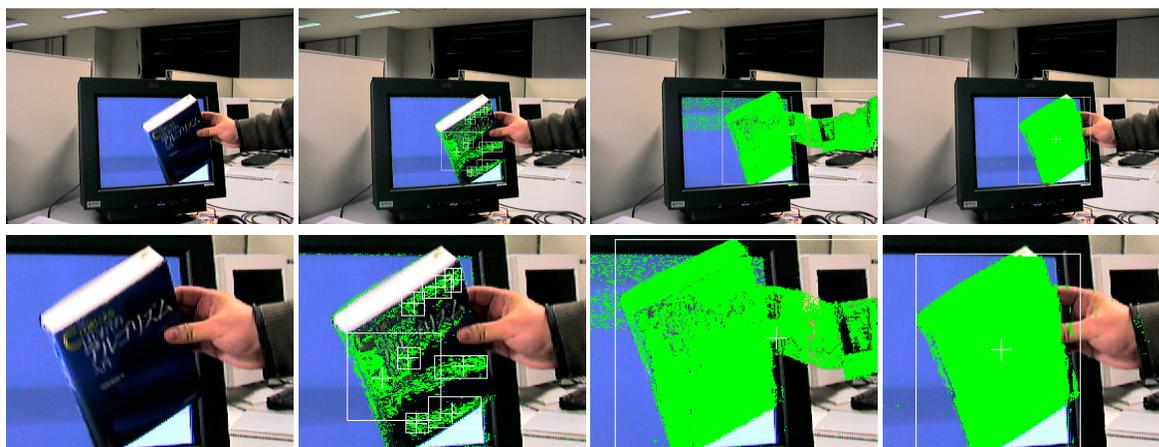
4.3 カラーディスプレイを背景とした検出

次に CRT の前にターゲットを置いて検出した場合の結果を図 5 に示す。この場合はターゲットは青色の本とし、これを薄い青色を表示したディスプレイの前で本の傾きを変えながら撮影したシーケンスを用いて、半透明物体の場合と同様に全て



(a) 原画像 (b) 手法 1: 色検出 (c) 手法 2: 背景差分 (d) 手法 3: 統合手法

図 4: 半透明物体の検出結果 (上段: 全画面表示, 下段: 拡大表示)



(a) 原画像 (b) 手法 1: 色検出 (c) 手法 2: 背景差分 (d) 手法 3: 統合手法

図 5: 背景にディスプレイがあるシーンの検出結果 (上段: 全画面表示, 下段: 拡大表示)

のフレームでできるだけ正しく検出されるようにトレーニングを行ったときの結果である。

色検出 (a) の結果では、影になって暗い部分で検出が失敗していることがわかる。これは、ディスプレイの枠や後の電源を来ってあるディスプレイの色が暗いため、ターゲット中の暗い部分で検出に失敗したためである。また、背景差分 (b) の結果では、ターゲット領域はほぼ検出できているが、ディスプレイの部分に誤検出が存在していることがわかる。これは、ディスプレイの同期周波数とカメラのシャッタースピードとの関係によって、フリッカーが生じディスプレイ上に帯上に暗い部分ができたためである。これらの結果に対し統合手法 (c) では、ほぼ完全にターゲットの部分だけを検出できていることがわかる。

4.4 検出速度の評価

次に、検出速度についての評価を行った結果を図 6 に示す。これは、図 5 の結果を含むシーケンスの検出を行ったときの結果であり、1 フレーム毎

のキャッシュのヒット率を (a) に、1 フレームあたりの検出時間を (b) に、キャッシュサイズを (c) に示す。

キャッシュのヒット率 (a) の結果より、フレームの始めの部分では、キャッシュのヒット率が低く、その後ヒット率が上昇し、100 フレームほどでほぼ 1 に近づいていることがわかる。また 100 フレームから 200 フレームのあたりでヒット率が低下しているが、これはこのフレームで入力画像に大きな変化があったためである。また、シーケンス全体を通してヒット率が最悪でも 0.99 程度であることから、同じ画素値を持つ画素が多く出現していることがわかり、キャッシュが非常に有効に働いていることが分る。

また、検出時間 (b) の結果より、ヒット率が高い部分では高速であり、ヒット率が低くなると検出に時間がかかっていることがわかる。この結果より、キャッシュが有効に働いている部分では高速な検出が実現できていると言える。また、6 次元ベクトルを用いる統合手法では、特にその差が顕著で

