平成22年度
中部大学大学院工学研究科情報工学専攻

博士学位論文

画像局所特徴量を用いたオンライン学習による物体追跡と行動認識に関する研究

山下隆義

論　文　要　旨

本論文では，カメラ映像から人の行動を理解するために必要となる物体検出，物体追跡，行動認識の各要素技術について，新たな枠組みを提案する．

物体検出は，顔を中心に実用化が進んでいるが，監視カメラで撮影されるような検出対象が小さい場合や混雑して重なりが生じる環境において，検出対象を高精度に検出できない．そのため，人体のパーツ検出器を組み合せて統合的に検出精度を向上させる方法がある．この方法の場合，物体検出をより高精度に実現するためには，パーツ間の統合精度が重要となる．そこで，本論文では，近赤外光により輝度画像と距離情報を取得できる TOF カメラを用いたパーツ検出器による人物検出手法を提案する．提案手法では，TOF カメラにより得られた距離情報を用いて，パーツ検出結果を統合することで，人物検出の高精度化を実現することができる．

物体追跡は，追跡対象の変化とともに周辺の変化に対しても頑健であることが求められる．これらの変化が生じる中で途切れなく追跡する方法として，物体検出手法を応用したオンライン学習による物体追跡手法がある．オンライン学習は，ある時刻の追跡対象に合わせて検出器を更新するため，追跡対象の大きさや見えの変化，照明変化に対して頑健である．一方で，時々刻々と弱識別器を更新するため，遮蔽が生じた場合に遮蔽物へ乗り移りが生じる．そこで，本論文では，オンライン学習において，物体らしさを表す確率密度関数を導入した Online Real Boosting を提案する．提案手法により，異なる物体への乗り移りを軽減させることが可能となる．

また，人物のように姿勢変化が生じるような物体を追跡するために，追跡対象に適応して形状変化が可能な特徴量を導入したオンライン学習についても提案する．従来のオンライン学習はあらかじめ学習した弱識別器から適した弱識別器を選択する．そのため，学習時に考慮していない姿勢が生じた場合，最適な弱識別器が選択されない場合がある．我々は，確率密度関数が類似する特徴量をグループとして 1 つにした弱識別器構造を導入する．これにより，姿勢変化が生じた場合においてもロバストに追跡することができる．

行動認識は，時系列での人の動線だけでなく，姿勢変化を捉える必要がある．従来は，あらかじめ検出された人の領域を正規化し，多クラス認識手法を用いる方法が一般的である．この方法は，物体検出と行動認識を別々の枠組みで行っているため，認識精度は，検出された人の領域に依存する．そのため，検出領域が正しく人の位置を捉えていない場合，正規化が正しく行われず認識精度が低下する．そこで本論文では，マルチクラスの識別器である Randomized Trees と Joint Boosting を組み合せたマルチクラスの学習手法について提案する．提案手法では，決定木のノード生成時に複数のクラスの識別に最適な特徴量を選択することができる．提案手法により，木構造の上位階層には物体と背景を識別するノード，下位の階層には姿勢を識別するノードを自動的に配置できるため，人体の検出と同時にその行動の認識が可能である．

目 次

図 目 次

表 目 次

第1章

序論

本章では，本論文中の研究における背景および目的，構成について述べる．

1.1 本研究の背景

デジタルカメラやネットワークカメラを中心としたデジタル画像機器は，ハードディスクの大容量化やインターネットの普及，CPU の高性能化により映像の撮影および記録が容易となり，セキュリティ分野，マーケティング分野，映像撮影分野など多岐にわたり活用されている．このような分野において，これまでは大量の映像を蓄積することが一般的であり，大量に蓄積された映像から目的となる情報を取得するのは，作業者の目視作業に頼らざるを得なかった．また，その作業は，非常に高負荷であり作業者のストレスが大きな問題となっている．一方，蓄積された映像はデジタル情報として撮影するため，画像の補正や解析が容易である．そのため，映像を解析し，目的となる情報を容易に習得するような活用方法が求められている．画像の補正や解析を実現するためには，対象とする物体の位置や状態を正しく捉える必要がある．特に人を対象とする場合，人の位置や動き，行動などを理解することが重要となる．人の行動理解は，以下の 4 つの分野において新たな価値を創造することが可能となる．

[セキュリティ分野]

米国での同時多発テロ以降，急速に普及しているビデオ監視システムにおいて，より複雑な行動認識の実現が望まれている．従来のビデオ監視システムは，背景差分を主とした背景モデル技術をもとに移動物体を検出し，その大きさや動き，形の変化をもとに対象物体であるかどうかを判別している [1][2]．また，時系列における移動領域の形状変化から行動を認識しているため，必ずしも対象とする行動を捉えているか分からない．そのため，対象とする物体を検出し，その物体の動きからより複雑な行動を認識することが期待されている．

[マーケティング分野]

店舗において購買を促進させるための新たなマーケティング手段が求められている．従来は，利用顧客が購入した商品をもとに商品の人気や価値を判断している．すなわち，利用顧客の購入後の情報しか得ることができず，購入に至る情報を得ることができていない．そのため，利用顧客が店舗内においてどのようなルートを通ったのか，商品購入時にどのような商品と比較して選択したのか，購入を迷ったのかどうかなど，より利用顧客の行動に踏み込んだ情報を得ることができれば，商品の最適配置や商品の推奨方法を最適化することが期待されている．

[映像撮影分野]

デジタルカメラなどの撮影機器において，撮影者の力量に問わず誰でもきれいな写真を撮影することが求められている．これまでは，被写体の顔を見つけたり，あらかじめ指定した物体を追跡してオートフォーカスやオートアイリスを実現しており，自動化が進んでいる．しかし，動き回る被写体を撮影するためには，撮影者がカメラを被写体に合わせる必要があり，自動化できていない．そのため，撮影者は被写体を映像におさめるために，カメラのファインダーを注視しておかなければならなず，撮影の負担が大きい．人物を追跡することで，撮影者がファインダーを注視せず，撮影の負荷

を軽減することができる．また，人物の追跡を通して，追跡人物がフレーム外へ出るかどうかを判断することができれば，自動的にフレーミングしたり，撮影者へフレームアウトすることを警告することで，撮影の失敗を防ぐことができる．

[福祉分野]
福祉分野では，今後の高齢化社会での高齢者を支援するためのロボットやスマートホームによる生活支援が期待されている．これまでは，監視カメラ映像から顔認証を利用することで、徘徊者かどうかを認識し，徘徊者が施設外へ出ないように支援するシステム等が実現されている [3]．また，搭載されたカメラで顔を認識しコミュニケーションが可能なロボットが実現されている [4]．これらの支援システムは，顔を認識することで誰であるかを認識できているが，要支援者の行動認識まで至っていない．介護者の負担を減らしたり，より円滑なコミュニケーションロボットを実現するためには，対象の行動を理解することが重要である．

1.2 本研究の目的

人の行動を理解することは，前節で述べたような様々な分野において，新たな価値を創造することができる．人の行動理解には，図 1.1 に示す物体検出，物体追跡，行動理解の３つの要素技術が重要となる．

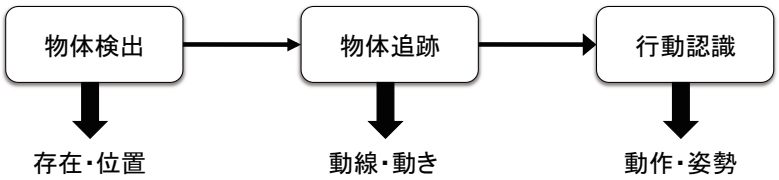


図 1.1: 人の行動理解における要素技術と行動の関係

物体検出技術は，人がいるかいないかを判定することで存在が分かり，その位置も同定することができる．物体追跡技術は，同一人物の時系列における位置を特定し，その軌跡からから動線を把握できる．行動認識技術は，人のより詳細な姿勢やその動きから動作を把握することができる．各要素技術は，物体検出，物体追跡，行動理解の順に実用化も進んでいるが，利用できる環境や認識可能な行動が限定されているのが現状である．

一方，各要素技術はそれぞれ関連しており，物体追跡技術は，物体検出結果をもとに追跡する物体を初期化する．また，行動認識は時系列での物体追跡結果やフレーム間での変化をもとに認識対象を決定する．このように，人の行動理解は物体検出から物体追跡，行動理解の一連の流れにより実現されており，前過程の処理結果が最終的な性能に大きく影響している．そのため，高精度に人の行動を理解するためには，各要素技術の性能を向上させることが重要であり，課題となっている．

そこで，本研究では，各要素技術における課題を解決するための新たな枠組みを提案することで，各要素技術の性能向上を図り，高精度に人の行動を理解することを目的とする．以下に各要素技術における提案内容を述べる．

-物体検出

物体検出は実用化が進んでいるが，監視カメラで撮影されるような検出対象が小さい場合や混雑して重なりが生じる環境において，検出対象を高精度に検出できないという問題がある．そのため，人体のパーツ検出器を組み合せて統合的に検出精度を向上させる方法が提案されている [5][6][7]．この方法では，物体検出をより高精度に実現するためには，パーツ検出結果の統合精度が重要となる．本研究では，近赤外光により輝度画像と距離情報を取得できる TOF カメラを用いたパーツ検出器による人物検出手法を提案する．提案手法では，TOF カメラにより得られた距離情報を用いて，パーツ検出結果を統合することで，人物検出の高精度化を実現することができる．

-物体追跡

物体追跡は，追跡対象の変化とともに周辺の変化に対しても頑健であることが求められる．これらの変化が生じる中で途切れなく追跡する方法として，物体検出手法で一般的なブースティングを用いたオンライン学習による物体追跡手法が提案されている [8][9]．オンライン学習は，ある時刻の追跡対象に合わせて検出器を更新するため，追跡対象の大きさや見えの変化，照明変化に対して頑健である．一方で，時々刻々と弱識別器を更新するため，遮蔽が生じた場合に遮蔽物へ乗り移りが生じる．そこで，本研究では，オンライン学習において，物体らしさを表す確率密度関数を導入した Online Real Boosting を提案する．確率密度関数により物体らしさを判断することで，異なる物体への乗り移りを軽減させることが可能となる．

また，人物のように姿勢変化が生じるような物体を追跡するために，追跡対象に適応して形状変化が可能な特徴量を導入したオンライン学習についても提案する．従来のオンライン学習はあらかじめ学習した弱識別器から適した弱識別器を選択する．そのため，学習時に考慮していない姿勢が生じた場合，最適な弱識別器が選択されない場合がある．本研究は，確率密度関数が類似する特徴量をグループとして 1 つにした弱識別器構造を導入する．これにより，姿勢変化が生じた場合においてもロバストに追跡することができる．

-行動認識

行動理解は，対象物体の動き量や動き方向など動線をベースとする方法と検出対象の形状から姿勢変化をベースとする方法がある [10][11]．動線をベースとした方法は，物体追跡手法から得られた情報をもとに推定することが可能である．本研究では，姿勢変化をベースとした方法による行動理解技術について検討する．従来の行動認識は，背景差分や人検出などを用いて、人の領域を抽出した後，人の領域の動き情報を捉えて行動を認識する．そのため，人の領域を正しく検出できていないと，行動認識の精度が低下する．そこで，本研究では人体を検出すると同時に姿勢の推定および行動の認識を 1 つのフレームワークで実現可能な Boosted Randomized Trees を提案する．提案手

法は，Randomized Trees を構成する決定木のノード生成時に，Joint Boosting を用いて特徴選択を行っている．これにより，各ノードに適した認識に有効な特徴を選択することが可能となり，人体検出率および行動認識率を向上させることができる．

1.3 本論文の構成

本論文の内容は，図 1.2 に示すように，7 つの章から構成以下のような構成されている．1 章は序論であり，本研究の背景，目的を述べる．本研究は，人を理解する上で重要となる物体検出、物体追跡、行動認識の 3 つの要素技術について，それぞれ新たな枠組みを提案する．それにより，セキュリティ分野やマーケティング分野，最適撮影分野など多岐にわたり新たな価値を創造することができる．

2 章では，物体検出について述べる．物体検出は，統計的学習手法を用いて検出対象に適した特徴量を選択し，検出器を学習する方法が一般的である．検出器の学習に用いる特徴量は，輝度やエッジに着目するものを中心として多数提案されている．そこで，物体検出に用いられる画像局所特徴量について調査してまとめる．また，予備実験として，顔だけでなく，人や車を検出対象とした場合に適した特徴量について比較実験を行う．

3 章では，物体検出を TOF カメラに応用した手法を提案する．提案手法は，人が小さい場合や混雑したシーンにおいて，高精度に検出するためにパーツベースの検出器を用いる．また，パーツ検出器ごとの検出結果を統合する際に，TOF カメラで得られた距離画像を用いることで，統合精度を向上させている．評価実験では，各パーツ単体で検出した場合とパーツを統合した場合の検出性能の比較実験を行う．また，距離情報を用いた統合手法と距離情報を用いない統合手法についても検出性能の比較実験を行う．これらの評価実験を通して，距離情報を用いたパーツ統合を導入することで検出性能が向上することを示す．

4 章では，新たなオンライン学習として，Real AdaBoost を用いた追跡手法について述べる．Real AdaBoost を用いることで，少ない弱識別器数で物体を追跡することができ，オンライン学習の計算コストを削減できる．また，Real AdaBoost で学習した確率密度関数を導入することで，遮蔽物などへの追跡対象以外への乗り移りを軽減することができる．評価実験で，顔の向き変化が生じるようなシーンにおいて，追跡性能が向上していることを示す．また，人体の追跡が可能であることについても示す．

5 章では，形状変化が生じるような物体を追跡するために，追跡物体の変化に合わせて，形状の変化が可能な弱識別器を提案する．提案手法を姿勢変化を伴う人物の追跡に用いた場合について，評価実験を行い，従来の弱識別器構造に比べて追跡性能が向上していることを示す．

6 章は，物体の検出と同時に行動認識が可能なマルチクラスな認識手法である Boosted Randomized Trees を提案する．提案手法は，Randomized Trees のノード生成時に，クラス尤度をもとにした階層的な認識クラスの定義方法および，Joint Boosting による事前特徴選択を導入することで，認識に有効な特徴を効率的に選択することが可能である．これにより，上位ノードは人体検出，下位ノードは姿勢推定または行動認識を階層的に行うことができる．評価実験では，従来の Randomized Trees

に比べて，人体検出率および行動認識率が向上していることを示す．

7 章は，結論であり，本研究についてまとめる．

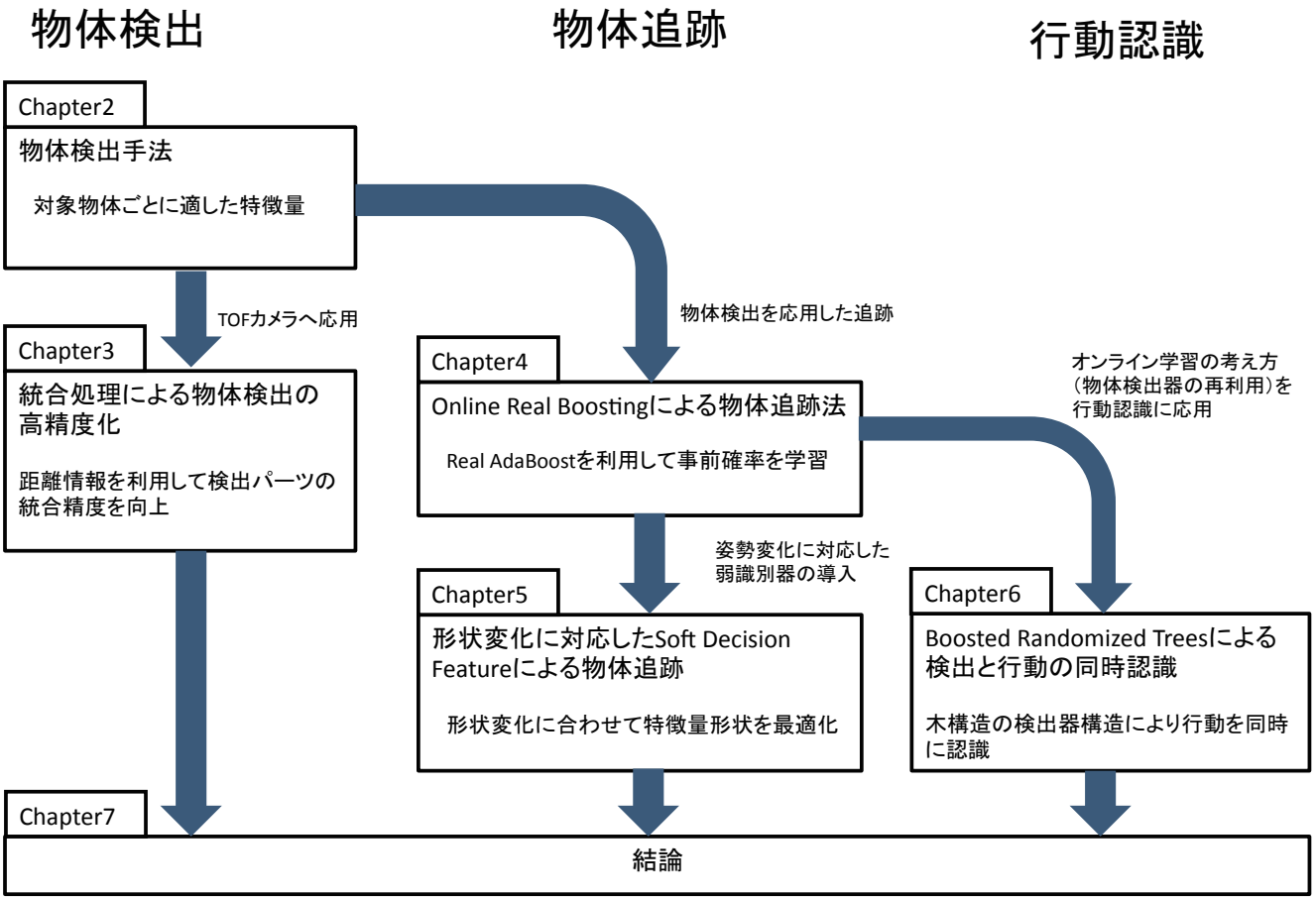


図 1.2: 本論文の構成

第2章

画像局所特徴量による物体検出

特徴量は，対象物体全体の情報を用いる特徴量と検出対象の局所的な領域の情報に着目する特徴量がある．特に局所領域に着目した特徴量は，統計的学習手法の一つであるブースティングにより学習することで高精度な物体検出手法が実現できることが知られている．本章では，物体検出に用いられる画像局所特徴量とブースティングを用いた学習方法について述べる．また，検出対象ごとにどのような特徴量が検出に適しているかを予備実験により明確にする．

2.1 物体検出について

物体検出は，画像中から対象物体の位置を同定する技術であり，コンピュータビジョンの分野で最も重要なトピックの 1 つとして古くから研究が行われている [12]. 物体検出の代表的な対象は顔であり，近年では顔検出がデジタルカメラや携帯電話などへ搭載され，物体検出の実用化が進んでいる. 物体検出の流れを図 2.1 に示す. 物体検出は，事前学習と検出処理の 2 つの処理からなる. 事前学習では，対象物体を検出するための検出器を構築する. 検出器は，人の事前知識をもとにした手法 [13] や人の事前知識を利用せず統計的な学習による手法 [14] により構築される. また，検出処理は，入力画像中を網羅的に探索し，検出器が対象物体と識別する領域を抽出する. 検出対象の物体は，物体の向きや照明変化などにより見えが変化する. 例えば顔の場合，顔の向きや照明変化だけでなく，表情変化や年齢，性別，個人などによる変化も生じる. このように変化する対象を検出するためには，汎用的に検出が可能な検出器が必要となる. そのため，近年は，あらかじめ大量の学習サンプルを用いて統計的アプローチをより，検出器を学習する方法が一般的である. すべての人物の顔や各顔の変化に対応した画像を集めた場合，非常に膨大なデータ量となるため，対象物体に共通な要素を最小限抽出する必要がある. この際に，抽出する要素が特徴量であり，最小限抽出する方法が学習手法である. これらには密接な関係があり，新たな学習手法が提案されると，それに応じて性能を向上させる特徴量が提案されている.

特徴量は，輝度分布や Wavelet のように，対象物体全体の情報を用いる特徴量と検出対象の局所的な領域の情報に着目する特徴量がある. 特に局所領域に着目した特徴量は，統計的学習手法の一つであるブースティングにより学習することで高精度な物体検出手法が実現できることが知られている. 本章では，物体検出に用いる特徴量とその特徴量の学習方法について述べた後，対象物体ごとにどのような特徴量が有効であるかの評価実験を行う.

2.2 画像全体に着目した特徴量

検出物体を捉えるための画像特徴量の一つとして，対象物全体を捉える特徴量がある. 例えば，対象物全体の輝度や色の分布をヒストグラム化し，多次元ベクトルで表現方法や対象物全体の空間周波数に着目する Wavelet 特徴量などである.

これらの特徴量を用いた学習手法は，二つに分けることができる. 1 つ目は，研究者の経験に基づいた知識ベースの学習手法であり，初期の物体検出の手法に多く用いられている. 研究者の知識ベースの手法として，顔のエッジのある領域とない領域にわけた顔のモデルを作成し，このモデルを用いて顔検出を行う手法 [12] や複数の顔テンプレートを用いた顔検出法 [13]，顔の器官の関係に基づいた顔検出法 [15] などが挙げられる. これら知識ベースの手法は，複雑なモデルを作ることが難しいため，比較的少ない特徴量を用いた手法が多い.

2 つ目は，ニューラルネットワークや線形判別分析法 (LDA)，サポートベクターマシン (SVM) など統計的学習手法を用いた学習手法である. これらは，知識ベースの手法よりも大量の学習サンプルを用いることで高精度な検出を行うことができる. 知識ベースの学習手法では，研究者の知識に基づ

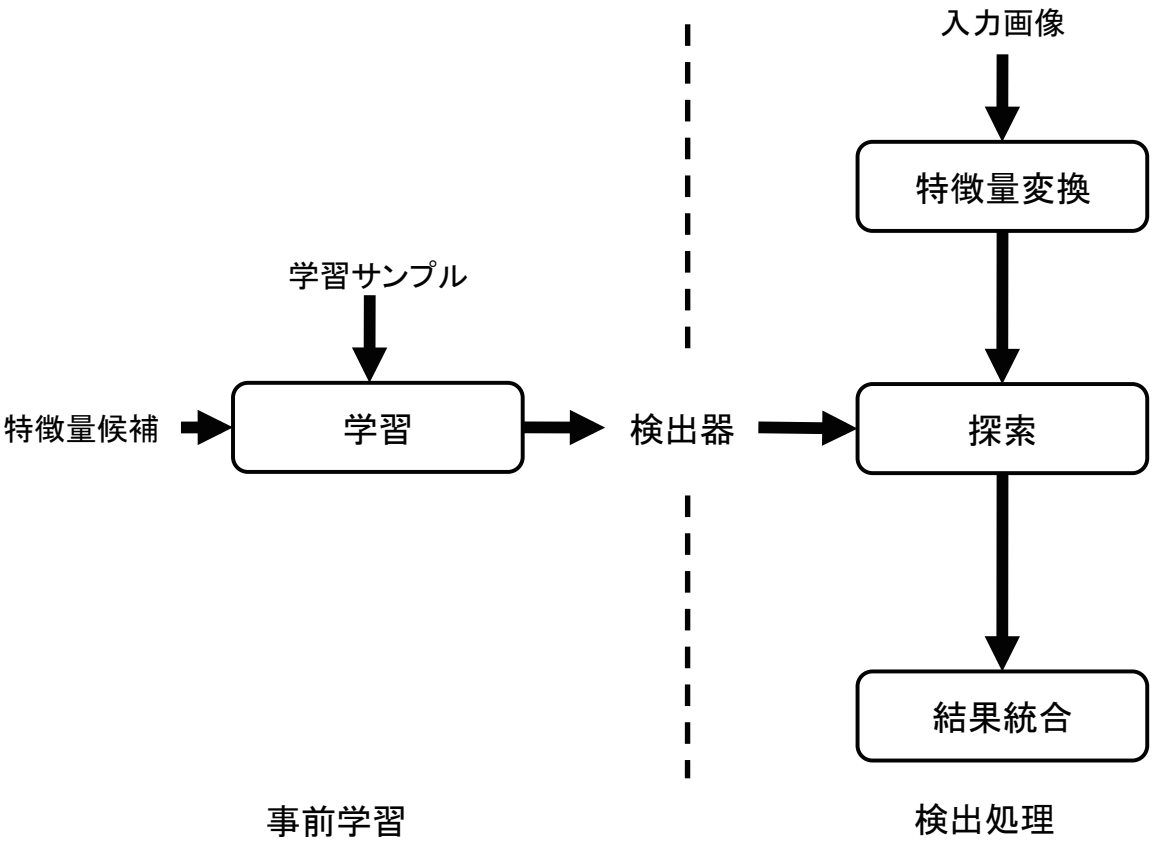


図 2.1: 物体検出の流れ

いてどこの領域に着目するかを定義していた．統計的学習手法を用いた手法では，物体全体から抽出した特徴量のうち，多数の学習サンプルに共通する情報に着目した特徴量が自動的に定義される．これらの統計的学習を用いた代表的な手法として，Rowley らのニューラルネットワークを用いた顔検出法 [16] や Papageorgiou らの人体検出法 [17] がある．

画像全体に着目する特徴量は，ヒストグラム作成やエッジ抽出手法などの基本的な画像処理手法により容易に実装が可能という特長がある．一方，対象物体全体を 1 つの特徴量として表現するため，形状変化や照明変化等の変形に弱いという欠点がある．

2.3 局所領域に着目した特徴量

前節で述べた特徴量は，対象画像全体を 1 つのベクトル情報として捉える特徴量であるため，部分的な形状変化や照明変化に対して頑健とはいえない．対象物体全体を捉えるのではなく，対象物体の局所領域の情報を捉える特徴量がある．局所的な情報を捉えるため，一つ一つの特徴量は対象物体の情報を多く含まない Low level な特徴量となる．しかし，多数の局所領域から得られた Low level

な特徴量を組み合わせて用いることで，より高次な情報として抽出することができる．

特徴量は輝度情報をベースにしたものと，エッジ情報をベースにしたものに分類することができる．以下に，代表的な輝度情報およびエッジ情報をベースとした特徴量を示す．これらの特徴量について，詳細に述べる．

- 輝度に着目した特徴量
  - Haar-like 特徴量 [14]
  - ピクセル差分特徴量 [18]
- エッジに着目した特徴量
  - Edge of Orientation Histograms(EOH) 特徴量 [19]
  - Histograms of Oriented Gradients(HOG) 特徴量 [20]
  - Edgelet 特徴量 [7]

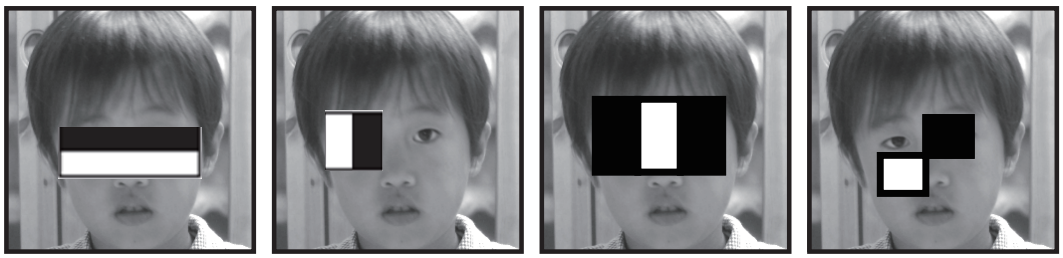


図 2.2: Haar-like 特徴量

2.3.1 Haar-like 特徴量

Haar-like 特徴量は，図 2.2 に示すように，白の領域 $r1$ と黒の領域 $r2$ の 2 つの領域の輝度差を式 (2.1) より算出し Haar-like 特徴量としている．

$$H(r1, r2) = S(r1) - S(r2) \tag{2.1}$$

ここで，$S(r)$ は，領域 r の輝度和を算出する関数である．2 つの領域のパターンは多数あり，図 2.2 に示すようなパターンが一般的に用いられる．これらのパターンは，縦方向の明暗差，横方向の明暗差，斜め方向の明暗差を捉えることが可能である．他にも，3 つの領域に着目することで線分を捉えるパターンもある．これらのパターンをベースとして，様々な形状が提案されている [21]．

このような基本となるパターンをもとに，位置やスケールを網羅的に変化させることで膨大な特徴量を生成する．これらのパターンから対象物体に適した Haar-like 特徴量をブースティングにより選択する方法が Viola らにより提案されている [14].

Haar-like 特徴量は Viola らが顔検出に用いて以降，非常に多くの物体認識手法に用いられている [22][23][24][25]．また，Haar-like 特徴量は，積分画像を用いることで高速に特徴量を算出することが可能である．積分画像を用いることで，各領域の合計輝度を領域の大きさに関わらず，4 つの値の加減算で算出できるため，計算量を大幅に削減することが可能である．

2.3.2 ピクセル差分特徴量

ピクセル差分特徴量は任意の画素間の輝度差に着目した特徴量である [18]．前述の Haar-like 特徴量は隣接する局所領域間での輝度差に着目している．一方，ピクセル差分特徴量は，図 2.3 に示すように，離れた任意の画素間での輝度差を捉えることができる．ピクセル差分特徴量 $P(I)$ は，式 (2.2) のように，2 つの画素間 I_1, I_2 での輝度差を特徴量として捉えている．

$$P(I) = I_1 - I_2 \tag{2.2}$$

ここで，I は入力画像の輝度値を表す．

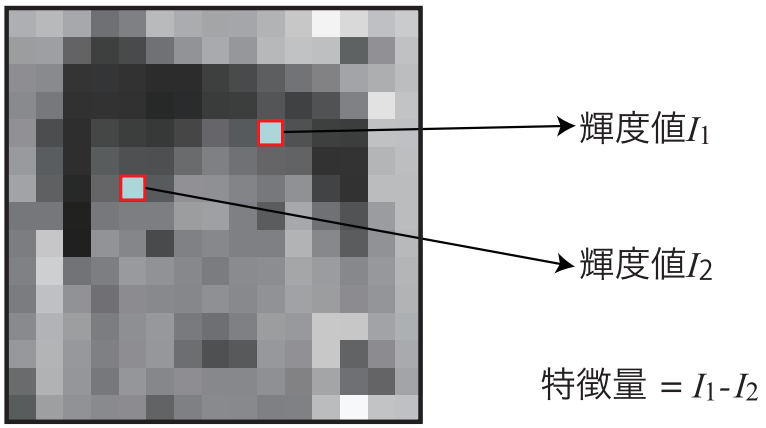


図 2.3: ピクセル差分特徴量

2.3.3 EOH 特徴量

EOH 特徴量は，1 つの局所領域内におけるエッジ勾配の関係に着目した特徴量である [19]．EOH 特徴量の算出には，図 2.4 の頬の領域のようにソーベルフィルタによりエッジ画像を作成し，式 (2.3)(2.4) のように各画素のエッジ強度 $G(x, y)$ とエッジ方向 $\theta(x, y)$ を算出する．

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \tag{2.3}$$

$G_x(x,y)$ は，x 方向のエッジ画像，$G_y(x,y)$ は y 方向のエッジ画像である．

$$\theta(x,y) = \tan^{-1}\left(\frac{G_x(x,y)}{G_y(x,y)}\right) \tag{2.4}$$

式 (2.5) より，エッジ方向 θ を k 段階に量子化し，各方向のエッジ強度画像を作成する．

$$\psi_k(x,y) = \begin{cases} G(x,y) & if \quad \theta(x,y) \in bin_k \\ 0 & otherwise \end{cases} \tag{2.5}$$

領域 R における方向 k の累積エッジ強度は式 (2.6) のように算出する．領域 R は，位置とスケールを網羅的に変化させることで，異なる位置と異なる空間的解像度に着目した特徴量を生成することが可能となる．

$$E_k(R) = \sum_{(x,y)\in R} \psi_k(x,y) \tag{2.6}$$

EOH 特徴量は，式 (2.7) のように方向 k_1 に対する累積エッジ強度と方向 k_2 に対する累積エッジ強度の累計の比を算出する．

$$A_{k1,k2}(R) = \frac{E_{k1}(R) + \epsilon}{E_{k2}(R) + \epsilon} \tag{2.7}$$

ここで，$E_k(R)$ は，R 領域における k 方向の勾配を求める関数，ϵ は 0 による除算を防ぐための係数である．エッジ方向は 4 から 8 方向に量子化している．

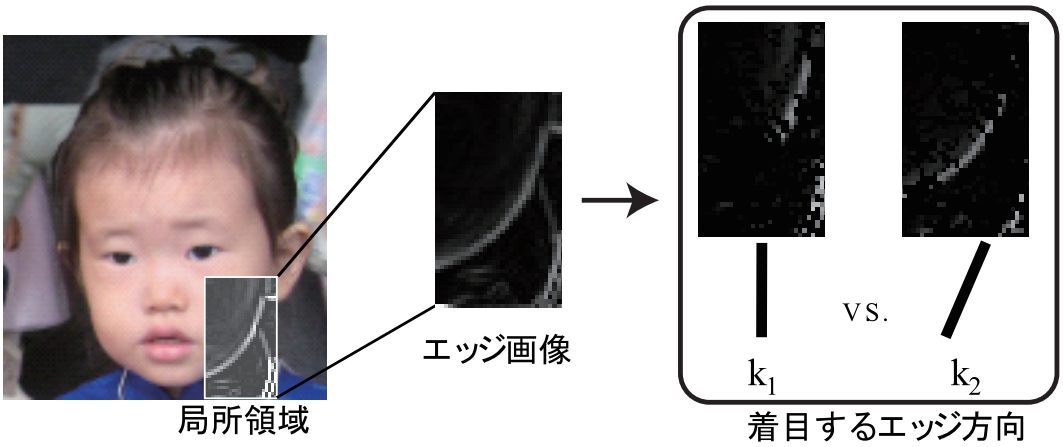


図 2.4: EOH 特徴量

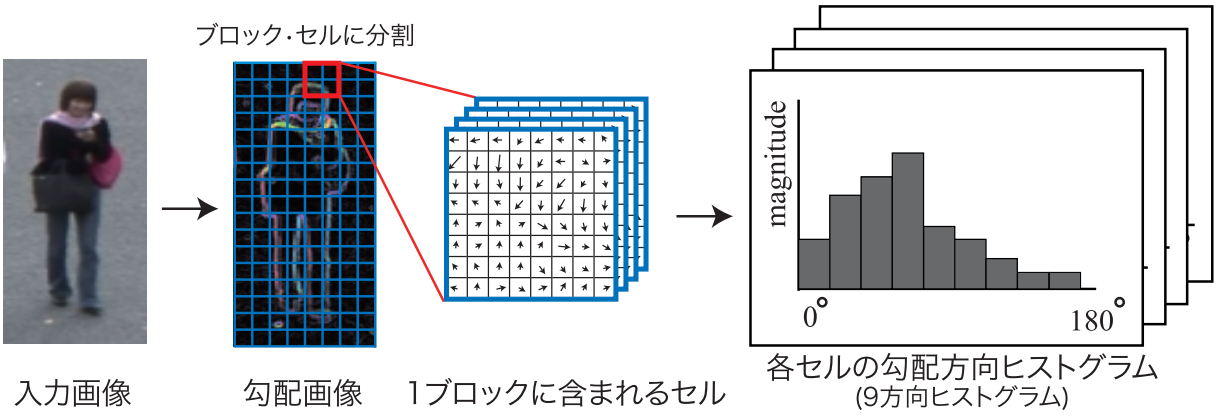


図 2.5: HOG 特徴量

2.3.4 HOG 特徴量

HOG 特徴量は 1 つの局所領域内におけるエッジ方向毎のエッジ強度に着目した特徴量である．局所領域は図 2.5 に示すように，ブロックとして表現され，ブロックには複数のセルと呼ばれるさらに小さな局所領域から構成される．特徴量は，セルのエッジ方向毎のエッジ強度ヒストグラムを結合した多次元ベクトルで表現される．Dalal らはセルサイズを 8×8 画素，エッジ方向を 9 方向とし，ブロックは 2×2 セルで構成される 36 次元ベクトルとしている．HOG 特徴量はブロックに対して，L1 ノルムまたは L2 ノルムにより，ブロックを正規化している．これにより，全体的な明暗条件によるエッジ強度がばらつく場合でも特徴量が大きく変化しないようにしている．

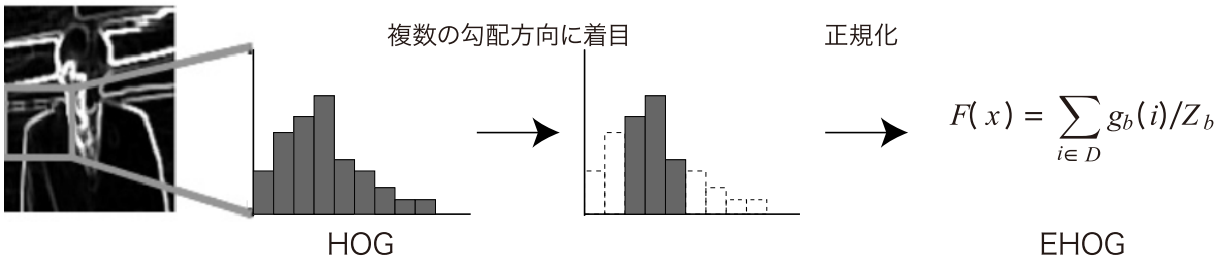


図 2.6: EHOG 特徴量

HOG 特徴量は，複数のセル内に含まれる勾配方向毎の情報を特徴量とする高次元特徴量である．Zhu ら [26] は SVM，LDA を用いて次元圧縮する方法を提案しているが，計算コストが高いなどの問題がある．計算コストを削減する方法として，HOG 特徴量を改良した EHOG(Extend Histograms of Oriented Gradients) 特徴量が提案されている [27]．EHOG は，HOG とは異なりブロック内に複数のセルを持たない．EHOG は図 2.6 のように，1 つのブロック内における複数のエッジ方向の総和に着目する．そのため，ブロックから得られる特徴量が 1 次元となり，HOG 特徴量と比べ高速に検

出を行うことが可能となる．特徴量 $F_b(D)$ は，式 (2.8) のように算出される．

$$F_b(D) = \sum_{b \in D} g(b)/Z_b \tag{2.8}$$

ここで，$g(b)$ は勾配方向 $b = \{1, \cdots, B\}$ を要素とするエッジ強度のヒストグラムであり，D は b の集合である．

また，Z_b は全勾配方向のヒストグラムの総和であり，式 (2.9) のように求められる．

$$Z_b = \sum_{b=1}^{B} g(b) \tag{2.9}$$

ここで，B は全勾配方向数であり，Hou らは 9 としている．

2.3.5 Edgelet 特徴量

EOH 特徴量や HOG 特徴量は局所領域内におけるエッジの分布に着目しているが，Edgelet 特徴量は図 2.7(a)(b)(c) のように，エッジの部分的な繋がりを捉える特徴量である．また，EOH 特徴量や HOG 特徴量は，入力画像の着目する局所領域内でのエッジ情報をもとに特徴量を算出しているが，Edgelet 特徴量は，定義された形状パターンと入力画像の局所領域間でのエッジ方向の差異をもとに特徴量を算出している．形状パターンは，直線的な繋がり (a) や，円弧 (b) およびそれらの対称性 (c) を捉えるものであり，ブースティングにより物体の検出に有効な形状を選択する．

Edgelet 特徴量の形状パターン E は，図 2.7(d) の左側のように K 個の画素から構成されている．各画素 u_i は着目するエッジ方向 V_i^E を持っており，図 2.7(d) のようにエッジ方向 V_i^E は 6 方向に量子化した値である．入力画像 I に対する Edgelet 特徴量 $S(I)$ を式 (2.10) に示す．Edgelet 特徴量 $S(I)$，形状パターン u_i の位置に相当するエッジの強度 $E^I(u_i)$ と，形状パターンと入力画像のエッジ方向の差異をもとにした近似類似度関数 $l[\cdot]$ の内積を K 個の要素について累積し，正規化した値である．

$$S(I) = (\frac{1}{K}) \sum_{i=1}^{K} E^I(u_i) l[V^I(u_i) - V_i^E] \tag{2.10}$$

エッジ強度 E^I はソーベルフィルタを用いて作成したエッジ画像 E の画素値から得ることができる．

また，近似類似度関数 $l[\cdot]$ は式 (2.11) に示すように，入力画像 I における i 番目の要素位置のエッジ方向 $V^I(u_i)$ と Edgelet 特徴量の形状パターンにおける i 番目の要素のエッジ方向 V_i^E との差をもとに算出される．

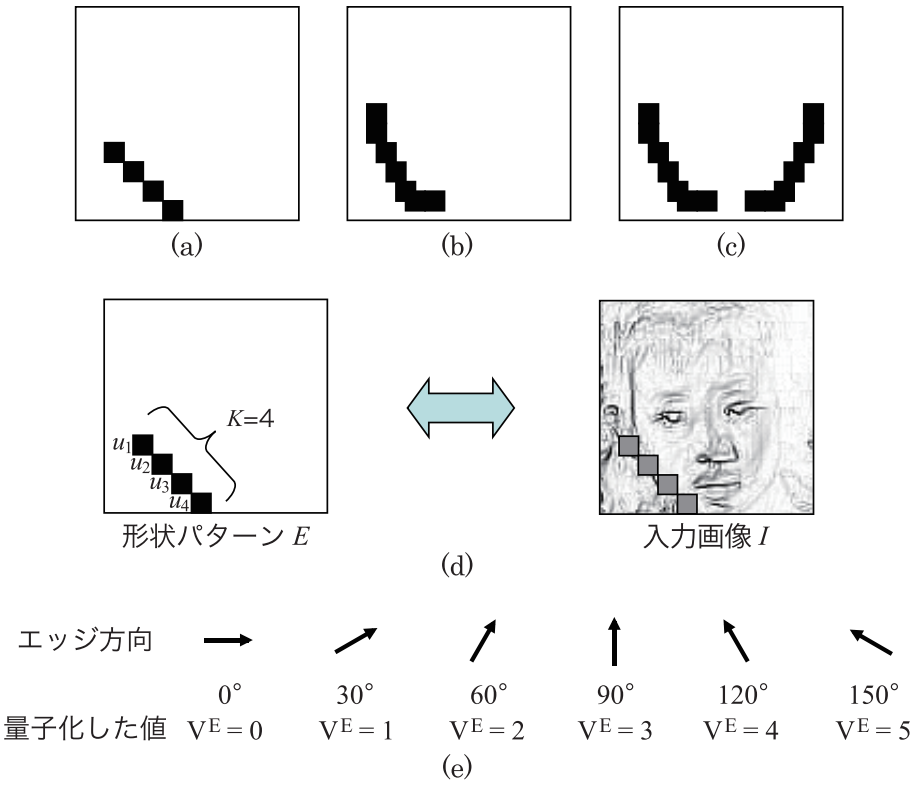


図 2.7: Edgelet 特徴量

$$l[\cdot] = \begin{cases} 1 & x = 0 \\ 0.8 & x = \pm 1, \pm 5 \\ 0.5 & x = \pm 2, \pm 4 \\ 0 & x = \pm 3 \end{cases} \tag{2.11}$$

ここで，$V^I(u_i)$ は V_i^E と同様に，6 段階へ量子化しており，x はエッジ方向 $V^I(u_i)$ と V_i^E の差である．x は -5 から 5 までの範囲の値であり，それぞれの値に応じて，式 (2.11) のような値をとる．

2.3.6 局所領域に着目した特徴量の特長

本節で述べた特徴量について，表 2.1 にまとめる．局所領域に着目する特徴量は，画像全体に着目する特徴量と比較して膨大な特徴次元数となるが，ブースティングと組み合わせることにより識別に有効な特徴量のみを用いる検出器の学習が可能となる．一方，統計的学習手法であるため，大量の学習サンプルが必要という欠点がある．

表 2.1: 局所領域に着目した特徴量の特長

長所	ブースティングと組み合わせることで， 研究者の事前知識なく検出性能の高い検出器の学習が可能
短所	最適な特徴量を選ぶためには大量のサンプルが必要

2.4 ブースティングによる特徴量の学習

局所領域に着目した特徴量は，着目する局所領域の位置や大きさ，形を変えることで膨大な特徴量候補を生成することができる．その中から，検出に有効な特徴量を選択する必要がある．特徴量の選択には，ブースティングが多く用いられる [7][14].

ブースティングは膨大な特徴量候補から，与えられた学習サンプルを検出するために適した特徴量の選択をすることができる．例えば，顔をサンプルとして与えた場合，特徴量候補の中から顔に共通する特徴量，目の付近に着目したもの，口の周りに着目したものなどを自動的に選択することができる．すなわち，研究者の事前知識なく，自動的に最適な特徴量を選択することが可能である．以下にブースティングのアルゴリズムについて述べる．

ブースティングは式 (2.12) に示すように，T 個の弱識別器 $h_t(x)$ を重みつき線形結合した強識別器 $H(x)$ として学習する．

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{2.12}$$

弱識別器 $h_t(x)$ は対象物体であるかどうかを出力する．出力値はブースティングの手法により異なるが，AdaBoost[28] は 0 または 1 の 2 値，Real AdaBoost[29] は実数で出力する．局所領域に着目した特徴量を用いて，ブースティングにより学習する流れを図 2.8 に示す．

検出対象と非検出対象の学習サンプルを多数用意し，各サンプルに重みを設定する．初期時はすべて同じ重みとする．また，特徴量の位置や形状を変えた特徴量候補を作成し，特徴量候補として用意する (a)．次に，特徴量候補から学習サンプルを最も正しく識別できる特徴量候補を弱識別器として選択する (b)．選択された弱識別器により，サンプル画像の重みを更新する (c)．正しく識別できたサンプルの重みは小さくし，誤識別したサンプルには大きな重みとする．これにより，次の選択時に正しく識別できなかったサンプルについて，正しく識別することが可能な弱識別器が選択されるようになる．これらの処理は，全ての学習サンプルが正しく識別できるまで，もしくは予め決めた弱識別器数となるまで繰り返し，強識別器を学習する (d)．このように，ブースティングを用いることで研究者の知識を必要とせず，多数の特徴量候補の中から対象物体の検出に有効な特徴量を学習することができる．

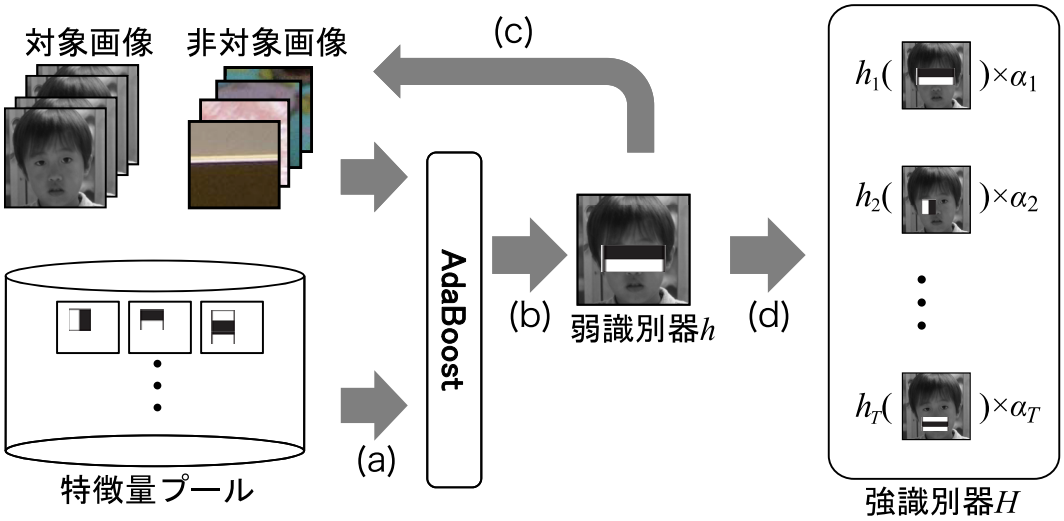


図 2.8: ブースティングによる学習の流れ

2.5 画像局所特徴量の比較

複数の検出対象を例に，局所領域に着目した特徴量の性能を評価するために比較実験を行う．特徴量は検出対象により向き不向きがあるため，顔と車，人の 3 つを検出対象とし，検出対象毎にどのような特徴量が有効であるか検証する．比較を行う特徴量は，Haar-like 特徴量，EOH 特徴量，Edgelet 特徴量，HOG 特徴量の 4 つとする．