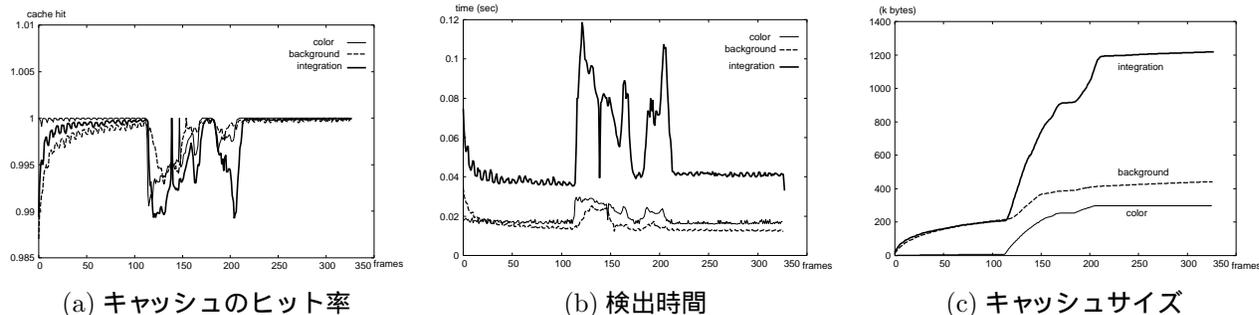


図 6: 検出時間とキャッシュの状況

ありキャッシュによる高速化の効果が高いことがわかる。

次に (c) のキャッシュサイズについて、これはハッシュ表に登録されているエントリの数と、1 エントリに必要なバイト数の積を求めたものである。この結果より、6 次元ベクトルを用いる場合でも、最終的に 1.2M 程度のメモリ使用量であり、メモリの使用効率も良好であることがわかる。

5 まとめ

本研究では、最近傍識別器を用いたターゲット検出手法として、[6] で提案した色検出に加え、背景差分に適用する方法、さらに色検出と背景差分を統合したターゲット検出手法を提案した。また、6 次元ベクトルを用いた最近傍識別を高速に実行するために、ハッシュ表によるキャッシュを用いた高速化手法を提案した。

実験では、色検出と背景差分を統合することにより、半透明物体や背景に変動がある場合など、従来検出が困難であったシーンでのターゲット検出を実現できていることを示した。また、検出速度に関しては 6 次元ベクトルを用いる場合でも 1 フレームあたり平均で約 0.05 msec、最悪でも 0.12 msec の検出が可能であり、ビデオレートには至らないものの、実時間アプリケーションに適用するのに十分な検出速度が得られていることを確認した。

本手法の問題点としては、キャッシュのヒット率がほぼ 100% に近い場合でも、文献 [6] の LUT を用いた手法に比べると検出に時間がかかっていることが挙げられる。これは、キャッシュのメモリ使用量は少ないものの、識別結果の登録時に動的にエントリ用のメモリの確保を行っているために、メモリの使用領域が広範囲に分散し、CPU のメモリキャッシュのヒット率が下がっているためであると予想される。このような問題に対しては、キャッシュエントリに用いるメモリ領域を予め連続領域として確保しておく、その領域からエントリ用の領域を確保することによって解決できると考えられる。

また、6 次元ベクトルを用いた場合に、キャッシュを外れたときの速度低下が顕著である。このような問題に対しては、最近傍識別自体の高速化が必要である。我々が提案した KDDT アルゴリズムは、

識別は高速であるが、学習に 6 次元では数十秒から数分かかり、また新たなトレーニングパターンが与えられると、その都度再学習が必要であることから、本稿で述べるようなインタラクティブなシステムには適していない。今後の課題としては、KDDT の学習時間の高速化、トレーニングの追加に対するインクリメンタルな学習の実現などの改善により、KDDT を本手法に適用することが挙げられる。

参考文献

- [1] D. W. Aha. Case based learning algorithms. In *Case based Reasoning Workshop*, pp. 147–158, 1991.
- [2] S. Arya, D. M. Mount, N. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. In *5th ACM-SIAM Symposium Discrete Algorithms*, pp. 573–582, 1994.
- [3] B.K.B. T.M. Cover and P.E. Hart. Nearest neighbor classification. *IEEE Transactions on Information Theory*, Vol. IT-13, No. 1, pp. 21–27, 1967.
- [4] Tomoyuki Shibata, Takekazu Kato, and Toshikazu Wada. K-d decision tree: An accelerated and memory efficient nearest neighbor classifier. In *ICDM'03*, pp. 641–644, Nov 2003.
- [5] 武本浩二, 加藤丈和, 和田俊和. 画像の 4 分木表現に対する最近傍識別. 信学技報 PRMU, Vol. 103, No. 390, pp. 1–6, Oct 2003.
- [6] 和田俊和. 最近傍識別器を用いた色ターゲット検出—「らしさ」に基づかない識別とコンピュータビジョンへの応用—. 情処研報 CVIM134-3, No. 134, pp. 17–24, 2002.
- [7] 和田俊和, 加藤丈和. 近接性グラフに基づく効率的 condensing の理論. 信学技報 PRMU, Vol. 103, No. 96, pp. 13–18, May 2003.
- [8] 加藤丈和, 和田俊和. 近接性グラフに基づく効率的 condensing のアルゴリズムと評価. 信学技報 PRMU, Vol. 103, No. 96, pp. 19–24, May 2003.
- [9] 柴田智行, 和田俊和, 加藤丈和. K-d decision tree: 最近傍識別器の高速化. 信学技報 PRMU, Vol. 103, No. 295, pp. 85–90, Sep 2003.