2.5.1 実験概要

学習と評価の実験には，公開されているデータベースと Web から収集した画像を利用する．利用するデータベースを下記に示す．

- 顔データセット
  - 学習 : Web から収集した約 5000 枚
  - 評価 : MIT+CMU Frontal Face Image[30]
- 車データセット
  - 学習 : UIUC Cars [31]
  - 評価 : UIUC Cars [31]
- 人データセット
  - 学習 : INRIA Person Dataset[32]
  - 評価 : USC Pedestrian Detection Test Set[33]

比較実験では，Viola らが提案したカスケード型 AdaBoost により学習を行う．カスケード型 AdaBoost は，段階的に計算量の異なる強識別器を直列に繋げた識別器である．カスケードの各段が学習用検出対象サンプルを通過させる割合を $D_r(0 < D_r < 1)$，学習用非検出対象サンプルを通過させる割合を $F_p(0 < F_p < 1)$ としたとき，カスケード n 段通過後は，$(F_p)^n$ だけ通過することになる．カスケード構造により，非検出対象のサンプルは比較的早い段階で判別することができるため，計算効率が良い．

実験は特徴量を平等に評価するために，$D_r = 0.995$，$F_p = 0.300$ に固定し，カスケードの段数の最大 $n_{max} = 16$，1 つの強識別器に用いる弱識別器の最大数 $h_{max} = 100$ に統一した．

2.5.2 評価方法

評価実験は，評価用サンプルを使用し，対象の物体検出を行う．検出結果より，対象物の検出率と 1 枚の画像中における平均誤検出数を算出し，ROC(Receiver Operating Characteristic) カーブにより比較を行う．ROC カーブの縦軸に検出率，横軸に誤検出率をプロットすることで，各特徴量の性能を比較することができる．ROC カーブは，グラフの左上ほど検出器の性能が高いことを表す．検出率と誤検出率は式 (2.13)(2.14) より求めることができる．

$$\text{検出率} = \frac{\text{検出した正解数}}{\text{検出対象数}} \tag{2.13}$$

$$\text{平均誤検出数} = \frac{\text{誤検出した数}}{\text{評価用画像数}} \tag{2.14}$$

2.5.3 実験結果

図 2.9，図 2.10，図 2.11 に各検出対象における特徴量ごとの検出性能を示す．実験結果より検出対象と特徴量によって検出性能が異なることがわかる．

顔の場合では，輝度に着目した特徴量である Haar-like 特徴量が最も検出精度が良く，次にエッジに着目した特徴量である EOH 特徴量，Edgelet 特徴量，HOG 特徴量の順となっている．顔の特徴として，目，鼻，口などの器官はどのような人でも同じ位置にあり，形状変化が少なく，比較的剛体に近いため，顔の器官を捉えることができる Haar-like 特徴量の検出性能が最も高いと考えられる．

車の場合では，全ての特徴量において同程度の検出精度が得られている．車画像は，顔と同様に形状変化が少なく，テクスチャの多様さも少ない．そのため，エッジに着目する特徴量も輝度に着目する特徴量も高精度な検出を行うことができたと考えられる．

人の場合では，顔や車と比べて検出精度が低いことがわかる．これは，顔や車は比較的剛体に近い物体であるのに対して，人は姿勢の変化による形状変化が大きい．また，画像中での人同士の重なりによるオクルージョンも多く発生するため，顔や車と比べて検出精度が低くなる．このような問題に対しては，エッジに着目した HOG 特徴量が最も検出精度が良く，EOH 特徴量，Edgelet 特徴量，

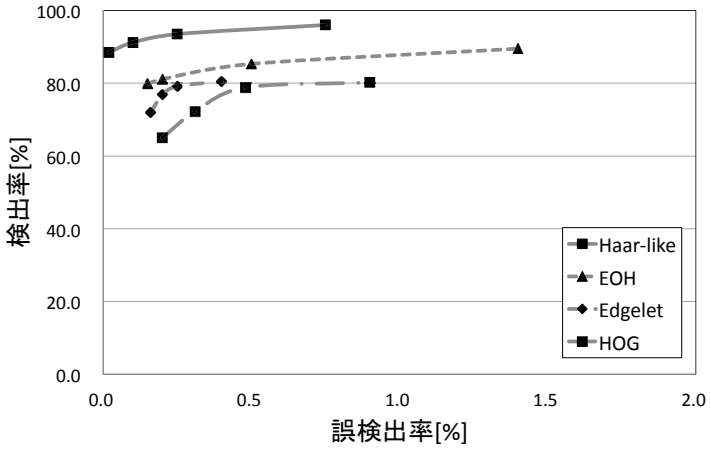


図 2.9: 顔画像における検出性能の比較実験

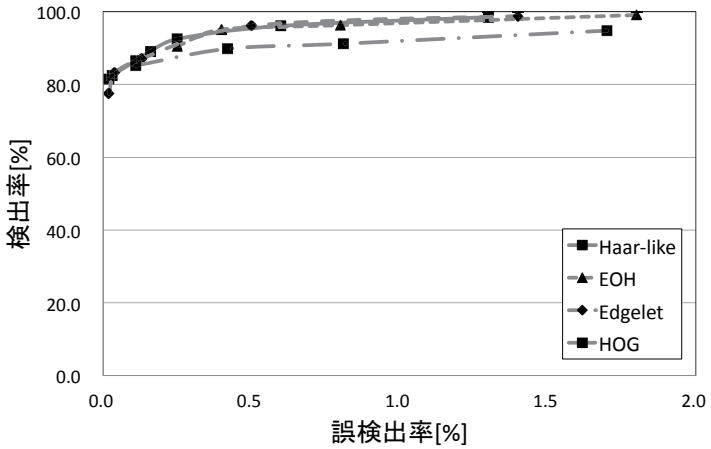


図 2.10: 車画像における検出性能の比較実験

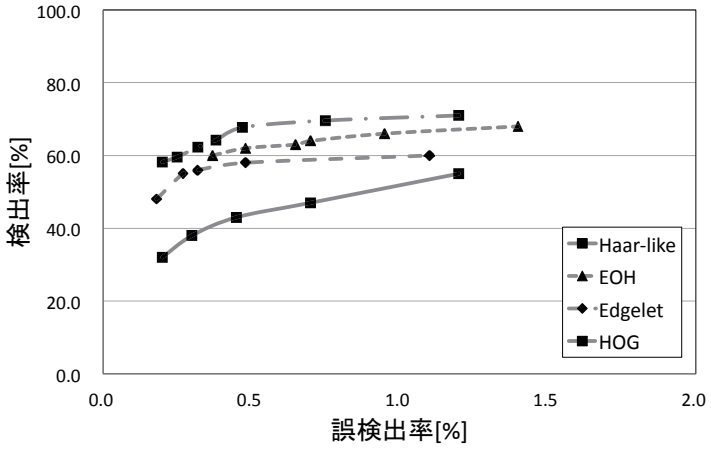
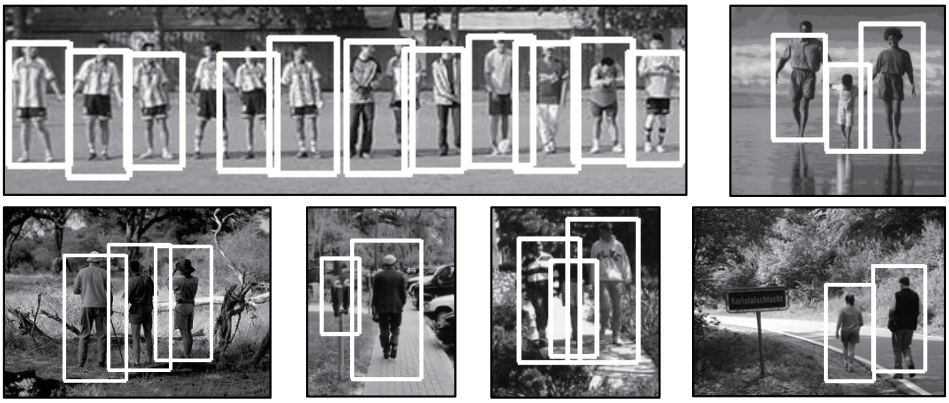


図 2.11: 人画像における検出性能の比較実験

(a) 顔検出 (Haar-like)

(b) 車両検出 (Edgelet)



(c) 人検出 (HOG)

図 2.12: 物体検出例

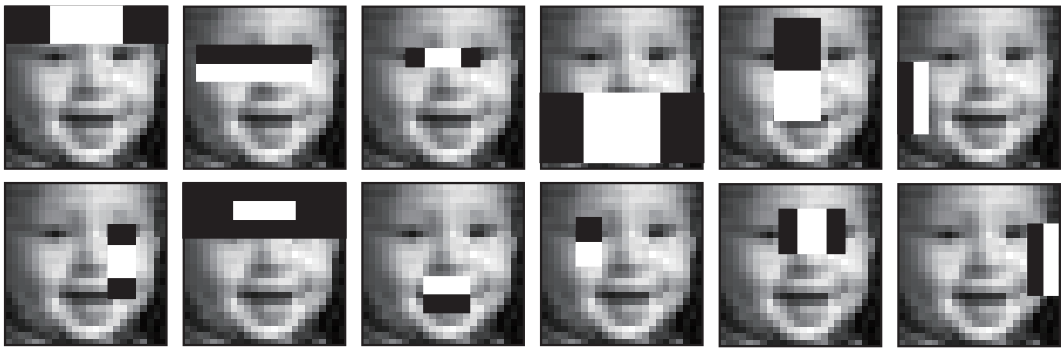


図 2.13: 顔データにおいて選択された Haar-like 特徴量の一部

Haar-like 特徴量の順になっている．エッジに着目した特徴量は，輝度に着目した特徴量よりも，衣服の色の違いに対してロバストな特徴量を抽出することができるためである．エッジに着目した特徴量の中でも，HOG 特徴量と EOH 特徴量は検出精度が高い．これは，局所領域内でエッジの方向ヒストグラムを作成することにより，局所領域内での若干の形状変化を吸収できるためである．

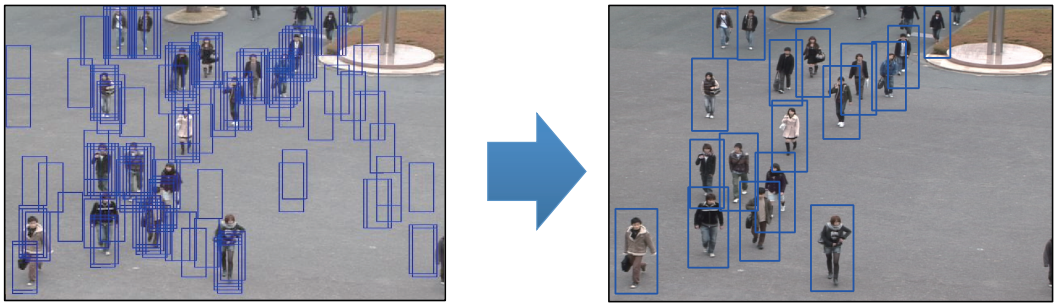
図 2.12 に検出結果の例を示す．顔検出は Haar-like 特徴量，車検出は Edgelet 特徴量，人検出は HOG 特徴量による検出結果である．

2.5.4　選択された特徴量

識別器の構築にブースティングを用いた場合，多数の特徴量候補の中から識別に有効な特徴量を自動的に選択するため，容易に識別に有効な特徴量を把握することができる．実験において，顔データセットと Haar-like 特徴量を用いて学習した結果より，選択された Haar-like 特徴量の可視化結果を図 2.13 に示す．図中の顔画像は，学習に用いた全ての顔画像から得られる顔の平均輝度画像である．平均輝度画像は，全ての顔画像に共通する輝度を表わしている．作成された平均輝度画像に着目すると，額や頬，鼻柱等の輝度値が高く，目や口等の輝度値が低いことがわかる．そのため，これらの輝度値の違いを捉えることで顔を検出することが可能である．選択された Haar-like 特徴量に注目すると，額，目，鼻の明暗差を捉えるような特徴量が多く選択されていることがわかる．このように Haar-like 特徴量は，学習データセットに含まれる検出対象の共通する明暗差を捉えることが可能であり，高精度な物体検出を行うことができる．

2.6　検出結果の統合

画像中を網羅的に探索処理すると，図 2.14(a) のように，1 つの物体に対して複数の検出結果を得ることができる．そして，図 2.14(b) のように，各物体について 1 つの検出結果として統合し，最終的な検出結果を出力する必要がある．ここでは，各物体に対して行う統合処理について述べる．図 2.15 に統合処理における処理画像を示す．



(a)統合前　　(b)統合後

図 2.14: 統合処理の有無よる検出結果例

統合処理では，統合前の検出結果 (図 2.15(a)) から中心座標を算出し (図 2.15(b))，中心座標間の距離から矩形をクラスタリングし (図 2.15c))，最終的な統合後の検出結果 (図 2.15(d)) を得る．検出矩形のクラスタリングは，クラスタリング手法の 1 つである Mean Shift 法を用いるのが一般的である．Mean Shift クラスタリングは，以下の手順により同一物体の統合を行う．最初に，$i = 1 \ \ (i = 1, .., n)$ の検出矩形の中心座標 x_1 を統合後の中心座標 p_1 とする．Mean Shift クラスタリングは，局大値へ移動するための Mean Shift Vector を算出し，収束するまで移動を繰り返す．Mean Shift Vector は式 (2.15) のようになる．

$$p_{j+1} = \frac{\sum_{i=1}^{n} x_i k(\| \frac{p_j - x_i}{h} \|^2)}{\sum_{i=1}^{n} k(\| \frac{p_j - x_i}{h} \|^2)} \tag{2.15}$$

ここで，x_i は i 番目の検出矩形の中心座標，h はバンド幅，p_j は j 番目の繰り返し回数における統合後の中心座標である．また，k はカーネル関数であり，ここでは正規分布とする．p_{j+1} と p_j の差が一定の値より小さくなるまで繰り返し行うことで，統合後の中心座標を得ることができる．

2.7　物体検出のまとめ

本章で述べた特徴量について，表 2.2 にまとめる．表 2.2 の項目にある候補数は，24×24 画素の顔画像から得られる特徴量の個数である．検出対象ごとの比較実験を通して，輝度に着目する特徴量は顔検出に適しているが，人体のように形状やテクスチャが多様な物体に対しては不向きである．一方，エッジに着目する特徴量は，人体等の形状やテクスチャが多様な物体に有効であることが分かった．3 章では，局所領域に着目した特徴量を用いたパーツベース検出器と距離画像を考慮した検出結果の統合方法による人物検出手法について提案する．

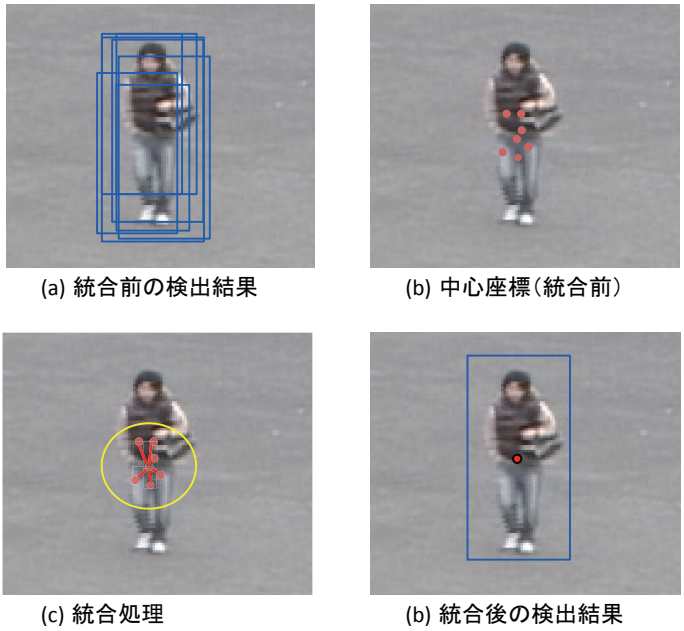


図 2.15: 検出結果の統合処理例

表 2.2: 特徴量の比較

特徴量	候補数	長所	短所
Haal-like	18 万	· 演算が非常に高速 (積分画像を用いた場合) · 顔や車両などの形状変化の少ない物体に有効	· 人等の形状やテクスチャが多様な物体には不向き
ピクセル差分	16 万	· 演算が非常に高速 · 顔や車両などの形状変化の少ない物体に有効	· 照明変化などの撮影条件により性能が低下
EOH	60 万	· 演算が高速 (積分ヒストグラムを用いた場合) · 形状変化の少ない顔のみでなく形状変化を含む人にも有効	· 低コントラストなどの撮影条件により性能が低下
HOG	900	· 人などの形状やテクスチャが多様な物体に有効	· 低解像度の物体や小さい物体の検出には不向き · 演算が遅い
Edgelet	60 万	· 形状変化が少なくテクスチャが多様な顔や車両などに有効	· 輪郭形状が多様な物体には不向き · 演算が遅い

第3章

統合処理による物体検出の高精度化

混雑シーンで，顔検出および人検出を用いる場合，人の重なりが生じるため検出できない場合が多い．そのようなシーンにおいて，高精度に物体を検出する手法として，パーツベースの検出手法がある．パーツベースの検出手法は各パーツをどのように統合するかが大きな問題である．本章では，パーツベースの人検出における統合時に，パーツ間の位置関係を定義するとともに，距離情報を用いることで統合エラーを削減し，検出率を向上させる方法を提案する．また，距離画像を用いた場合と用いない場合の比較実験を行う．

3.1 背景

監視カメラやデジタルカメラにおける人の位置同定において，顔検出および人検出を用いる場合，人の見えや大きさが性能に大きく関わる．顔検出を用いる場合，後ろ向きやマスク，サングラスなどの装飾品により，顔が見えない場合は検出できない．また，人検出は，体の一部が画面からはみ出している場合や人の重なりによる隠れが生じる場合，検出することができない．さらに混雑シーンでは，人の重なりにより検出できない場合が多い．このようなシーンにおいて，正確に人物を検出するためには，パーツベースの検出手法が有効である．しかし，パーツベースの検出手法は各パーツをどのように統合するかが大きな問題である．Wu らは，特徴量としてエッジベースの Edgelet 特徴を用いており，抽出されたエッジ特徴の情報と各パーツの位置関係から人物の位置を同定している [7]．Wu らの手法は，人が重なるような場合において，パーツ間の関連性を捉える事ができないため，異なる人物のパーツが統合されて検出が失敗するケースがある．

そこで，我々は，パーツベースの人検出における統合時に，パーツ間の位置関係を定義するとともに，距離情報を用いることで統合エラーを削減し，検出率を向上させる方法を提案する．提案手法では，距離情報を得るために，TOF(Time of Flight) カメラを用いる．TOF カメラは単眼カメラと LED 照明を組み合わせたカメラで，LED から照射される赤外光が反射して観測されるまでの時間差から距離情報を得ることができる．

3.2 距離画像の取得

距離情報を取得する方法として，2 台のカメラを用いたステレオ方式が一般的である [34][35][36]．近年，単眼カメラと LED を組み合わせて奥行き情報を取得可能な TOF(Time Of Flight) カメラが実用化されている．TOF カメラは，カメラ周辺に設置した LED から照射される赤外光が物体に反射し，カメラで観測されるまでの時間差から物体までの距離を計測することができる．図 3.1(a) に本研究で用いている TOF カメラ (MESA 社 SR-4000) を示す．SR-4000 は，0.8m から 5.0m までの距離情報をリアルタイムに取得することができる．SR-4000 はカメラの周辺に 24 個の LED が設置されている．図 3.1(b) はカメラから得られた反射強度画像，図 3.1(c) は距離情報を示している．反射強度画像は，LED から照射された赤外光が反射する光の強さを表しており，可視光の輝度画像に相当する．取得される画像サイズは，144×176 ピクセルであり，画角は 34.6° ×43.6° である．TOF カメラを用いることで，1 つのデバイスで輝度画像と距離情報の 2 つの情報をセンシングすることができる．

3.3 提案手法

実環境での監視カメラやデジタルカメラにおいて，人物の体全体が映っている場合は少ない．また，顔が見えない場合や装飾品で隠れている場合も多い．そこで，提案手法は，人全体が画面に

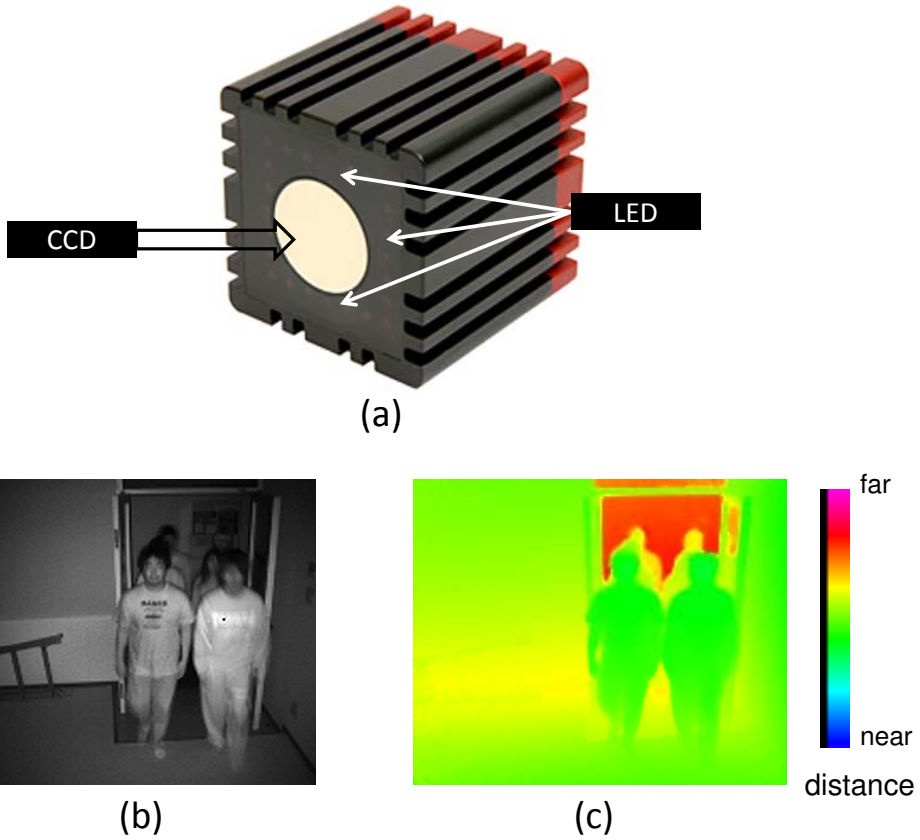


図 3.1: TOF カメラの外観と得られる画像

映っていない場合や顔が見えない場合でも人物を検出するために，パーツベースの検出手法を用いる．提案手法で対象とするパーツは顔と上半身である．

図 3.2 に提案手法の流れを示す．提案手法は，TOF カメラにより取得された反射強度画像から顔および上半身検出を行う．各パーツの検出処理は独立しており，探索ウインドウを変えながら画像中を網羅的に探索する．各検出処理により複数の顔および上半身領域を検出することができる．検出された領域は同一人物に対して，複数得られることがあるため，得られた顔および上半身領域を統合する．統合処理には，各検出結果および TOF カメラにより得られた距離情報を用い，距離情報を考慮した統合処理を行う．本章では，各検出処理および距離情報を考慮した検出結果の統合方法について述べる．

3.3.1 顔検出

本研究では，Viola らが提案した AdaBoost を用いたカスケード型の検出器を用いる．顔検出は，Viola らの手法をベースとして様々な手法が提案されており，特徴量として輝度情報に着目した手法，エッジ情報に着目した手法などがある．本研究では，輝度に着目した Haar-like 特徴量 [14] とエッジ

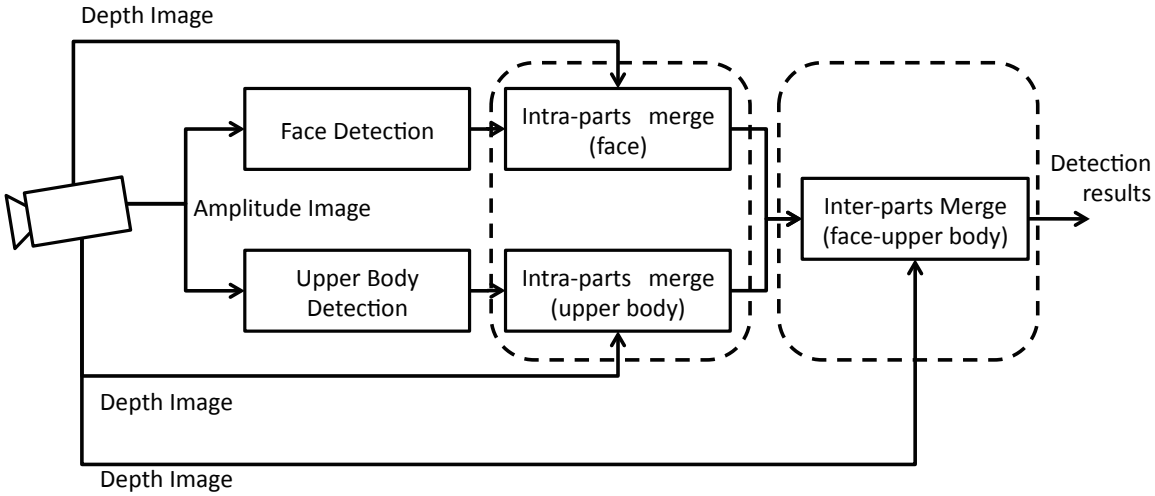


図 3.2: 提案手法のフレームワーク

に着目した EOH 特徴量 [19] の 2 つを用いる．検出器は，統計的学習手法の 1 つである AdaBoost[28] により検出に有効な特徴量を弱識別器として学習する．AdaBoost は式 (3.1) に示すように，T 個の弱識別器 $h_t(x)$ を重みつき線形結合した強識別器 $H(x)$ を学習する．

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{3.1}$$

弱識別器 $h_t(x)$ は対象物体であるかどうかを出力する．AdaBoost の場合，出力値は 0 または 1 の 2 値である．α_t は弱識別器の重みであり，与えられた学習サンプルを正しく識別できるほど値が大きくなり，識別能力が低い場合は小さな値となる．

本研究では，学習された強識別器を階層的な構造に配置するカスケード型の検出器構造を用いる．各強識別器において，顔と判断された場合，次の強識別器でさらに顔かどうかを識別する．一方，顔でないと判断された場合は，識別処理を終了し，次の強識別器での識別は行わない．カスケード型の検出器構造は，顔らしい領域に対しては，より詳細に識別処理を行い，明らかに顔でない領域は早期に認識処理を打ち切ることができる．これにより，検出処理を高速に行うことができる．

3.3.2 上半身検出

本研究で用いる上半身検出は，顔検出と同様に AdaBoost とカスケード型検出器構造を用いる．特徴量は，人検出で一般的に用いられる HOG 特徴量 [20] を用いる．提案手法では，HOG 特徴量の各ヒストグラムのビンを特徴量候補として，AdaBoost により識別に有効な特徴量を選択する．

3.3.3 距離情報を考慮した検出結果の統合法

提案手法では，パーツ内での統合とパーツ間の統合の 2 段階により統合を行う．統合処理は，重み付きクラスタリングや Mean Shift クラスタリングを用いる方法 [16] がある．

本研究では，距離情報を考慮した Mean Shift クラスタリングを提案し，それをパーツ内統合および，パーツ間統合の両統合処理で用いる．以下では，提案するクラスタリング法を説明した後，それを用いた両統合方式について述べる．

■ Mean Shift クラスタリング

Mean Shift クラスタリングは，クラスタ数をあらかじめ指定する必要がないクラスタリング手法である．最初に，$i = 1 \ \ (i = 1, .., n)$ の検出矩形の中心座標 x_1 を統合後の中心座標 p_1 とする．Mean Shift クラスタリングは，局大値へ移動するための Mean Shift Vector を算出し，収束するまで移動を繰り返す．Mean Shift Vector は式 (3.2) のようになる．

$$p_{j+1} = \frac{\sum_{i=1}^{n} x_i k(\| \frac{p_j - x_i}{h} \|^2)}{\sum_{i=1}^{n} k(\| \frac{p_j - x_i}{h} \|^2)} \tag{3.2}$$

ここで，x_i は i 番目の検出矩形の中心座標，h はバンド幅，p_j は j 番目の繰り返し回数における統合後の中心座標である．また，k はカーネル関数であり，ここでは正規分布とする．p_{j+1} と p_j の差が一定の値より小さくなるまで繰り返し行うことで，統合後の中心座標を得ることができる．

■ 距離情報を利用した Mean Shift クラスタリング

大きさの異なる検出矩形を統合する場合，Mean Shift クラスタリングは，バンド幅をどのように設定するかが問題となる．すなわち，小さな人物を検出する場合，バンド幅が大きいと異なる人物を誤って統合する．一方，大きな人物を検出する場合，バンド幅が小さいと一人の人物を複数に分割して検出する．カメラで撮影される人物の大きさはあらかじめ分からない場合がほとんどであるため，バンド幅の設定によっては，上記のような誤統合または分割が生じる．また，バンド幅を検出矩形サイズから決める場合においても，大きな検出矩形の近くに存在する小さな検出矩形は統合されやすくなる．そこで，本研究では距離情報を用いた項を追加することで，バンド幅による誤統合を防ぐ．距離情報を用いた Mean Shift Vector の算出は式 (3.3) および式 (3.4) のようになる．

$$p_{j+1} = \frac{\sum_{i=1}^{n} x_i g(x, d)}{\sum_{i=1}^{n} g(x, d)} \tag{3.3}$$

$$g(x, d) = \frac{1}{h_x h_d} k(\| \frac{p_j^x - x_i}{h_x} \|^2) k(\| \frac{p_j^d - d_i}{h_d} \|^2) \tag{3.4}$$

ここで，d は距離情報であり，x_j^x は統合後の中心座標，x_j^d は統合後の距離である．また，p は中心座標と距離を持つベクトルである．これにより，中心座標の距離から統合される場合でも，距離情報を考慮することで，異なる距離の検出矩形は統合されなくなる．

■ パーツ内統合

対象物体周辺では，異なる位置や大きさでも物体を検出する．そのため，検出結果を統合し，1 人の人物に対して，1 つの検出結果を出力する必要がある．本研究では，同一パーツに対する検出結果を距離情報を利用した Mean Shift クラスタリングにより統合する．図 3.3 に同一パーツ内での統合の流れを示す．Mean Shift クラスタリングを用いることで，位置関係から各パーツの検出結果を統合することができる．

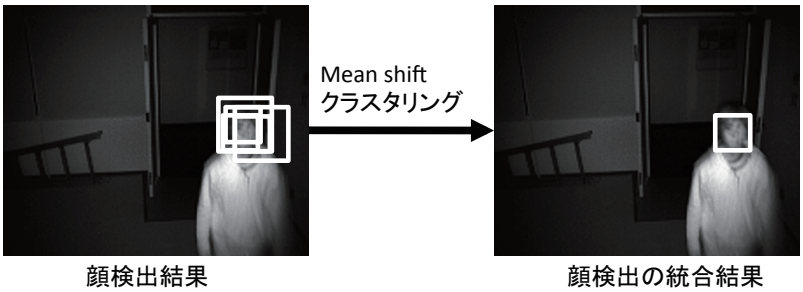


図 3.3: パーツ内の統合の流れ

■ パーツ間統合

パーツ内での検出矩形の統合後，パーツ間の統合を行う．異なるパーツを統合するために，上半身の検出矩形をから顔の位置を推定する．顔と上半身は，図 3.4 に示すような関係になっている．パーツ間の統合の流れを図 3.5 に示す．上半身検出結果から推定された顔の位置と顔検出結果を距離情報を考慮した Mean Shift クラスタリングにより統合する．統合した結果から，顔と上半身の関係から上半身の位置を推定し，人物位置として出力する．図 3.5 のように一人の場合は，顔検出結果と上半身検出結果が 1 つずつであり，位置関係から統合することも可能である．しかしながら，複数人が近傍にいる場合，位置関係だけでは異なる人物を誤って統合することがある．そのため，検出した結果の位置関係だけでなく，距離情報も重要となり，顔位置推定と距離情報を用いることで，同一人物の異なるパーツの統合が可能となる．

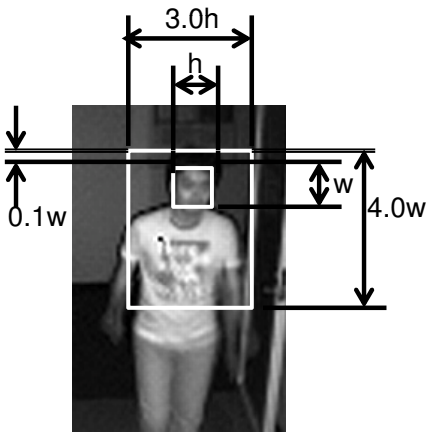


図 3.4: 顔と上半身の位置関係

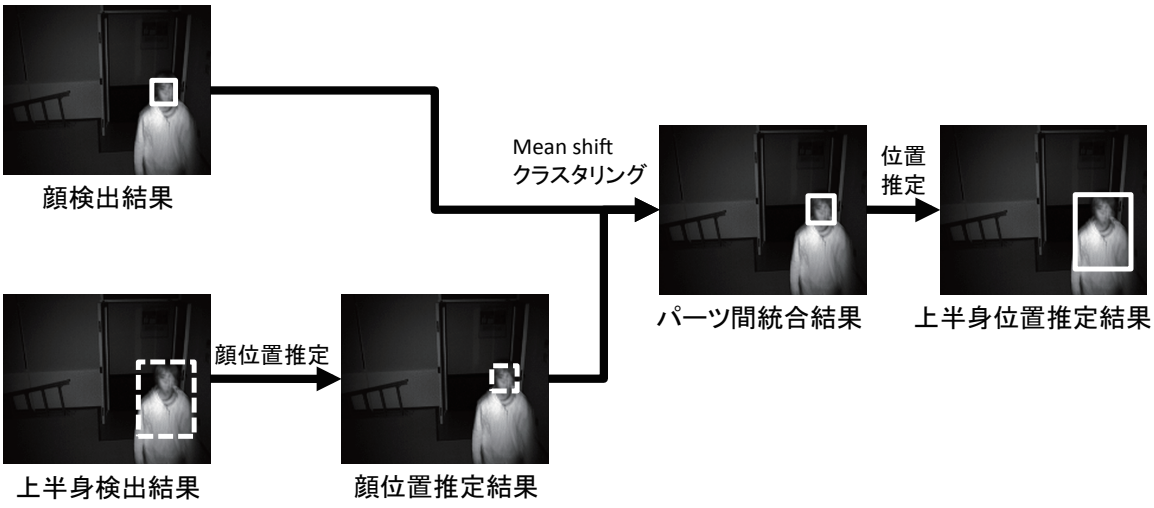


図 3.5: パーツ間の統合の流れ

3.4　実験

3.4.1　実験概要

提案手法の有効性を示すために，パーツベースによる検出手法とパーツ単体での検出手法の比較実験および，検出結果の統合における距離情報の有無による比較実験を行う．まず，各パーツの検出器をあらかじめ学習しておく．各パーツの検出器は，距離情報の有無に関わらず共通である．顔検出の学習は，24× 24 に正規化した正解画像 5000 枚，非正解画像 5000 枚を用いる．カスケード型検出器の各階層において，強識別器は検出率が 99.5%となる数の弱識別器を学習する．各階層に用いる非正解画像は，人が映っていない風景画像 5 万枚から前階層までの強識別器が誤検出した領域を用いる．学習する階層数は 16 である．また，上半身検出の学習も同様に，24× 32 に正規化した正解画像 5000 枚，非正解画像 5000 枚を学習サンプルとして用いる．検出器の各階層の検出率も同様に 99.5%である．

比較実験には，一人歩行シーン，複数人歩行シーン，混雑シーンの 3 種類を評価データとして用いる．歩行シーンは，人が奥から手前へ歩行しており，一人歩行シーンは 6 人が 1 回ずつ歩行する 6 シーケンス，複数人歩行は 6 人が同時に歩行する 3 シーケンスを用いる．混雑シーンは，6 人が立ち止っている 3 シーケンスを評価に用いる．各評価データの全枚数は，一人歩行シーンが 240 フレーム，複数人歩行シーンが 150 フレーム，混雑シーンが 450 フレームである．図 3.6 に評価データの画像例を示す．図 3.6(a)(b)(c) は各シーンの反射強度画像，図 3.6(d)(e)(f) は距離画像である．

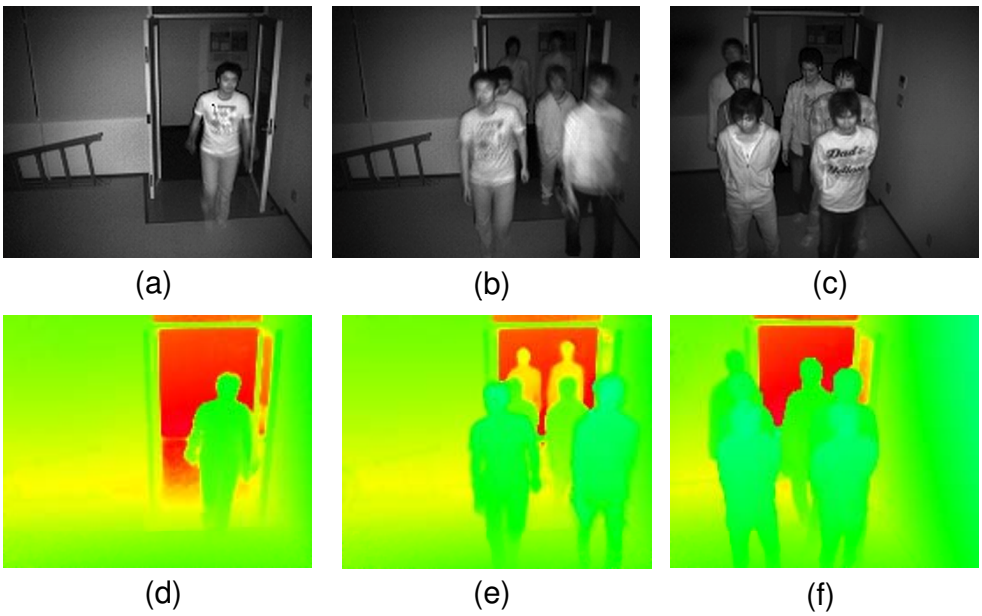


図 3.6: 評価画像例

3.4.2　検出器の構成による性能比較

顔または上半身検出の単体で検出した場合と顔および上半身をパーツとして統合した場合の比較実験結果を図 3.7，図 3.8，図 3.9 に示す．図 3.7 は一人歩行，図 3.8 は複数人歩行，図 3.9 は混雑シーンにおける ROC カーブである．歩行シーンにおいて，上半身検出が顔検出より検出率が高い．これは，歩行シーンで得られる反射強度画像はフレームレートの影響でぼけており，顔が鮮明でないため顔検出の検出率が低くなっているためである．また，画面奥では顔が小さいため検出することができないことも原因の 1 つである．

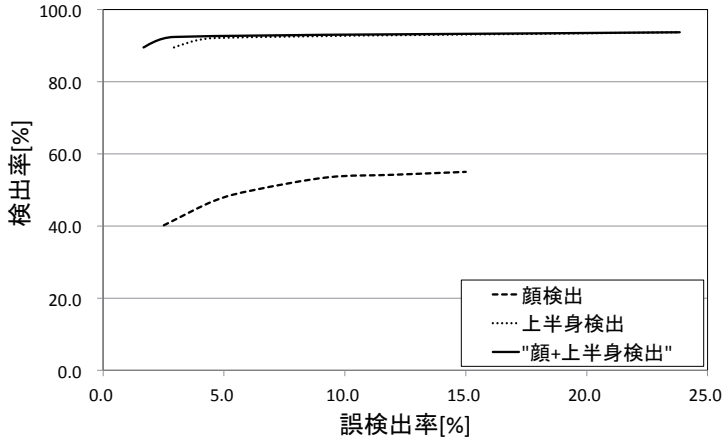


図 3.7: 一人歩行シーンにおけるパーツ検出器ごとの ROC カーブ

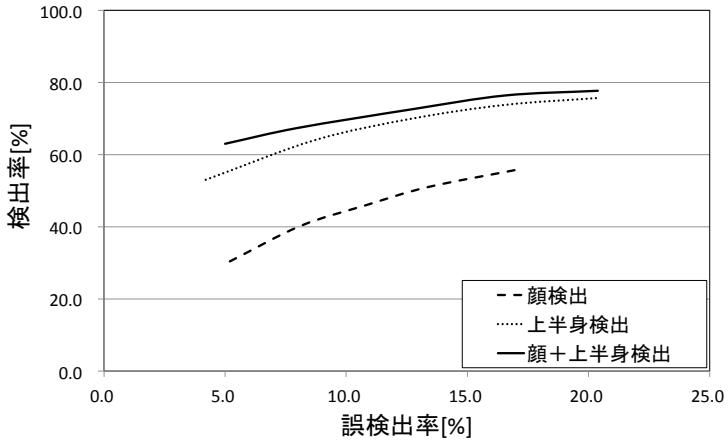


図 3.8: 複数人歩行シーンにおけるパーツ検出器ごとの ROC カーブ

一方，混雑シーンにおいては，図 3.9 よりどちらのパーツがよいか優劣をつけることができない．これは，画面の位置により各パーツのメリットおよびデメリットがあるためである．画面手前では，立ち止まっている顔が鮮明に得ることができるため，顔検出がしやすい．一方，画面手前では画面か

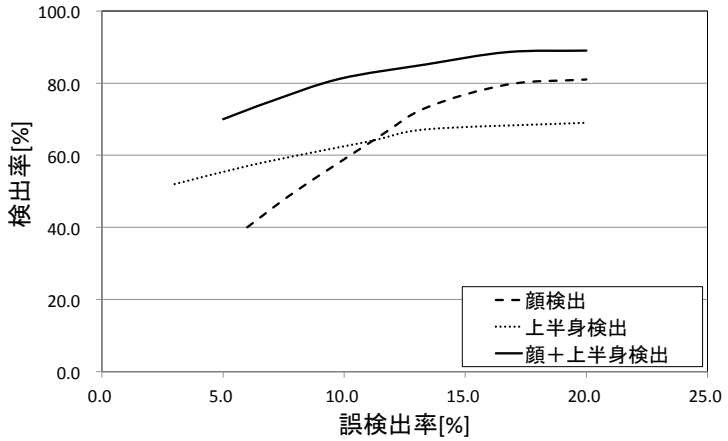


図 3.9: 混雑シーンにおけるパーツ検出器ごとの ROC カーブ

ら上半身がフレーム外へはみ出るため，上半身検出では検出できない場合がある．また，画面奥では，顔が小さいため，顔検出が検出しにくいが，上半身検出が顔が小さくても検出することができる．顔検出と上半身検出の結果を統合することで，各パーツのメリットを組み合わせた検出結果を得ることができる．図 3.9 から顔検出と上半身検出を統合することで，検出率が顕著に向上することが分かる．

検出結果の一例を図 3.10，図 3.11，図 3.12 に示す．図 3.10(a)，図 3.11(a)，図 3.12(a) は顔検出のみの結果，図 3.10(b)，図 3.11(b)，図 3.12(b) は上半身検出のみの結果，図 3.10(c)，図 3.10(c)，図 3.10(c) は顔検出結果と上半身結果を統合した結果を示している．

3.4.3 検出結果統合方法による性能比較

距離情報を用いた場合の検出結果統合と用いない場合の検出結果統合の比較実験結果を図 3.13，図 3.14 に示す．また，Hayashi らや池村らは距離画像から人の検出を行う手法を提案している [37][36]．池村らの距離画像のみを用いた人検出結果を用いた場合の結果についても比較実験結果に示す．ここで，一人歩行の場合，近距離に他の人物が存在しないので，距離情報による影響はない．そのため，本比較実験は複数人歩行と混雑シーンの 2 つのシーンでの結果を示している．

複数人歩行の場合，近距離に他の人物がいるため，距離情報を用いることで誤検出率 10%時に検出率が 1%程度向上している．さらに，混雑シーンでは，複数人歩行よりさらに近距離に人物が存在するケースが多い．そのため，距離情報を用いることで誤検出率 10%時に検出率が 3%程度向上している．これより，検出結果の統合に距離情報を用いることで，検出性能が向上することが分かる．

また，距離画像のみの場合と比較しても性能が向上していることが分かる．距離画像のみの場合，混雑すると距離情報が正確にとれないため，検出率が低下している．

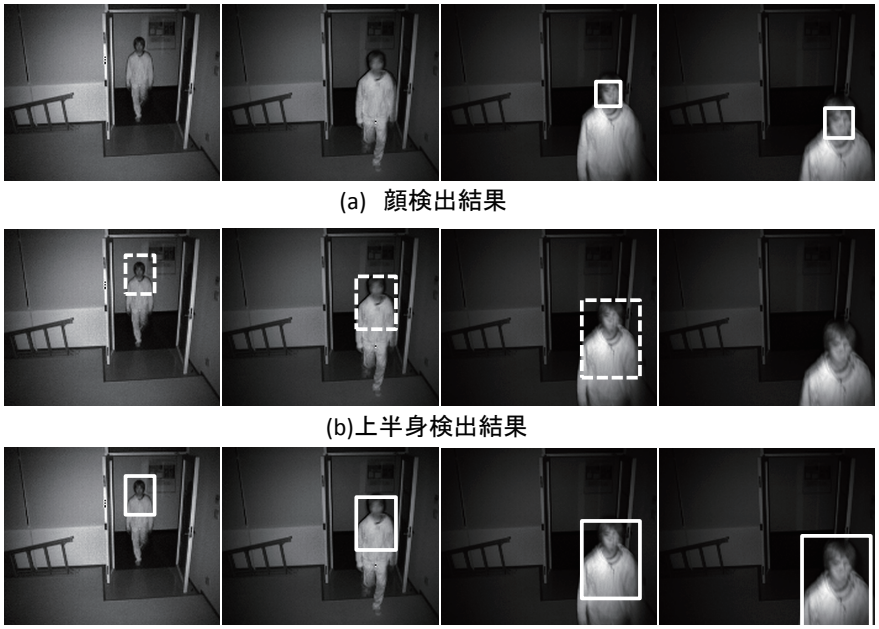


図 3.10: 一人歩行シーンにおける検出結果例

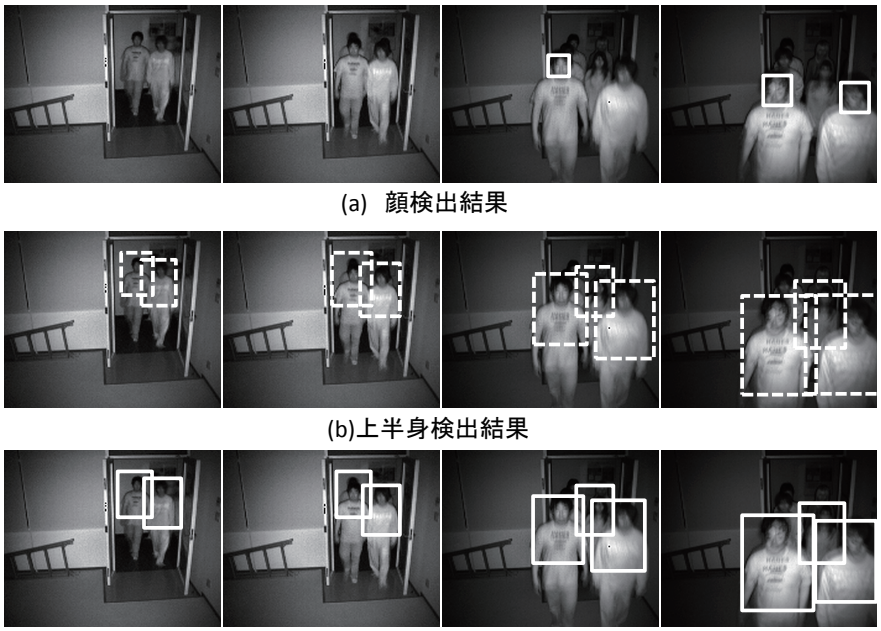


図 3.11: 複数人歩行シーンにおける検出結果例

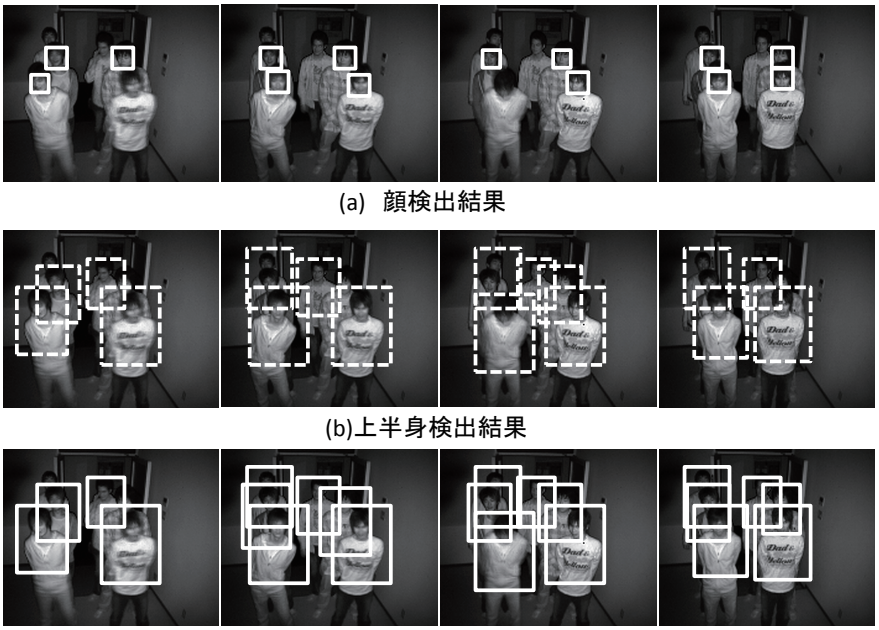


図 3.12: 混雑シーンにおける検出結果例

3.5 考察

従来の Mean Shift クラスタリングはバンド幅内に存在する検出結果を統合する．本研究では，検出率が最も高くなるように，バンド幅を全検出結果の平均幅の 0.25 倍としている．検出率が最も高くなるように設定した場合においても，近距離に他の人物が存在すると統合されてしまうケースがある．図 3.15 に混雑シーンにおいて，距離情報を用いない場合に統合されるケースを示す．図 3.15 の (a) は統合前の各パーツの検出結果，(b) は各パーツごとに統合した結果，(c) は距離情報を用いずにパーツ間の統合を行った結果，(d) は距離情報を用いたパーツの統合結果である．また，顔の検出結果を実線，上半身の検出結果および上半身の検出結果から想定される顔の位置を点線で示している．このシーンでは，手前の人物の上半身検出結果から想定される顔の位置と奥の人物の顔検出結果がバンド幅より小さいため，図 3.15(c) のように 1 つの人物として統合される．一方，統合時に距離情報を用いることで，図 3.15(d) のように，個別に検出することができている．これより，距離情報を考慮して検出結果を統合することで，検出性能の向上を図れることが分かる．

3.6 まとめ

本章では，距離情報を用いた顔および上半身検出結果を統合する方法を提案した．提案手法では，顔と上半身の位置関係を定義して統合するだけでなく，反射強度画像と同時に取得した距離情報を用いて結果を統合している．提案手法により，複数人歩行のシーンにおいて，距離画像を用いない場

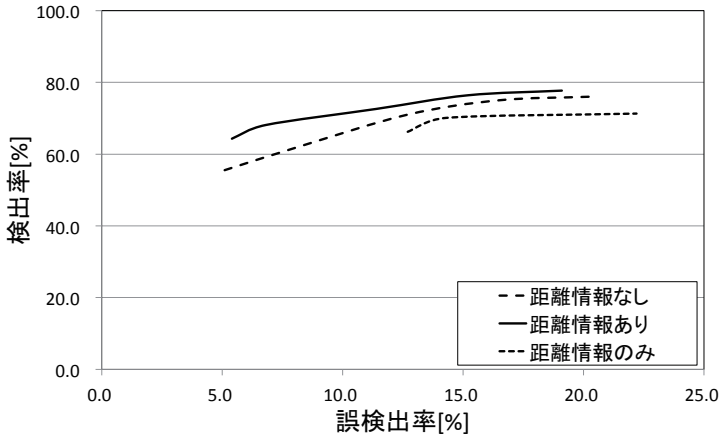


図 3.13: 複数人歩行シーンにおける距離情報の有無による ROC カーブ

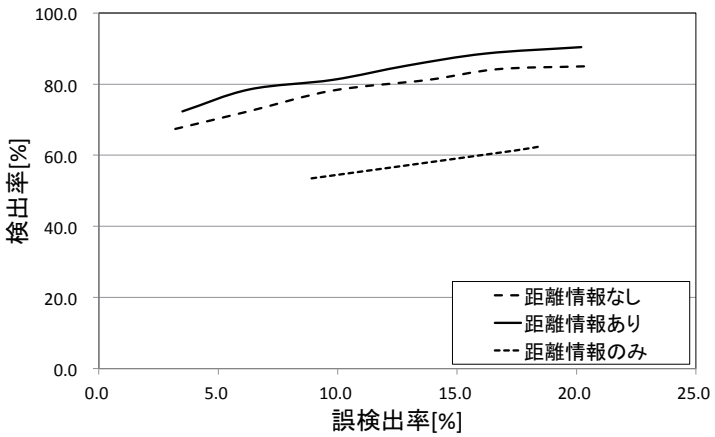


図 3.14: 混雑シーンにおける距離情報の有無による ROC カーブ

合と比較して誤検出率 10%時に検出率が 1%程度向上している．また，複数の人物が立ち止るような混雑シーンでは，誤検出率 10%時に検出率が 3%程度向上している．これより，検出結果の統合に距離情報を用いることで，検出性能を向上させることができている．

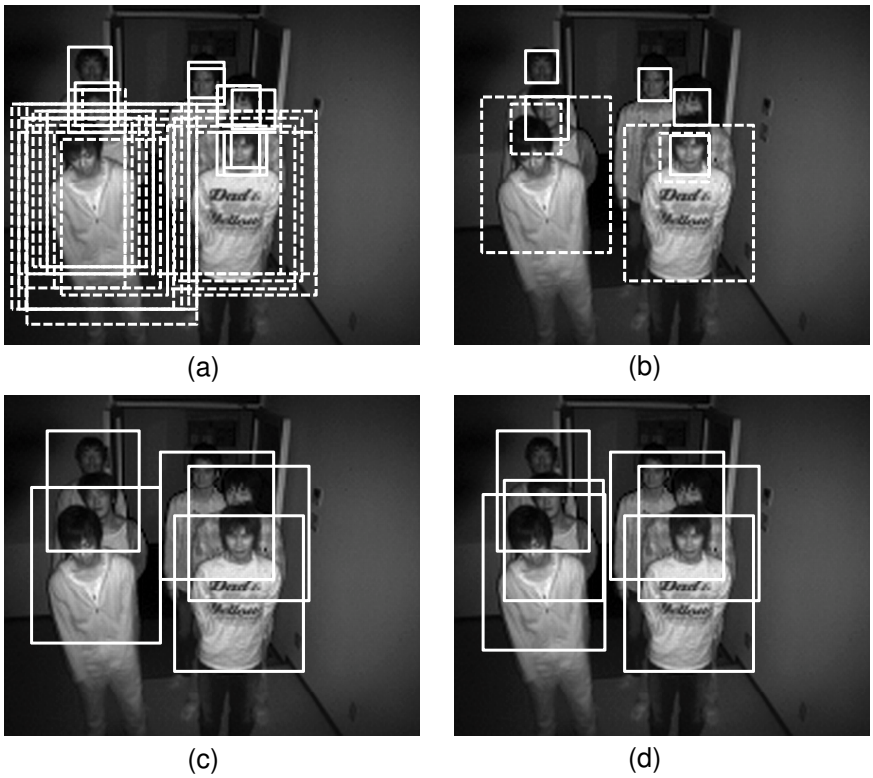


図 3.15: 混雑シーンにおける距離情報の有無による検出結果例

第4章

Online Real Boostingよる物体追跡

ブースティングを用いたオンライン学習は，ある時刻の追跡対象に合わせて検出器を更新するため，追跡対象の大きさや見えの変化，照明変化に対して頑健である．一方で，時々刻々と弱識別器を更新するため，遮蔽が生じた場合に遮蔽物へ乗り移りが生じる．本章では，オンライン学習において，物体らしさを表す確率密度関数を導入した Online Real Boosting を提案する．確率密度関数は，事前にオフラインでの検出器の学習時に導出されており，確率密度関数からあらかじめ想定されている追跡対象と異なる領域を誤って検出しにくくなっている．そのため，異なる物体への乗り移りを軽減させることが可能となる．評価実験では，顔向きが生じるシーンにおいて，頭部を正しく追跡できることを示す．また，人体追跡への応用の可能性についても示す．

4.1 従来の物体追跡技術

物体追跡は，人の行動を理解する上で重要な要素技術であり、実用化が進んでいる。セキュリティ市場においては，不審行動を検知するための監視カメラによる映像監視に用いられている．また，デジタルカメラ市場においては，動物体をきれいに撮影するために自動追尾機能として搭載されている．物体追跡技術は実用化されているが，追跡対象の動きや追跡対象周辺の状況により追跡できない場合がある．さらに，デジタルカメラなどの非固定カメラの場合は，カメラの動きも考慮する必要がある．このように，現状は利用できる環境やシーンが限定されている．そのため，様々な環境やシーンにおいて利用可能な物体追跡が期待されており，現在においても多数の物体追跡の手法が提案されている．

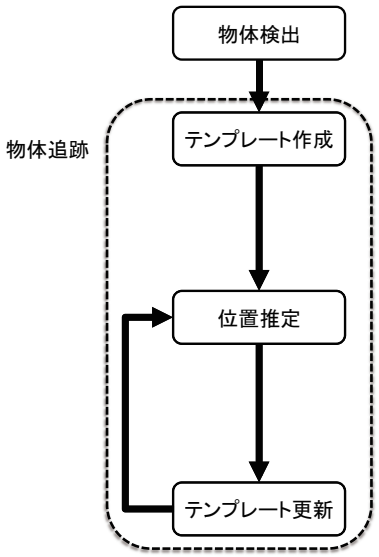


図 4.1: 物体追跡の基本的な流れ

物体追跡の基本的な流れは図 4.1 に示すように，大きく 4 つの処理から構成される．まず，最初に追跡対象の位置を決める．デジタルカメラなどのタッチパネルが搭載されている機器においては，物体の位置を利用者が手動で設定することができる．多くの場合，物体検出手法を用いて自動的に追跡対象を設定するのが一般的である．

次に，追跡対象のテンプレートを生成する．テンプレートはカラーヒストグラムなどの特徴量を算出して作成する．そして，生成されたテンプレートを用いて，次フレーム以降の画像に対して，テンプレートと類似する物体の位置を推定する．推定した位置から得られた追跡対象物体の情報を利用して，テンプレートを更新する．このような流れで物体を追跡することができるが，追跡対象の向きや姿勢変化，照明変化や隠れ，さらに高速化などに対応するために，様々な改良を加えた手法が多く提案されている。Mean Shift 法は追跡物体の位置を高速に推定する方法として提案されており，パーティクルフィルタは，追跡位置の事前予測に従った位置推定方法を提案している。ここでは，代表的な手法として，Mean Shift 法 [38] とパーティクルフィルタ [39] について述べる．

4.1.1 Mean Shift による物体追跡

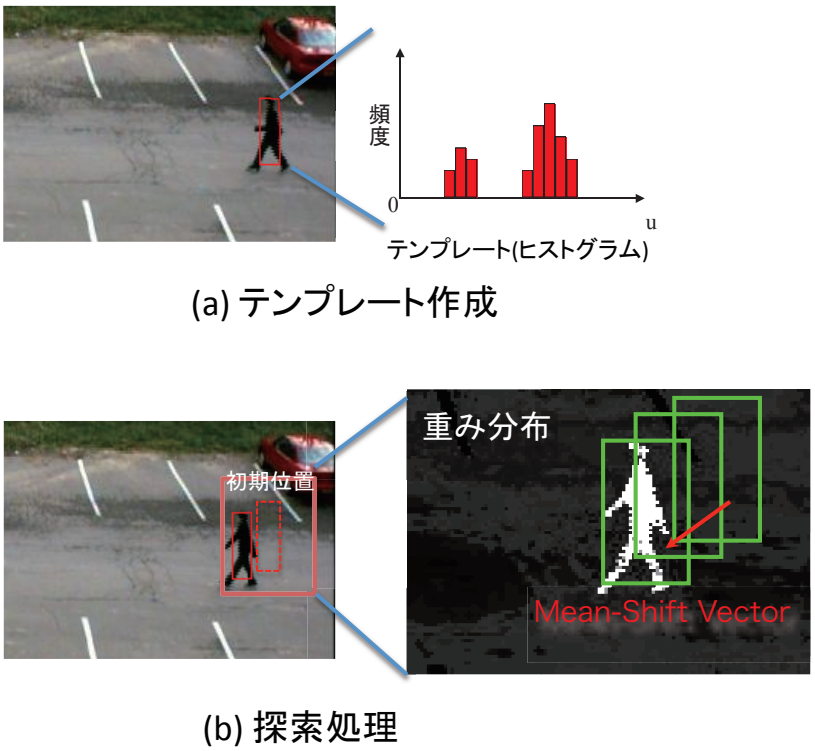


図 4.2: Mean Shift による追跡

Mean Shift は，局所領域内においてある重み関数 $f(x)$ の値が大きくなる方向に中心位置を移動させる勾配法の一つであり，前節で述べたクラスタリングだけでなく，物体追跡にも応用されている．図 4.2 に示すように，追跡対象のテンプレートをヒストグラムとして作成し (図 4.2(a))，そのヒストグラムをもとに重み関数を更新する．重みが大きくなる方向に探索位置をずらしながら，移動量が一定量より小さくなるまで繰り返し行う (図 4.2(b))．これにより，局所領域内での $f(x)$ が極大となる点を求めることができる．

Mean Shift による物体追跡の流れを図 4.3 に示す．Mean Shift による物体追跡の場合，追跡対象のカラーヒストグラムを用いることが多い．これにより，様々な形状の物体を追跡することが可能であり，部分的な隠れやカメラの位置変化に対して頑健であるという特徴がある．ここでは，カラーヒストグラムをベースとした特徴量を用いることを前提として述べる．

まず，検出された対象物体の領域内におけるカラーヒストグラムを作成する．カラーヒストグラ

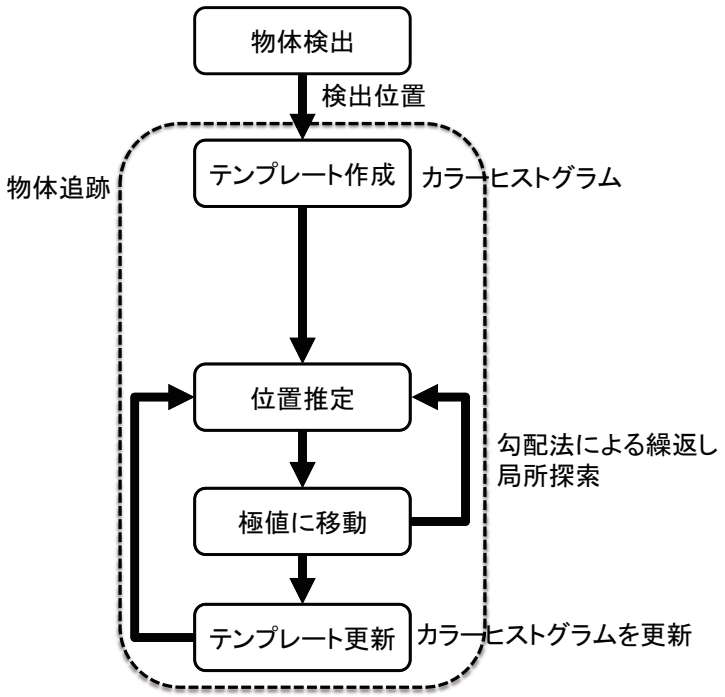


図 4.3: Mean Shift による物体追跡の流れ

ムは次式のように，色成分ごとに正規化して作成する．

$$m_u^R = \frac{\sum_{i=1}^{n} \delta_K[b_R(x_i^*) - u]}{n} \tag{4.1}$$

$$m_u^G = \frac{\sum_{i=1}^{n} \delta_K[b_B(x_i^*) - u]}{n} \tag{4.2}$$

$$m_u^B = \frac{\sum_{i=1}^{n} \delta_K[b_G(x_i^*) - u]}{n} \tag{4.3}$$

ここで，m_u^R，m_u^G，m_u^B は各色成分のヒストグラムの要素 u における頻度，δ_K は Kronecker のデルタ関数，$b_R(x_i^*)$，$b_G(x_i^*)$，$b_B(x_i^*)$ は対象物体領域の位置 x_i における各色成分を示している．このカラーヒストグラムをテンプレートとして保持しておく．

次に，現フレームの画像に対し，対象物体のカラーヒストグラムを用いて位置を推定する．現フレームでの対象物体の位置推定は，前フレームにおける位置を初期位置として，移動量を算出する．そのために，初期位置におけるカラーヒストグラムを作成する．各色成分のカラーヒストグラムを d_u^R，d_u^G，d_u^B とする．テンプレートとして保持しているカラーヒストグラムと現フレームでの初期位置におけるカラーヒストグラムから位置 i の各色成分に対する重み ω を次式のように算出する．

$$\omega_i^R = \sqrt{m^R_{b_{R(x_i)}} / d^R_{b_{R(x_i)}}(x)} \tag{4.4}$$

$$\omega_i^G = \sqrt{m^G_{b_{G(x_i)}} / d^G_{b_{G(x_i)}}(x)} \tag{4.5}$$

$$\omega_i^B = \sqrt{m^B_{b_{B(x_i)}} / d^B_{b_{B(x_i)}}(x)} \tag{4.6}$$

位置 i における重みは式 (4.7) のように加算する．

$$\omega_i = \omega_i^R + \omega_i^G + \omega_i^B \tag{4.7}$$

そして，式 (4.8) のように Mean Shift Vector を求める．

$$\Delta x = \frac{\sum_{i=1}^{n} \omega_i (x_i - x)}{\sum_{i=1}^{n} \omega_i} \tag{4.8}$$

得られた Mean Shift Vector をもとに対象物体の位置を移動させ，移動後の位置に対しても同様に Mean Shift Vector を求める．移動量が一定以上小さくなるか，予め設定している繰り返し回数まで移動を繰り返すことで，極大値に達することができる．

4.1.2 パーティクルフィルタによる物体追跡

パーティクルフィルタは，時系列フィルタリングの一種であり，過去に観測された信号系列から未来の値を予測したり, ノイズを含む観測値から対象の状態を推定する問題などに有効な理論である．また、非線形や非ガウス型の状態空間モデルに対して, 効率よく状態を推定することもできる. 図 4.4 に示すように，パーティクルフィルタはテンプレートを作成し (図 4.4(a))，テンプレートと類似する位置を探索する．その際，網羅的に探索するのではなく，物体が存在する確率の高い位置に対してのみ探索処理を行う (図 4.4(b))．

パーティクルフィルタは，時刻 t における事後確率密度 $p(x_t|z_t)$ を，N 個のパーティクルを用いて近似する．各パーティクルは式 (4.9) のように，状態 x と尤度 π からなる．

$$s_t^{(i)} = \{x_t^{(i)}, \pi_t^{(i)}\} \tag{4.9}$$

時刻 t において，画像から観測値 z_t が得られると，追跡対象の状態 x_t を確率変数とする確率密度は，事後確率密度 $p(x_t|z_t)$ として表される．ベイズの定理から，事後確率密度は

$$p(x_t|z_t) = p(z_t|x_t)p(x_t|z_{t-1}) \tag{4.10}$$

となり，現時刻 t において得られた観測値 z_t と時刻 $t-1$ の観測値から予測した状態 x_t から求めることができる．パーティクルフィルタによる物体追跡の流れを図 4.5 に示す．

物体検出により追跡対象が得られた後，その位置を状態ベクトルとして N 個のパーティクルを生成する．また，追跡対象の領域からテンプレートを作成する．ここでは，カラーヒストグラムを特徴量として用いる．

次に，状態遷移モデルに従ってパーティクルを遷移させて位置を予測する．状態遷移モデルは，パーティクルをどのように遷移させるかを決めるためのモデルであり，物体の運動をあらかじめ考慮する

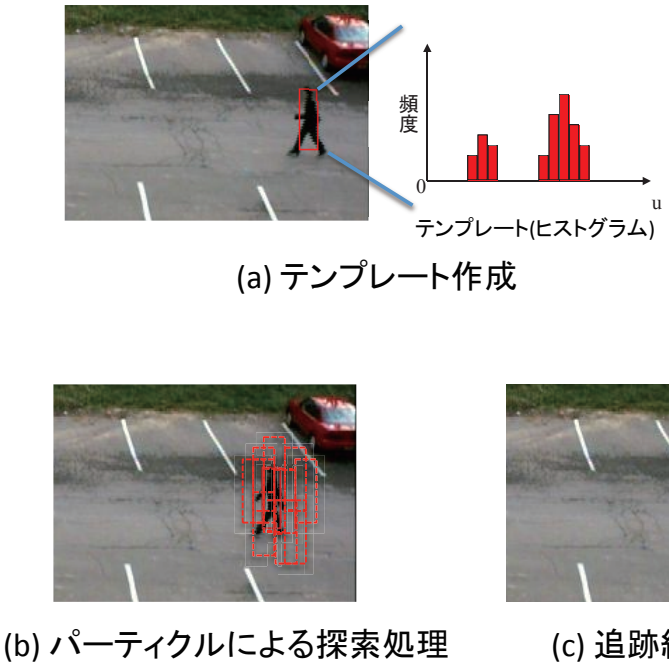


図 4.4: Particle Filter による物体追跡の流れ

ことで効率的なパーティクルの配置が可能である．物体の運動が未知な場合や動きが複雑な場合，状態遷移モデルとしてランダムウォークを用いる．ランダムウォークは前の状態ベクトルに対して，正規乱数を用いてパーティクルを遷移させる．パーティクルはガウシアンノイズ ω_t を用いて式 (4.11) のように遷移する．

$$x_t^{(i)} = x_{t-1}^{(i)} + \omega_t \tag{4.11}$$

また，状態遷移モデルは，等速運動を仮定した線形予測モデルなどもある．

各パーティクルが遷移した位置において，特徴量を観測値として算出し，尤度を計算する．尤度 $L_t^{(i)}$ は下記のように，テンプレート q_u と観測値 p_u のバチャタリア係数により類似度 S を求め，類似度から算出する．

$$S = \sum_{u=1}^{m} \sqrt{p_u q_u} \tag{4.12}$$

$$L_t^{(i)} = exp(kS) \tag{4.13}$$

ここで，k は係数である．

各パーティクルに対する尤度を求めた後，状態ベクトルと尤度から現時刻での位置を推定する．位

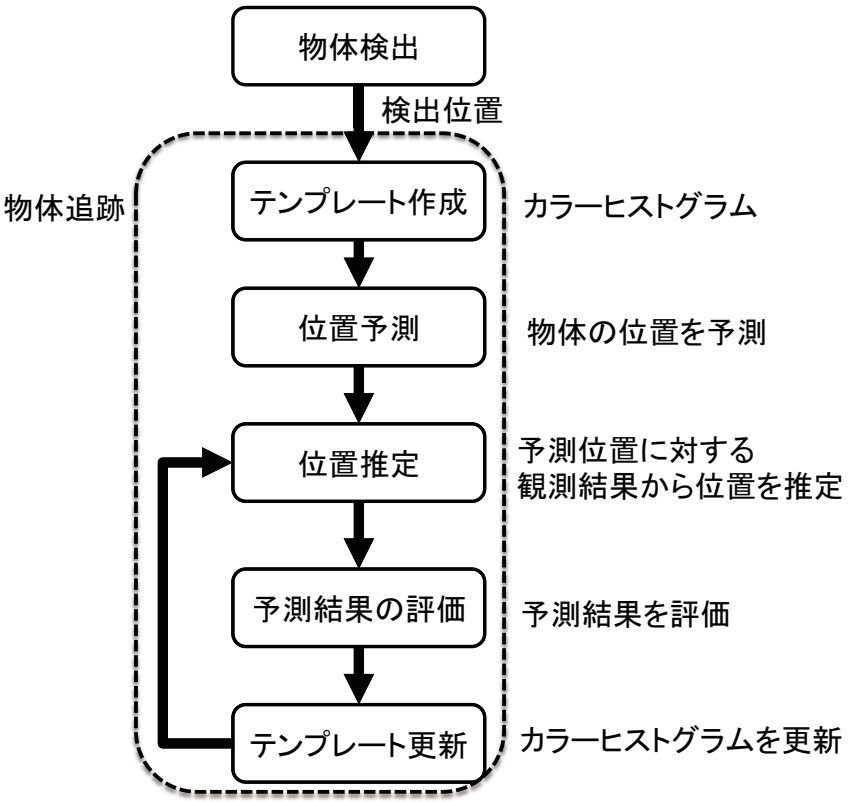


図 4.5: パーティクルフィルタによる物体追跡の流れ

置 X_t は，式 (4.14) のように重み付き平均により算出する．

$$X_t = \sum_{i=1}^{N} \pi_t^{(i)} x_t^{(i)} \tag{4.14}$$

ここで，$\pi_t^{(i)}$ は各パーティクルの尤度の合計値が 1 になるように正規化後の尤度である．

そして，次の時刻に継承するパーティクルを尤度をもとに決める．尤度の低いパーティクルは消滅させるだけでなく，一定の数を確保するためにパーティクルの生成も行う．多くの場合，追跡対象から離れたパーティクルは消滅し，追跡対象付近に新たなパーティクルが生成される．

また，尤度をもとに推定された現時刻での位置に対して，テンプレートを作成し，次の時刻のテンプレートして用いる．これらの処理を毎時刻繰り返し行うことで，物体の追跡を実現する．

パーティクルフィルタは，事後確率分布を多数のパーティクル (サンプル) を用いて近似する手法である．多数のパーティクルにより，状態空間内全体での確率分布を近似することで，ノイズや環境の変動に対して頑健な追跡を行うことができる．

4.1.3　オンライン学習による物体追跡

Mean Shift およびパーティクルフィルタは検出された追跡対象領域からカラーヒストグラムなどの特徴量を算出し，その類似度をもとに現時刻での位置を推定することで，物体追跡を実現している．しかしながら，カラーヒストグラムを特徴量として用いているため，類似色の物体や背景に乗り移りやすいという課題がある．また，照明変動が生じるシーンや複雑背景下では正しく追跡することができないため，特定の環境に特化して用いることが多い．

これらの問題を解決する方法として，物体検出に用いる検出器を追跡に応用する手法が注目されている [8][40]．その中でも Oza ら [40] が提案した Online Boosting は，物体検出に用いられるブースティングをオンライン学習に応用している．ブースティングは，アンサンブル学習手法の 1 つであり，識別能力の低い弱識別器を多数組み合わせることで高精度の強識別器を検出器として学習することができる．物体追跡では，追跡対象に適した検出器を逐次オンラインで学習し，類似する物体を検出することで追跡を行っている．図 4.6 にオフラインおよびオンラインによる学習の流れを示す．

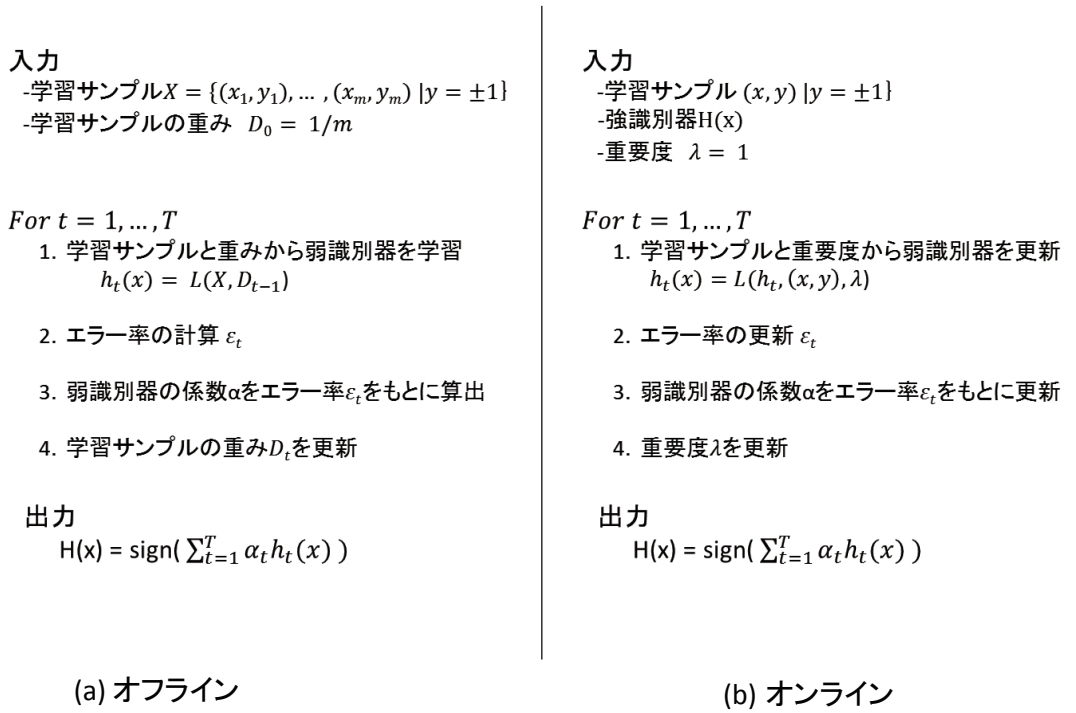


図 4.6: オフラインおよびオンライン時のブーティング学習の流れ

オフライン学習の場合，学習サンプル画像とその重みをともに識別エラーが最小となる弱識別器を選択する．

$$h_t = \arg\min_{h_j \in H} \epsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)] \tag{4.15}$$

そして，選択された弱器別器を評価する係数 α_t は, 識別エラーから算出される．

$$\alpha_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t} \tag{4.16}$$

学習サンプルの重みは学習した弱識別器 h_t に対する応答値から更新する．

$$D_{t+1} = \frac{D_t exp(-\alpha_t y_i h_t(x_i)}{Z_t} \tag{4.17}$$

これらを T 回繰り返すことで，T 個の弱識別器を得ることができる。

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{4.18}$$

一方，オンライン学習の場合，入力される学習サンプルはある時刻における 1 枚の学習サンプルのみである．Online Boosting では，全ての弱識別器に対して，学習サンプルに対するエラー率をもとに重み α を更新する．1 枚の学習サンプルは，重みを変えて k 回に複製される．重みを変える回数は，ポアソン分布に従う．ポアソン分布は，

$$P(K=k) = \frac{exp^{-\lambda}\lambda^k}{k!} \tag{4.19}$$

となり，重要度 λ をもとに繰り返し回数 k を決定する．弱識別器 h_t において，学習サンプルが正しく識別できた場合，重要度およびエラー率を

$$\lambda_m^{sc} \leftarrow \lambda_m^{sc} + \lambda \tag{4.20}$$

$$\epsilon_m \leftarrow \frac{\lambda_m^{sw}}{\lambda_m^{sc} + \lambda_m^{sw}} \tag{4.21}$$

$$\lambda \leftarrow \lambda(\frac{1}{2(1-\epsilon_m)}) \tag{4.22}$$

のように算出する．

一方，学習サンプルを正しく識別できなかった場合は，

$$\lambda_m^{sw} \leftarrow \lambda_m^{sw} + \lambda \tag{4.23}$$

$$\epsilon_m \leftarrow \frac{\lambda_m^{sw}}{\lambda_m^{sc} + \lambda_m^{sw}} \tag{4.24}$$

$$\lambda \leftarrow \lambda(\frac{1}{2\epsilon_m}) \tag{4.25}$$

となる．

弱識別器 h_{m+1} は，弱識別器 h_m の更新時に得られた λ をもとに，繰り返し回数 k が決まる．

識別器 h_m が正しく識別された場合は，λ が小さくなり，繰り返し回数 k が小さくなる．一方，誤った識別がされた場合は，繰り返し回数が大きくなる．

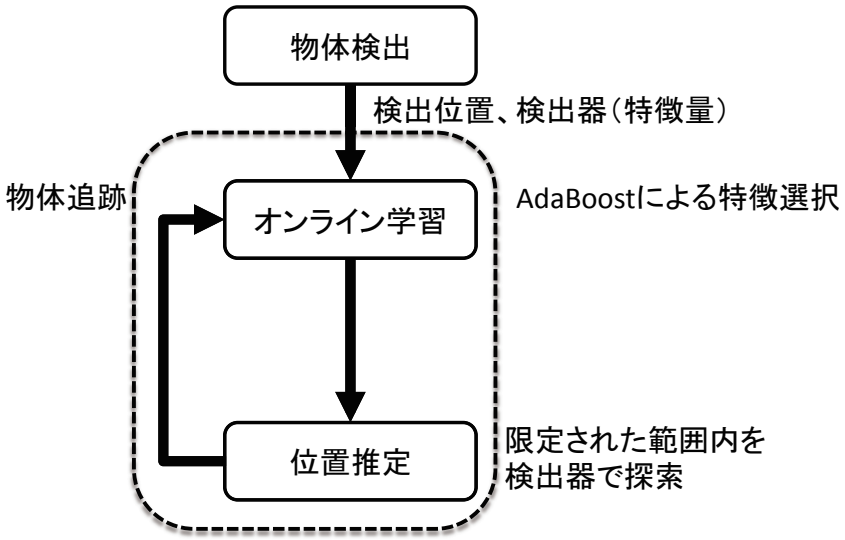


図 4.7: オンライン学習による物体追跡の流れ

オンライン学習による物体追跡の流れを図 4.7 に示す．

Mean Shift とパーティクルフィルタは，物体検出の手法に関係なく物体追跡のためのテンプレートをカラーヒストグラムなどを用いて生成している．これらの手法との大きな違いは，あらかじめ学習している検出器により検出された物体を入力として，この物体に適した検出器を新たに学習する点である．オンラインでの検出器の学習は，上述した Online Boosting を用いて，弱識別器の重み α_i を更新する．

Online Boosting は全ての弱識別器の重みを更新する．一方で，Grabner ら [9] はオンライン学習時に追跡に有効な弱識別器の選択を行う．これにより，物体の見え方や隠れが生じるようなシーンにおける物体追跡を実現している．

オンライン学習は，物体検出に用いる検出器を追跡に応用し，物体追跡を実現している．そのため，追跡物体と類似色の物体や背景の乗り移りを軽減することが出来ている．しかし，Online Boosting の計算コストは弱識別器の数に大きく依存しており，弱識別器を多くすると追跡性能を高くすることができるが，重みの更新に時間がかかってしまう．一方，計算コストを削減するために，弱識別器数を少なくすると，性能が低下する．そのため，いかに識別能力の高い弱識別器を選別し，少ない弱識別器で十分な性能を達成させるかが重要である．また，物体検出に用いる検出器を応用するため，あらかじめ学習していない姿勢や形状に変形した場合，追跡が失敗することがある．そこで，本研究では，少ない弱識別器数で十分な性能を達成させるためのオンライン学習手法について提案する．

4.2　提案手法

4.2.1　提案手法のフレームワーク

従来の Online Boosting の課題である高い識別能力の弱識別器を選択する方法を，本研究では弱識別器の特徴量タイプと弱識別器の学習手法の 2 点から検討する．顔向きなど物体の形状が変わる場合，変化によって追跡に有効な特徴量は異なると考えられる．例えば，正面を向いている頭部を追跡する場合，顔の濃淡パターンを捉える Haar-like 特徴量のような特徴量が有効である．一方で，横向きなど顔の領域が少ない場合は，頭部の輪郭を捉えるエッジを利用した特徴量が有効である．提案手法では向き変化に応じて最適な特徴量のタイプをオンラインで選択できるように，複数の特徴量タイプを用いて，オフラインで学習を行う．

次に，従来の Online Boosting は AdaBoost により弱識別器を選択している．AdaBoost で選択された弱識別器はしきい値により 0 または 1 の 2 値のみを出力するため，識別結果の確からしさを得ることができない．そこで，提案手法では実数による確からしさを出力することで，高い識別能力を持つ弱識別器を学習できる Real AdaBoost を用いる．

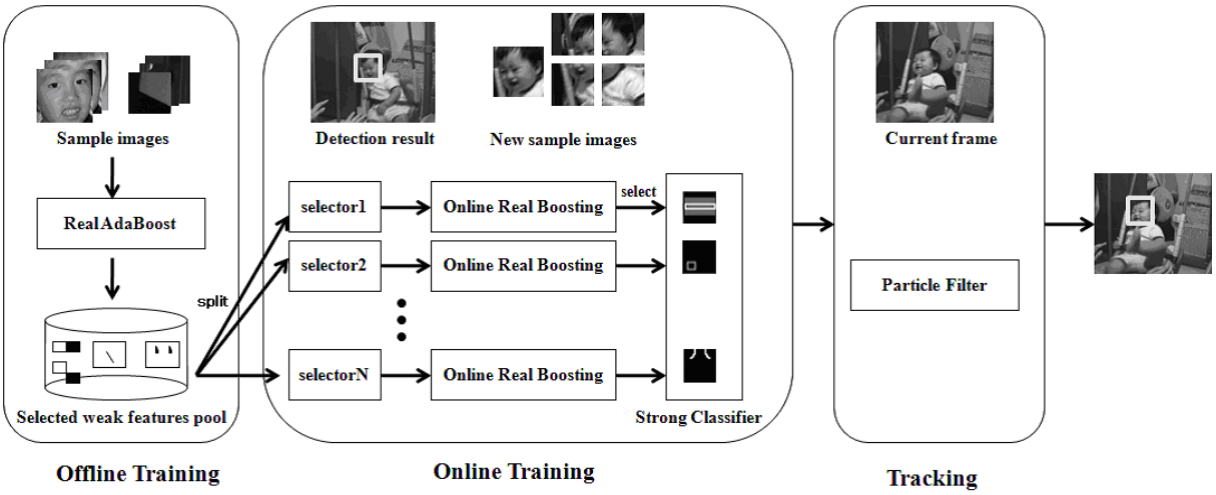


図 4.8: 提案手法のフレームワーク

本研究で提案する Real AdaBoost を用いた Online Real Boosting のフレームワークを図 4.8 に示す．提案手法では追跡したい物体である顔または人体をあらかじめオフラインで学習し，オンライン学習時に選択するための弱識別器群を用意しておく．オフライン学習には Real AdaBoost を用いる．オンライン学習は検出器により検出した位置をもとに学習サンプルを作成する．次に，あらかじめ学習して得られた弱識別器を複数のセレクタに分割し，各セレクタの中からサンプル画像を最も正しく判別できる弱識別器を Online Real Boosting により 1 つ選択する．選択された各セレクタの弱識別器を組み合わせて強識別器を作成する．次の時刻に入力された画像からオンラインで学習

された強識別器を用い，パーティクルフィルタにより追跡物体の位置を検出する

4.3 追跡に用いる画像局所特徴量

物体を追跡する上で，物体の形状や見えの変化を考慮する必要がある．顔を追跡する場合，表情などの輪郭変化に対しては輝度に着目する特徴量が有効である．一方，顔向きなどの形状変化に対してはエッジに着目する特徴量が有効である．そこで，顔を追跡する場合に有効な特徴量について検討する．輝度に着目する特徴量としては，Haar-like 特徴量および ABS Haar-like 特徴量，エッジに着目する特徴量には，Edgelet 特徴量と EOH 特徴量を用いる．ABS Haar-like 特徴量は 2 つの領域の差の絶対値を特徴量とし，顔向きや体の向きなどの急激な向き変化に強い特徴量である．

これら 4 つの特徴量について，すべての特徴量を用いた場合を基準とし，各特徴量を除いた場合の検出率の低下を比較した．まず，すべての特徴量を弱識別器の候補として Real Adaboost により 300 個の弱識別器を選び出し，強識別器とする．次に，Haar-like 特徴量を除いた残りの特徴量について，同様に 300 個の弱識別器を選び出し，強識別器を作成する．以下同様に，ABS Haar-like 特徴量，Edgelet 特徴量，EOH 特徴量をそれぞれ除いた場合の強識別器を作成し，検出率を比較した．評価データとして CMU+MIT データベースを用いた．全ての特徴量による検出率 (All feature) と各特徴量を除いた場合の検出率 (exc. $\cdots$) を図 4.9 に示す．図 4.9 より EOH を除いた場合，最も検出率が低下していることがわかる．これは頭部や輪郭のエッジが特徴量として非常に有効であるからと考えられる．一方，Egdelet 特徴量は検出率の低下が少なく，他の特徴量タイプに比べて検出性能に対する影響度が低い．これは，Edgelet 特徴量は EOH 特徴量に比べて，形状の変化に対するロバスト性が低いためと考えられる．Edgelet 特徴量はパターン内の要素ごとにエッジ情報を比較し，判別するのに対して，EOH 特徴量は領域全体のエッジ情報を比較する．そのため，EOH 特徴量はある程度の変化量であれば許容でき，正しく識別できると考えられる．このように，同じエッジ情報を用いた特徴量タイプにも関わらず，大きな違いが生じている．また，ABS Haar-like 特徴量を除いた場合も検出率が大きく低下している．ABS Haar-like 特徴量は領域間の差の絶対値を特徴量としており，顔の向きが大きく異なる場合でも符号が反転することがないため，正しく識別できていると考えられる．これより，ABS Haar-like 特徴量は向き変化に有効な特徴量であるといえる．

4.4 Online Real Boosting

これまで述べた特徴量を学習するための Online Real Boosting について述べる．提案手法の処理手順を図 4.10 に示す．Online Real Boosting は，オンライン学習のための準備，弱識別器の選択，弱識別器の更新の 3 つのステップからなる．これらについて詳しく説明する．

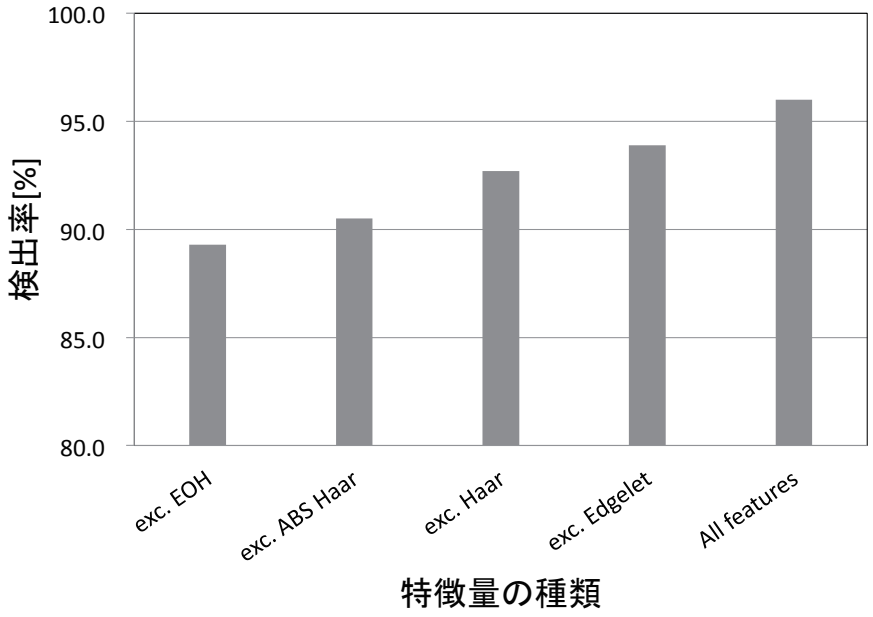


図 4.9: 各特徴量を組み合わせた場合の顔検出の性能

4.4.1 オンライン学習のための準備

はじめに，あらかじめ追跡したい物体である頭部の画像を用いてオフライン学習を行い，強識別器 $H_{offline}(x)$ を用意しておく．Grabner らはオフラインで得られた弱識別器セットの中から弱識別器の選択と，学習サンプルの重みの更新をセレクタの数だけ繰返し行う．全セレクタが同一の弱識別器セットから選択する場合，サンプルによりセレクタ間で同一の弱識別器が選ばれる可能性がある．提案手法は Grabner らと異なり，図 4.10 の手順 2 において，弱識別器セットをセレクタ数と同じ N 個のグループに分け，各セレクタは対応するグループから 1 つの弱識別器を選択する．各グループに含まれる弱識別器セットが異なるようにしており，同一の弱識別器が異なるセレクタで選択されることはなく，強識別器の識別能力を高めることができる．また，学習サンプルの重みの更新を行う必要もない．

学習サンプルは図 4.8 に示すように，正解サンプルに前フレームの頭部領域，非正解サンプルに頭部周囲の背景領域とし，合計 L 個のサンプルを用いる．非正解サンプルの背景領域は頭部位置を一部含むような範囲を設定することで，頭部追跡の追跡位置のずれを軽減することができる．図 4.10 の手順 3 において，各セレクタの識別能力を示す評価値 λ_n を式 (4.26) のように初期化する．

0. オフラインで強識別器を学習

強識別器 $H_{offline}(x)$(弱識別器 M 個)

弱識別器 $h_{offline,m}(x) m = 1, ..., M$

1. 新しいサンプルデータ $(x_1, y_1), .., (x_L, y_L), y_i = \pm 1$, L はサンプル数

2. 弱識別器 M 個をセレクタ数と同じ N 個に分割

3. セレクタの評価値 λ_n の初期化

$$\lambda_n = 1/N \tag{4.26}$$

弱識別器の識別エラー ϵ の初期化 $\epsilon = 0$

4.For n=1,2,..,N //セレクタ数

4.1.For m=1,2,..,M/N //弱識別器数

4.1.1. 弱識別器の識別関数を更新

If $h_{n,m}(x) \in bin_j$

$$W^y_{n,m,j} = W^y_{n,m,j} + \lambda_n \tag{4.27}$$

4.1.2. 弱識別器を更新

If $h_{n,m}(x) \in bin_j$

$$h_{n,m}(x) = \frac{1}{2} \ln \frac{G(W^+_{n,m,j}) + \beta}{G(W^-_{n,m,j}) + \beta} \tag{4.28}$$

G はガウシアン平滑化による正規化関数

4.1.3. 弱識別器選択のための評価関数

$$Z_{n,m} = 2 \sum_{j=1}^{T} \sqrt{W^+_{n,m,j} W^-_{n,m,j}} \tag{4.29}$$

T はヒストグラムの要素数

4.2. 最も Z が小さい弱識別器を選択

$$\begin{aligned} m^* &= \arg\min_m Z_{n,m} \\ h_n &= h_{n,m^*} \end{aligned} \tag{4.30}$$

4.3. 弱識別器 $h_m(x)$ の識別エラー ϵ_n を算出

If $y \cdot h_n(x) \geq 0$

$$\epsilon_n^{+1} = \epsilon_n^{+1} + \lambda_n \cdot |h_n| \tag{4.31}$$

Else

$$\epsilon_n^{-1} = \epsilon_n^{-1} + \lambda_n \cdot |h_n| \tag{4.32}$$

$$\epsilon_n = \frac{\epsilon_n^{+1}}{\epsilon_n^{+1} + \epsilon_n^{-1}} \tag{4.33}$$

4.4. 弱識別器の重みを設定

$$\alpha_n = \frac{1}{2} \log \frac{1 - \epsilon_n}{\epsilon_n} \tag{4.34}$$

4.5 セレクタの評価値を更新

If $y \cdot h_n(x) \geq 0$

$$\lambda_n = \frac{\lambda_n}{2(1 - \epsilon_n)} \tag{4.35}$$

Else

$$\lambda_n = \frac{\lambda_n}{2\epsilon_n} \tag{4.36}$$

5. 強識別器

$$H(x) = sign(conf(x)) \tag{4.37}$$

$$conf(x) = \sum_{n=1}^{n} \alpha_n h_n(x) \tag{4.38}$$

図 4.10: Online Real Boosting の処理手順

4.4.2 弱識別器の選択

次に，図 4.10 の手順 4 においてセレクタごとに最適な弱識別器を 1 つ選択する．図 4.10 の手順 4.1.1 では式 (4.27) のように，各弱識別器の確率密度関数 $W^y_{n,m,j}$ にセレクタの評価値 λ_n を重みとして加算して更新する．確率密度関数 $W^y_{n,m,j}$ はオフライン学習時に得られた頭部および背景の確率分布であり，正解サンプルは $W^{+1}_{n,m,j}$，非正解サンプルは $W^{-1}_{n,m,j}$ を更新する．確率密度関数 $W^y_{n,m,j}$ はヒストグラムで実現されており，ヒストグラムの要素位置 bin_j は特徴量の値から算出する．全サンプルデータによる確率密度関数の更新後，式 (4.28) のように各弱識別器を更新する．弱識別器は確率密度関数 $W^y_{n,m,j}$ と同様にヒストグラムで実現されている．サンプルデータにより更新される確率密度関数の要素が 1 つであると，要素間での確率分布の変動が大きくなる．そのため，1 つのサンプルデータが属する要素は，その前後の要素位置の確率分布を用いて弱識別器を更新するガウシアン平滑化関数を用いる．ここで，β は 0 による除算を防ぐための係数である．

弱識別器の選択は図 4.10 の手順 4.1.3 の式 (4.29) のようにバチャタリア距離 $Z_{n,m}$ に基づく評価関数を用い，手順 4.2 ですべての弱識別器から $Z_{n,m}$ が最も小さな弱識別器をセレクタの代表として選ぶ．

4.4.3 弱識別器の更新

選ばれた弱識別器の識別能力を評価するために，図 4.10 の手順 4.3 でオンライン学習に用いたサンプルデータにより，エラー率を算出する．正しく識別できた場合，$y \cdot h_n(x)$ は正の値となり，式 (4.31) のように正解累積値 ϵ^{+1}_n に加算する．また，負の場合は，誤識別として式 (4.32) のように非正解累積値 ϵ^{-1}_n に加算する．式 (4.31)(4.32) において，Grabner らは λ_n を加算しているのに対して，提案手法では $\lambda_n \cdot |h_n(x)|$ を加算している．これは，Real AdaBoost により学習した弱識別器が出力する実数の確からしさを用いることで，より正確な識別エラーを式 (4.33) で算出できるためである．式 (4.33) で得られた識別エラー ϵ_n により，弱識別器の有効性を示すための値として α_n を式 (4.34) のように求める．Real AdaBoost は各弱識別器が識別の信頼度を出力するが，提案手法では，セレクタで選択される弱識別器の有効性を示す α_n を別に算出し，式 (4.38) において乗じるようにしている．

セレクタの評価値 λ_n は図 4.10 の手順 4.5 で識別エラー ϵ_n の値を基に，式 (4.35) および (4.36) のように更新する．評価値 λ_n はセレクタ n の識別能力を示す値である．λ_n が大きい場合，セレクタ n は追跡対象に対して有効な弱識別器を選択しているといえる．一方，λ_n が小さい場合，有効でない弱識別器が選ばれていると考えることができる．提案手法では，Grabner らと同様に最も λ_n の値が小さいセレクタの弱識別器を更新している．

4.4.4 弱識別器を用いたパーティクルフィルタによる追跡

選択された弱識別器から構成される強識別器を用いて，パーティクルフィルタにより追跡する．パーティクルフィルタにより決められたサンプリング点に対して，強識別器 $H(x)$ を用いることで，各サンプリング点の尤度を得ることができる．このサンプリング点のうち，最も $H(x)$ が大きい位置を人物追跡結果として出力する．パーティクルフィルタの各サンプリング点の重みは，強識別器 $H(x)$ の値をもとに更新する．

4.5 提案手法と Online Boosting の違い

提案手法 (Online Real Boosting) と従来手法 (Online Boosting) の違いについて，表 4.1 にまとめる．従来手法と大きく異なるのは，各弱識別器が正解および非正解クラスの確率密度関数 $W^{y}_{n,m,j}$ を持っており，それを利用して Real AdaBoost により学習している点である．従来の Online Boosting は AdaBoost を用いているため，各弱識別器が確率分布を持つことができない．これにより，従来手法は新しいサンプルに対して，弱識別器の選択と重みの更新を行うため，追跡物体の形状や照明状態が急激に変化すると，誤って学習してしまうため，正しく追跡できないことが多い．また，正解・非正解クラスの確率密度関数を持たない Real AdaBoost でも同様に，新しいサンプルの確率分布を計算し，弱識別器を選択するため，急激な変化が生じた場合は正しく追跡できない．そこで，提案手法は，あらかじめオフライン学習時に得られた正解および非正解クラスの確率密度関数を保持し，新しいサンプルに応じて確率密度関数を更新している．正解データの確率密度関数はオフライン学習時に様々な顔の向きや照明変化を学習データに加えることにより，これらを考慮した分布を得ることができる．そのため，頭部の向きや照明状態が大きく変化した場合でも安定して追跡することができる．

表 4.1: 従来手法と提案手法の違い

	従来手法 (Online Boosting)	提案手法 (Online Real Boosting)
弱識別器の構成	局所特徴量	局所特徴量 正解および非正解クラスの確率密度関数
識別関数	閾値	確率密度関数
学習手法	AdaBoost	Real AdaBoost

図 4.11: 学習データ例

4.6 実験

4.6.1 従来手法との性能比較

提案手法である Real AdaBoost による Online Real Boosting の有効性を示すために，従来手法である AdaBoost による Online Boosting との頭部追跡性能の比較実験をおこなった．評価データとして Jepson ら [41] が評価に使用している動画像 1145 フレームを用いた．まず，頭部検出器をあらかじめオフラインで学習した．学習データは正解データ約 5000 枚，非正解データ約 9000 枚を 24×24 ピクセルで正規化し，両手法とも 300 個の弱識別器を学習した．学習データの一例を図 4.11 に示す．学習データは正面顔から横顔まで様々な顔向きの顔を含んでおり，頭部の上部と下部の長さが 20 ピクセルになるようにし，背景を一部含むようなサイズで正規化した．提案手法で選ばれた弱識別器の上位 10 個を図 4.12 に示す．選ばれた弱識別器のうち，(a), (b), (e), (g), (h) のように白と黒の 2 色で表現されている特徴量は Haar-like 特徴量であり，(c), (d) のように白線で囲まれた領域で表現している特徴は EOH 特徴量，(f), (i), (j) のように黒い領域と白線で表現されている特徴量は ABS Haar-like 特徴量である．

次に，追跡時のセレクタ数を 100 とし，あらかじめオフラインで学習した検出器の弱識別器から，オンラインで追跡に有効な弱識別器を選択する．また，パーティクルフィルタのパーティクル数は 200 とし，図 4.10 の式 (4.38) で最も信頼度が高い位置を追跡対象領域としている．また，両手法に対して，追跡する頭部の位置は初期フレームで同じ座標を与えている．表 4.2 に両手法及び Mean Shift による追跡対象領域の中心座標の正解座標からの平均誤差と標準偏差を示す．平均誤差は正解頭部の幅に対する正解座標からの位置ずれの割合としている．また，図 4.13 に各フレームでの正解

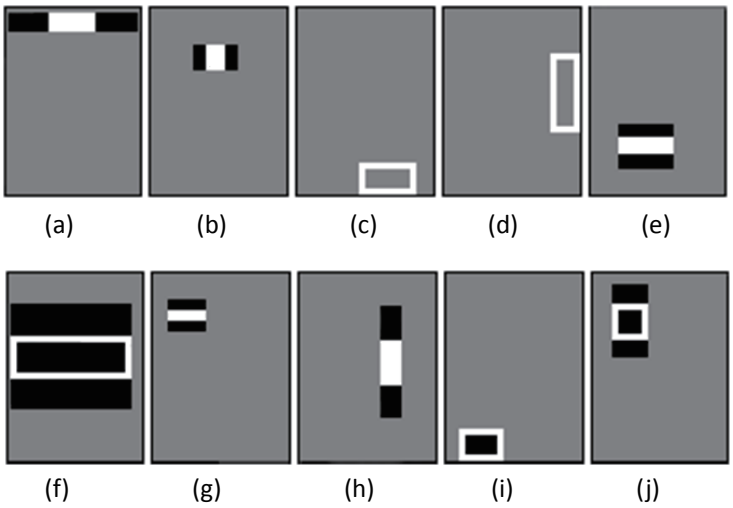


図 4.12: オフライン学習で選ばれた弱識別機

表 4.2: 追跡性能の比較

	平均誤差	標準偏差
Mean Shift	11.2%	20.1
OnlineBoosting	9.8%	6.1
提案手法	6.8%	5.7

位置からの誤差と評価データのサンプル画像を示す．表 4.2 より，提案手法が従来手法および Mean Shift に比べて平均誤差が小さく，追跡対象を正しく追跡できていることがわかる．また，提案手法は標準偏差が小さく，追跡対象の動きや隠れ，顔の向き変化が生じた場合でも追跡位置のばらつきが少ない．これらの代表的な結果として，200 フレーム付近および 800 フレーム付近の追跡結果を図 4.14 および図 4.15 に示す．図 4.14 および図 4.15 の (a) は Mean Shift，(b) は Online Boosting，(c) は Online Real Boosting の結果である．200 フレーム付近では，手による顔の隠れが生じており，Mean shift では手の動きにより追跡位置が大きくずれている．一方，提案手法は追跡位置の誤差が小さいことがわかる．800 フレーム付近では追跡対象人物が複雑な背景環境へ移動している．そのため，Mean shift では追跡対象と類似する背景に追跡位置がずれている．また，Online Boosting も追跡人物が複雑背景下へ移動することにより，弱識別器の判別誤りが増加するため，背景の影響を受け，追跡位置が右方向にずれている．一方，提案手法は複雑な背景下でも弱識別器の判別誤りが少ないため，正解に近い位置を正しく追跡できている．これにより，提案手法で用いた複数の特徴量タイプと Real AdaBoost による特徴量選択が有効であることが分かる．

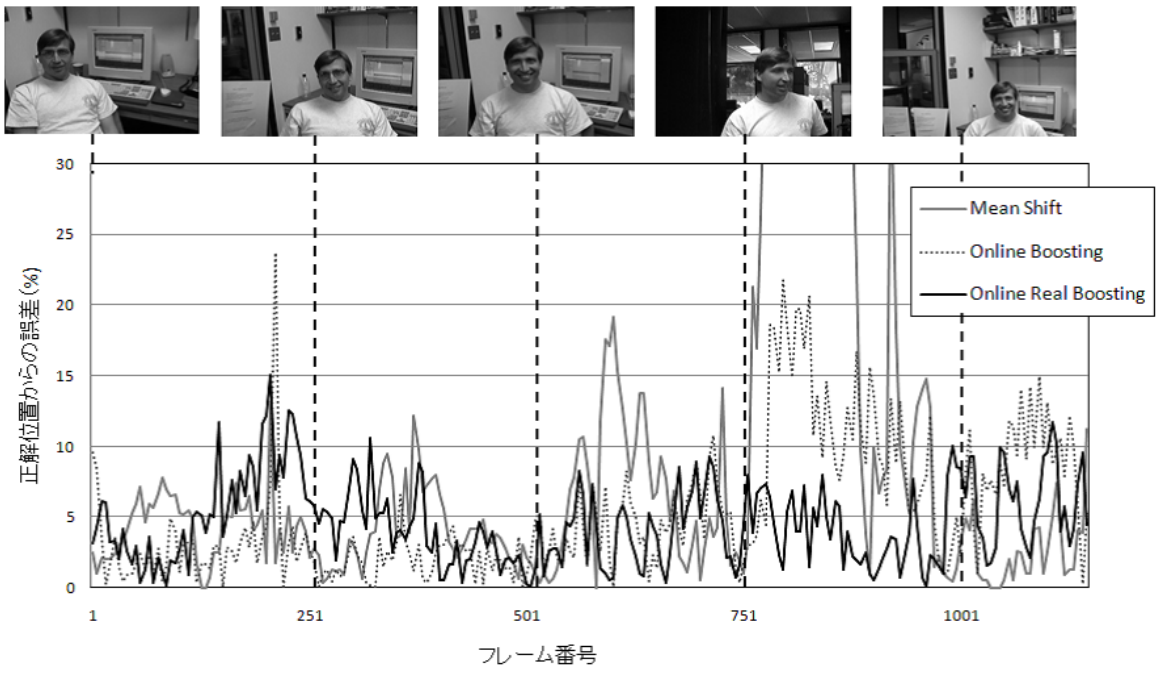


図 4.13: フレームごとの追跡位置の誤差

4.6.2 セレクタ数による性能比較

追跡時に用いるセレクタ数すなわち弱識別器の数による性能を比較する．図 4.16 にセレクタ数を変えた場合の正解位置からの平均誤差の変化を示す．評価データには前節と同様の動画像を用いた．セレクタ数を少なくすると，追跡位置の誤差が大きくなる．しかし，提案手法はセレクタ数を 30 とした場合でも，追跡位置の誤差は 8%程度であり，Online Boosting でセレクタ数を 90 とした場合の追跡位置の誤差 9.8%よりも小さい．これより，提案手法は Online Boosting に比べて，少ないセレクタ数でも安定した追跡が可能であることがわかる．

4.6.3 顔向き変化が生じるシーンでの性能

一般的なデジタルビデオカメラで撮影した動画像では様々な顔向き変化を生じることが多い．特に追跡対象が子供の場合，急激な顔向き変化を生じる．本節ではそのようなシーンにおいて，セレクタ数を少なくした場合の追跡性能を従来手法と比較する．評価データにはデジタルビデオカメラで撮影した動画像 500 フレームを用いた．表 4.3 に追跡の中心位置の誤差と標準偏差を示す．また，図 4.17 に Online Boosting，図 4.18 に Online Real Boosting による顔向き変化を生じるフレームでの追跡結果の例を示す．図 4.17 および図 4.18 の (a) はセレクタ数を 30，(b) は 50，(c) は 70，(d) は 90，(e) は 100 とした場合の追跡結果を示している．

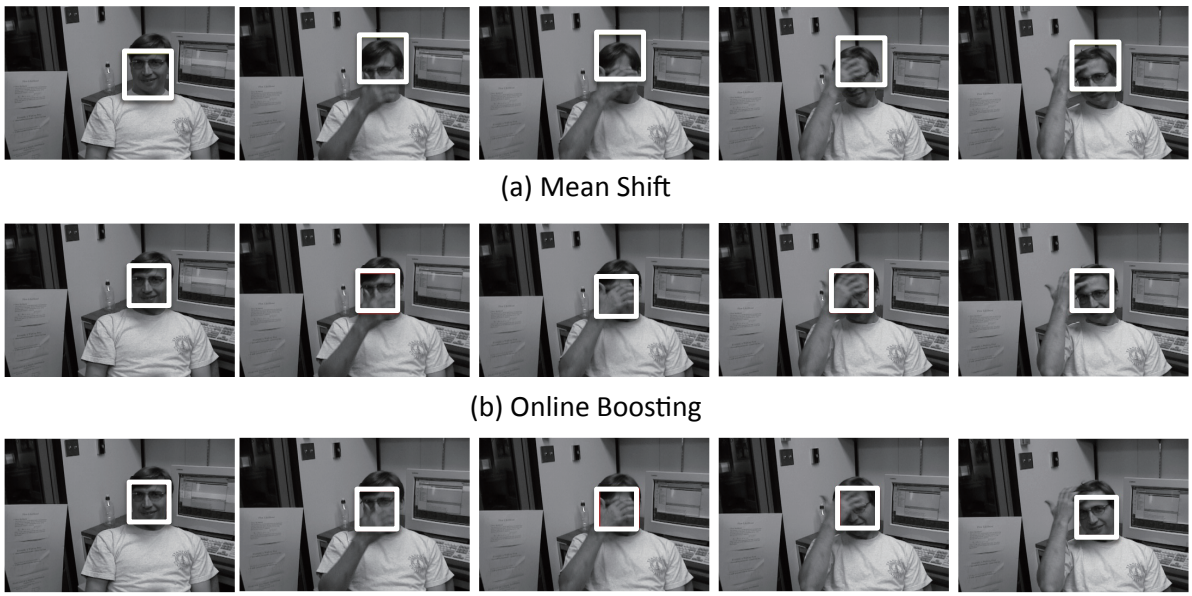


図 4.14: 200 フレーム付近における各手法の追跡結果例

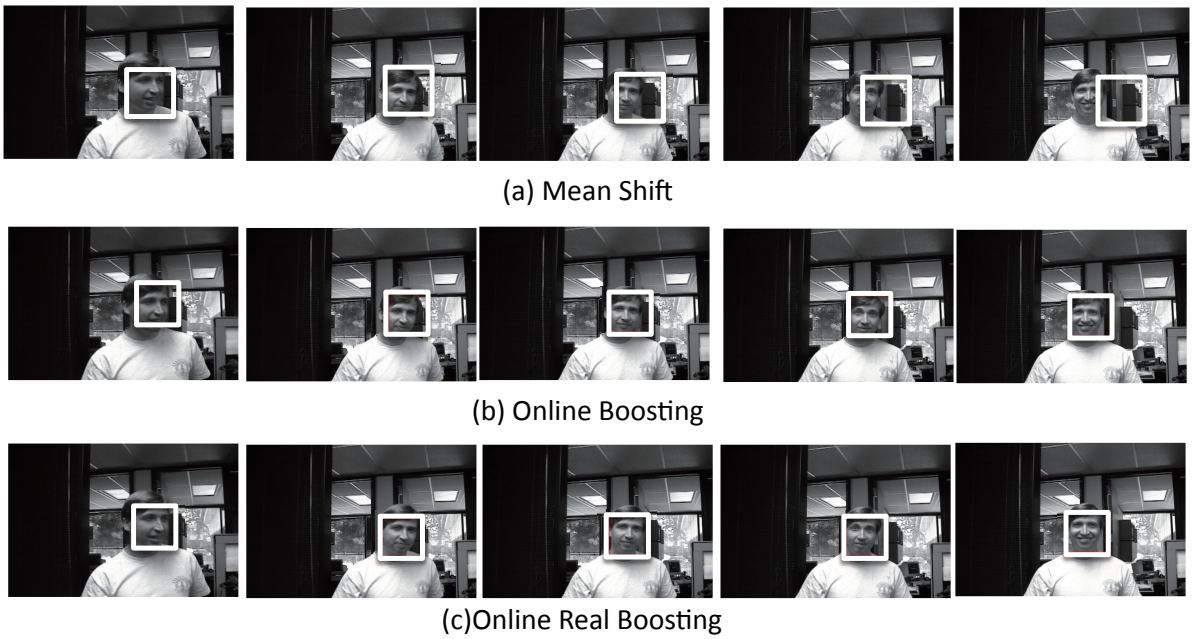


図 4.15: 800 フレーム付近における各手法の追跡結果例

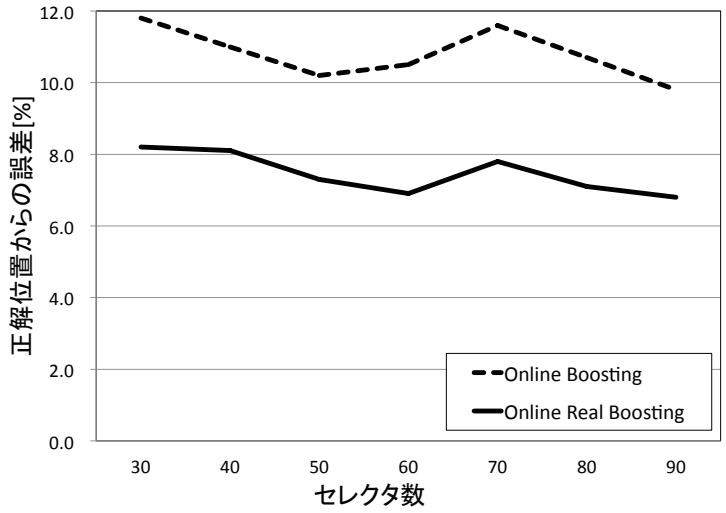


図 4.16: セレクタ数による正解位置からの誤差

表 4.3: 追跡性能の比較

	平均誤差	標準偏差
Mean Shift	12.0%	8.6
OnlineBoosting	9.5%	6.3
提案手法 (セレクタ数 100)	8.2%	5.1
提案手法 (セレクタ数 50)	8.8%	5.5
提案手法 (セレクタ数 30)	10.0%	6.4

表4.3 より，顔向き変化を生じるシーンにおいても提案手法は，Mean Shift や従来の Online Boosting に比べて，平均誤差が小さいことがわかる．本実験に用いたデータは 46 フレーム付近までは正面を向いており，127 フレーム目付近で斜め左下を向く．以降，265 フレーム目までにかけて，顔を上げて右を向くような動作をする．このような顔向きが変化するようなシーンにおいて，Online Boosting は十分なセレクタ数を設定しておくことで，正しく追跡できている．一方，提案手法では，セレクタ数を従来の Online Boosting より 50%削減した場合でも同等の追跡性能であることがわかる．これより，提案手法を用いることで，顔向き変化が生じるようなシーンにおいても，従来の Online Boosting より少ないセレクタ数で同等の追跡性能を示すことができた．

4.6.4 人体追跡への応用

提案手法はあらかじめ追跡したい物体の弱識別器をサンプル画像として学習しておくことで，さまざまな物体を追跡することができる．本節では，提案手法を人体追跡に応用した実験結果を示す．

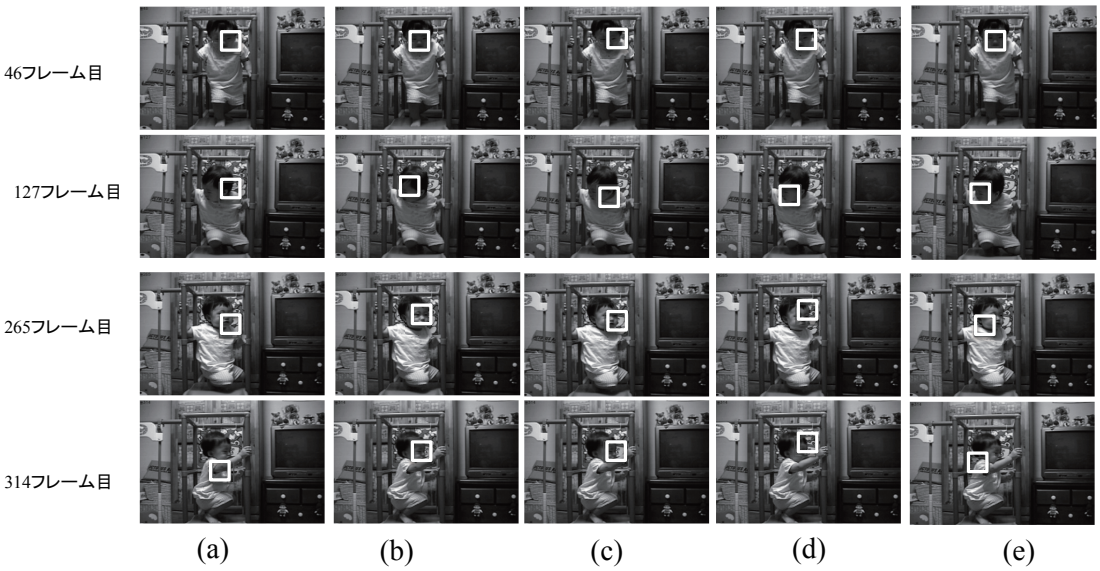


図 4.17: Online Boosting による顔向き変化のあるシーンでの性能

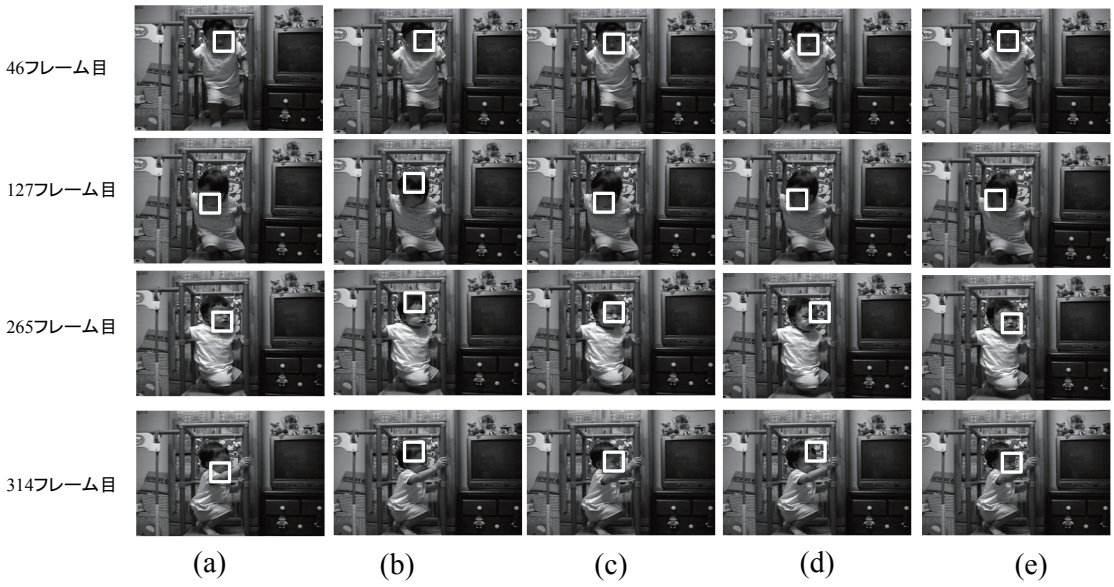


図 4.18: Online Real Boosting による顔向き変化のあるシーンでの性能

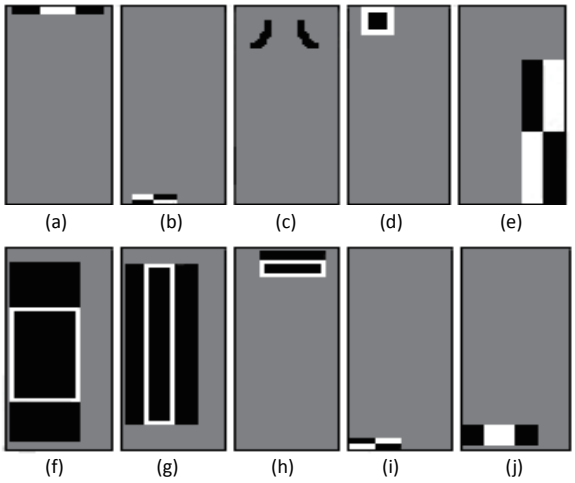


図 4.19: 人体データのオフライン学習で得られた弱識別器

人体データを約 6700 枚で用意し，オフライン学習を行う．人体のオフライン学習は頭部と同様に Real Adaboost により弱識別器 300 個を学習した．図 4.19 に選ばれた弱識別器のうち上位 10 個を示す．ここで，(c) のように，黒線で円弧のような特徴量が Edgelet 特徴量である．頭部の学習の際は Edgelet 特徴量が上位に選ばれていなかったが，人体の場合は頭部から肩の付近に相当するような位置が選ばれていることがわかる．図 4.20，図 4.21 に提案手法による人体追跡結果を示す．なお，図 4.20 および図 4.21 の追跡矩形の色は見やすさの観点から別色としている．図 4.20 は屋内で体の向きが変化するようなシーンである．体の向きに変化が生じるようなシーンにおいても人体を正しく追跡できていることがわかる．また，図 4.21 は照明状態が変化するようなシーンである．照明状態が暗く変化した場合でも正しく追跡することができていることがわかる．

4.7 まとめ

本章では新たなオンライン学習として，識別能力の高い特徴量を選択するために，Real AdaBoost を導入した Online Real Boosting による物体追跡手法を提案した．また，複数の特徴量タイプを組み合わせて追跡対象に合わせた有効な特徴量を選択できるようにした．これにより，従来の Online Boosting や Mean Shift に比べて複雑な背景下での追跡性能や顔向き変化が生じるようなシーンで追跡性能が向上していることを確認し，本手法の有効性を示した．また，提案手法の追跡対象を人体とした場合でも，体の向き変化や照明変化に対するロバスト性を確認することができた．

図 4.20: 体の向き変化が生じるシーンでの人体追跡の結果

図 4.21: 照明変化が生じるシーンでの人体追跡の結果

第5章

Soft Decision Featureによる姿勢変化に対応した物体追跡

Online Real Boosting を含む従来のオンライン学習手法は，オフライン学習で得られた弱識別器から追跡対象に適した弱識別器を選択することが可能である．しかしながら，人体のように形状変化を生じる物体を対象とする場合，直立姿勢や座り込み姿勢など形状ごとの弱識別器をあらかじめオフライン学習する必要がある．しかしながら，すべての姿勢に対応する弱識別器を用意することは困難である．そこで本章では，姿勢変化に柔軟に対応できる弱識別器の構造を提案する．提案手法は，オフラインで選ばれた弱識別器が追跡対象の形状変化に応じて適応的に弱識別器の形状を変化させることが可能である．評価実験では，従来の弱識別器の構造と追跡性能について比較実験を行い，提案手法が高精度に追跡できていることを示す．

5.1 従来のオンライン学習の問題点

Online Real Boosting は，オフライン学習で得られた弱識別器から追跡対象に適した弱識別器を選択することが可能である．しかしながら，人体のように形状変化を生じる物体を対象とする場合，形状ごとの弱識別器をあらかじめオフライン学習する必要がある．たとえば，人物を追跡対象とした場合，直立姿勢に対応した弱識別器，座り込み姿勢に対応した弱識別器が必要である．また，直立姿勢に対応した弱識別器の中でも正面向き，横向きなど向き方向ごとに対応した弱識別器も必要である．しかしながら，すべての姿勢に対応する弱識別器を用意することは困難である．また，オンライン学習においても弱識別器候補が増加すると弱識別器の選択にかかる計算コストが増加する．このように，従来のオンライン学習で学習できる弱識別器は形状が固定されており，また候補の数も限られている．そのため，形状変化を伴う物体を追跡する場合，最適な弱識別器が候補に存在せず，選択できないことが問題である．ここでは，あらかじめ定義された 1 つの形状にのみ対応する弱識別器を Hard Decision Feature（HDF）と定義する．

5.2 姿勢変化に対応した物体追跡

人物追跡における Online Real Boosting の問題点を解決するために，姿勢変化に柔軟に対応できる弱識別器の構造を提案する．提案手法は，オフラインで選ばれた弱識別器が，追跡対象の形状変化に応じて適応的に弱識別器の形状を変化させることができる Soft Decision Feature（SDF）とそのオフラインおよびオンライン学習手法からなる．本章では，これらについて述べる．

5.2.1 Soft Decision Feature(SDF)

本章で提案する SDF の概要を図 5.1 に示す．4 章で提案した Online Real Boosting における弱識別器（Hard Decision Feature）は，1 つの特徴形状（Feature）と物体らしさを出力する識別関数（Hypothesis），物体および非物体の確率密度関数（Probability density）からなる．一方，本章で提案する SDF は，複数の特徴形状（Features）と 1 つの識別関数（Hypothesis），物体および非物体の確率密度関数（Probability density）からなる．特徴形状は基準となる形状（Basis type）と変形した形状（Transform type）に分かれる．

Transform type の形状は，オフライン学習において，式 (5.1) のように Basis type の形状に対して，変形関数 G により変形させることで得ることができる．

$$F_{trans} = G(F_{org}; p, s) \tag{5.1}$$

ここで，F_{trans} は変形させた Transform type の形状であり，F_{org} は Basis type の形状である．変形パラメータは，形状の位置 p，スケール s である．

オフライン学習において，これらの変形により得られた Transform type の中から Basis type と類

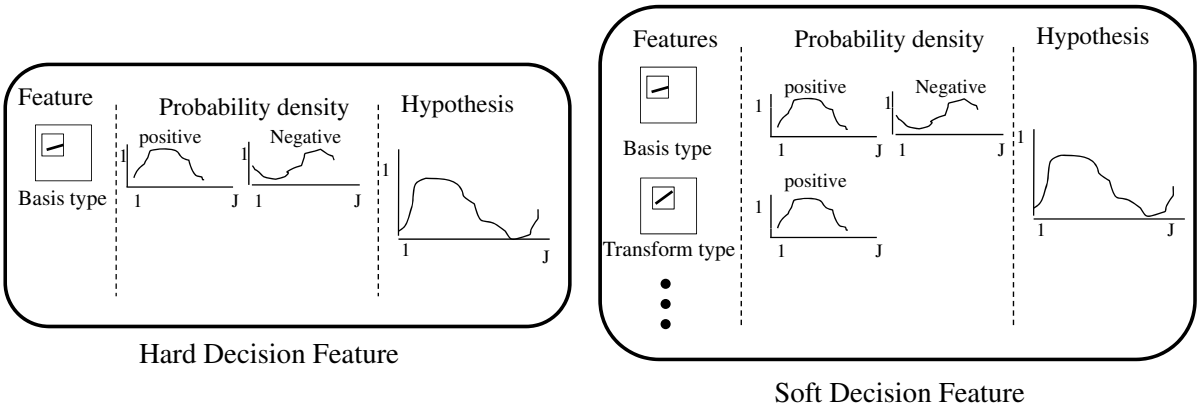


図 5.1: SDF の概要

似している形状をグルーピングし，SDF を学習する．オンライン学習では，複数の特徴形状から現在の追跡物体に最適な形状を選択する．これらの各学習について，次節以降で詳細に述べる．

5.2.2 SDF のオフライン学習

前節で述べた SDF のオフライン学習手順について，図 5.2 に示す．ここからは，各学習手順について詳細に説明する．また，従来の Online Real Boosting との違いについても述べる．

■ オフライン学習の準備

あらかじめ正解 $(y = +1)$ または非正解 $(y = -1)$ にラベル付けされた学習サンプル X= $(x_n, y_n; n \in [1, ..., N], y \in [-1, +1])$ を用意し，各サンプルの重みを $\lambda_n = 1/N$ に初期化する．オフライン学習では，M 個の弱識別器候補から T 個の弱識別器を選択する．各弱識別器の選択試行 t では，図 5.2 の手順 2.1 から手順 2.5 を繰り返し行う．

■ 弱識別器候補の生成

はじめに，各弱識別器候補における正解クラスおよび非正解クラスに対する確率密度関数 W^y_{org} を求める．

確率密度関数は正解クラスまたは非正解クラスらしさを表す関数であり，要素数 J のヒストグラムで表現されている．図 5.2 の手順 2.1.1 のように，学習サンプル x_n から得られた特徴量に対応するヒストグラムの要素位置に対して，重み λ_n を加算することで確率密度関数を導出することができる．

Input:学習サンプル X= (x_n, y_n); $n \in [1, N]$, $y_n \in [-1, +1]$

Outpt:強識別器 H_{off}(X)

1. 学習サンプルの重み初期化

$$\lambda_n = 1/N \tag{5.2}$$

2.For t $= 1, \ldots, T$　//学習の試行回数

2.1 For $m = 1, \ldots, M$　//弱識別器候補数

2.1.1 確率密度関数 W^y_{org} の作成

$$W^y_{org,m,j} = W^y_{org.m,j} + \lambda_n \tag{5.3}$$

2.1.2 弱識別器の変形

$$F_{trans} = G(F_{org}; p, s) \tag{5.4}$$

2.1.3 確率密度関数 W^y_{trans} の作成

$$W^y_{trans,m,j} = W^y_{trans,m,j} + \lambda_n \tag{5.5}$$

2.1.4 類似度を算出

$$S_m = 2\sum_{j=1}^{J} \sqrt{W^{+1}_{org,m,j} W^{+1}_{trans,m,j}} \tag{5.6}$$

2.1.5 確率密度関数を結合

if $S_m > \theta$

$$W^{+1}_{m,j} = W^{+1}_{org,m,j} + W^{+1}_{trans,m,j} \tag{5.7}$$

2.1.2 から 2.1.5 のグルーピングを繰り返す

2.1.6 評価値 Z の計算

$$Z_m = 2\sum_{j=1}^{J} \sqrt{W^{+1}_{m,j} W^{-1}_{m,j}} \tag{5.8}$$

2.2 弱識別器候補の選択

$$m^* = \arg\min_m Z_m \tag{5.9}$$

$$t = m^* \tag{5.10}$$

2.3 識別関数 h_t(X) の作成

For $j = 1, \ldots, J$　//ヒストグラムの要素数

$$h_{t,j}(X) = \frac{1}{2} \ln\Big(\frac{W_{t,j}^{+1} + \beta}{W_{t,j}^{-1} + \beta}\Big) \tag{5.11}$$

2.4 エラー率の計算 (初期値 : $\epsilon_t = 0$)

$$\epsilon_t^y = \epsilon_t^y + \lambda_n \cdot |h_t(x_n)| \tag{5.12}$$

$$\epsilon_t = \frac{\epsilon_t^{+1}}{(\epsilon_t^{+1} + \epsilon_t^{-1})} \tag{5.13}$$

2.5 重みの更新

if $y_n \cdot h_t(x_n) > 0$

$$\lambda_n = \frac{\lambda_n}{2(1 - \epsilon_t)} \tag{5.14}$$

else

$$\lambda_n = \frac{\lambda_n}{2\epsilon_t} \tag{5.15}$$

3. 強識別器

$$H_{off}(X) = \sum_{t=1}^{T} h_t(X) \tag{5.16}$$

図 5.2: SDF によるオフライン学習

■ 形状のグルーピング

次に，Transform type としてグルーピングする形状を作成する．手順 2.1.2 において，m 番目の弱識別器候補の形状 F_{org} を Basis type として形状を変形させ，変形した形状 F_{trans} を得る．変形は，各変形パラメータを変えて行い，変形した形状ごとに手順 2.1.3 において確率密度関数 W^{y}_{trans} を求める．Transform type の確率密度関数の導出は手順 2.1.1 の Basis type の確率密度関数と同様である．導出した Basis type と Transform type の確率密度関数の類似度を手順 2.1.4 のようにバチャタリア距離をもとに算出する．類似度が閾値 θ より大きい場合，SDF としてグルーピングを行い，確率密度関数を手順 2.1.5 のように結合する．一方，類似度が閾値 θ 以下の場合は結合しない．手順 2.1.2 の変形関数 G の各変形パラメータをあらかじめ決めた範囲内で変化させ，すべての組み合わせについて手順 2.1.2 から手順 2.1.5 を行い，さらにグルーピングしていく．このようなグルーピング処理をすべての弱識別器候補に対して行う．そして，手順 2.1.6 において，正解クラスと非正解クラスの類似度を評価値 Z として求める．ここで，確率密度関数はヒストグラムの要素を累積すると 1 になるように正規化されている．

■ 弱識別器の選択

手順 2.2 において，最も評価値 Z が小さい弱識別器候補を弱識別器として選択する．選択された弱識別器の識別関数 h_t(X) を手順 2.3 のように，正解クラスおよび非正解クラスの確率密度関数 W^{y}_{t} から求める．ここで，β はゼロ割を防ぐための定数である．

■ 重みの更新と強識別器の生成

手順 2.4 において，選択した弱識別器に対するエラー率を全学習サンプルから算出し，学習サンプルの重み λ_n を手順 2.5 のように更新する．これにより，選択された弱識別器に対して，正しく識別できたサンプルの重みは小さく，誤って識別されたサンプルの重みは重く更新される．これらのステップを繰り返し行い，T 個の弱識別器を選択し，強識別器 H_{off}(X) を得る．

■ 従来手法との違い

従来の Online Real Boosting との違いは，図 5.2 の手順 2.1.2 から手順 2.1. である．提案手法は，オフライン学習において，正解クラスの確率密度関数が類似する形状をグルーピングし，1 つの弱識別器として扱う．ここで，非正解サンプルは形状の変化有無により変わることはない．そのため，正解サンプルに対する確率密度関数が類似する場合，弱識別器の識別関数が類似することになる．手順 2.1.2 の変形プロセスにおいて，各変形パラメータ値の範囲を局所的に限定することで Basis type が着目する位置における小さな姿勢変化に対応した Transform type を得ることができる．これにより，各 Transform type がもつ確率密度関数は大きく異なることがなく，識別関数の共通化を図ることができる．

5.2.3 SDF のオンライン学習

SDF を用いたオンライン学習の手順を図 5.3 に示す．提案手法は図 5.3 の手順 3.1 と手順 3.2 における SDF の Basis type 形状の確率密度の更新と最適な SDF の選択処理，手順 3.3 と手順 3.4 における弱識別機の最適化と形状選択からなる．従来の Online Real Boosting は，手順 3.1 と手順 3.2 の確率密度の更新と選択処理のみであり，手順 3.3 と手順 3.4 の形状最適化を導入した点が従来手法との違いである．ここからは，詳細なオンライン学習の手順について説明する．

■ オンライン学習の前処理

オンライン学習は，現時刻における追跡対象物体を正解サンプル（$y = +1$），追跡対象物体の周囲を非正解サンプル（$y = -1$）として入力する．オフラインで学習された T 個の弱識別器を従来手法と同様に R 個のグループにランダムに分割する．各グループから SDF を選択する際，多くの候補が存在すると選択処理に時間がかかる．そのため，従来手法では高速化のためにグループにランダム分割している．

■ SDF の選択

図 5.3 の手順 2 のように，学習サンプルに重み λ_k を与える．

分割されたグループごとに図 5.3 の手順 3 を行い，弱識別器をセレクタとして選択する．まず，現時刻のサンプルの重みを用いて，グループ r における弱識別器候補 m の正解クラスおよび非正解クラスの確率密度関数 $W^y_{r,m}$ を手順 3.1.1 のように更新する．確率密度関数の更新処理はオフライン学習と同様に，学習サンプルから得られた特徴量に対応するヒストグラムの要素位置に対して，重み λ_k を加算する．また，ヒストグラムの累積が 1 になるように正規化する．次に，弱識別器候補に対して，評価値 Z を手順 3.1.2 のように算出する．手順 3.2 において，最も評価値が小さい弱識別器候補をセレクタとして選択する．

■ 形状の最適化

手順 3.3 では，セレクタとして選択された弱識別器の形状の最適化を行う．手順 3.3.1 および手順 3.3.2 において，弱識別器がもつ各 Transform type 形状の確率密度関数を更新し，評価値 Z を算出する．そして，手順 3.4 のように評価値 Z が最も小さい形状をセレクタの形状として選択する．これら手順 3 を各グループに対して行い，セレクタとして選択された SDF を結合し，強識別器 H_{on}(X) を得る．

従来の Online Real Boosting は，手順 3.2 で得られた弱識別器をセレクタとして用いている．一方，提案手法は，弱識別器候補が複数の形状をもっており，追跡物体の形状に適した形状をさらに選択することができる．これにより，物体の形状変化に対してより頑健に追跡することが可能である．

Input:学習サンプル X= (x_k, y_k); $k \in [1, K]$, $y_k \in [-1, +1]$

弱識別器候補 $H_{off}(\mathrm{X}) = h_t(\mathrm{X})$; $t \in [1, T]$

Output:強識別器 $H_{on}(\mathrm{X})$

1. 弱識別器 T 個を弱識別器グループ R 個に分割

2. サンプルデータの重みを初期化

$$\lambda_k = \frac{1}{K} \tag{5.17}$$

3.For $r = 1, 2, \ldots, R$ //弱識別器グループ数

3.1 For $m = 1, \ldots, T/R$ //弱識別数

3.1.1 弱識別器の確率密度関数 $W^y_{r,m}$ を更新

$$W^y_{r,m,j} = W^y_{r,m,j} + \lambda_k \tag{5.18}$$

3.1.2 弱識別器の評価値 Z を算出

$$Z_{r,m} = 2\sum_{j=1}^{J} \sqrt{W^{+1}_{r,m,j} W^{-1}_{r,m,j}} \tag{5.19}$$

3.2 弱識別器をセレクタとして選択

$$m^* = \arg\min_m Z_{r,m} \tag{5.20}$$

$$h_{org,r}(X) = h_{r,m^*}(X) \tag{5.21}$$

3.3 SDF の弱識別器の最適化

For $l = 1, \ldots, L$ //変形形状数

3.3.1 変形形状の確率密度関数を更新

$$W^y_{trans,r,l,j} = W^y_{trans,r,l,j} + \lambda_k \tag{5.22}$$

3.3.2 評価値 Z を算出

$$Z_{trans,r,l} = 2\sum_{j=1}^{J} \sqrt{W^{+1}_{trans,r,l,j} W^{-1}_{trans,r,l,j}} \tag{5.23}$$

3.4SDF の形状を選択

$$l^* = \arg\min_l Z_{trans,r,l} \tag{5.24}$$

$$h_r(X) = h_{r,l^*}(X) \tag{5.25}$$

4. 強識別器

$$H_{on}(X) = \sum_{r=1}^{R} h_r(X) \tag{5.26}$$

図 5.3: SDF によるオンライン学習

5.2.4 SDF に用いる特徴量

SDF は，物体に対する確率密度関数が類似している複数の特徴量の形状をグルーピングした弱識別器構造であり，追跡物体の形状変化に柔軟に対応することができる．ここまでは，SDF を作成するためのオフライン学習と最適な形状を選択するオンライン学習について述べた．本論文では，これらを実装する上でベースとなる特徴量に人物検出手法で高い性能を得ている Edgelet と EHOG を用いる．

5.2.5 SDF を用いたパーティクルフィルタによる追跡

物体を追跡する方法として，顔検出や人体検出，追跡問題でのテンプレートマッチングのように網羅的に探索するスライディングウィンドウと，パーティクルフィルタなどの確率分布を利用したサンプリング方式がある．スライディングウィンドウの場合，領域全体または部分領域全体に対して強識別器による識別処理を行うため性能を向上させることができるが，処理時間がかかる．一方，サンプリング方式は，前フレームでのサンプリング点に対して尤度をもとに重みづけを行い，処理する領域を限定することができる．そのため，高速な追跡処理が可能である．また，性能については，事前確率が大きく誤らない限りは，性能を保つことができる．提案手法では，高速化を図るために，パーティクルフィルタを追跡手法として用いる．パーティクルフィルタの状態ベクトルとして，追跡物体の位置 (x, y) およびスケール s を用いる．また，システムモデルはランダムウォークとし，各パーティクルは，前フレームでの中心座標を中心に追跡物体の幅の 2 倍，大きさは ± 0.1 倍の範囲に対して 2 次元正規分布に従ってランダムに散布している．各パーティクルの重みは，オンライン学習で得た強識別器が出力する値を用いて更新する．

5.3 実験

5.3.1 実験概要

SDF を利用した Online Real Boosting と従来の HDF を利用した Online Real Boosting の追跡性能の比較実験を行った．比較実験は人体の向きおよび姿勢が変化するシーン（約 800 フレーム）での追跡位置のばらつきを比較した．

はじめに，24×60 ピクセルに正規化した学習データを約 5000 枚用いて，オフラインで人体の弱識別器を 500 個学習した．図 5.4 にオフライン学習に用いた学習サンプル例を示す．学習サンプルは直立姿勢の画像のみ用いており，実験の映像に含まれるような座り込みの姿勢は含んでいない．

変形関数の変形パラメータについて，位置 p は，Basis type 形状に対して ± 1 ピクセル，大きさ s は Basis type 形状の幅に対して -0.8 倍から 1.2 倍まで 0.1 倍刻みで変化させている．また，勾配方向 g は，着目する勾配方向ヒストグラムの要素位置を ± 1 変化させている．これらの値は実験的に決めているが、変化させる範囲を大きくすることでより離れた位置にある形状をグルーピングするこ

図 5.4: オフライン学習サンプル例

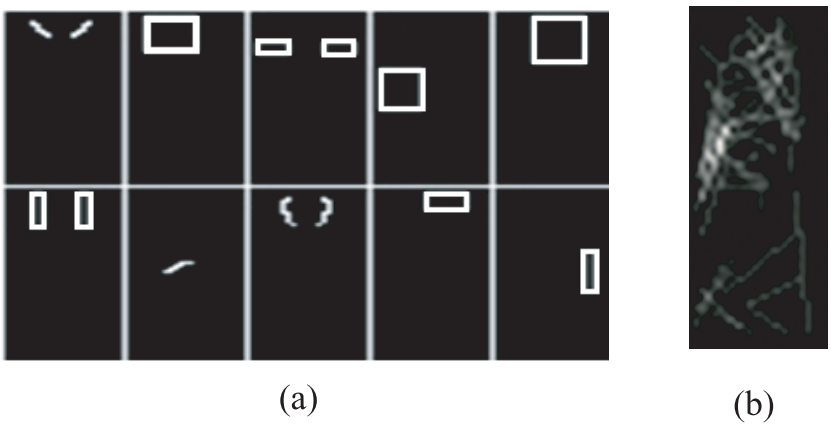


図 5.5: オフライン学習結果
(a) 上位 10 個の弱識別器，(b) 全弱識別器による特徴分布

とができる．しかしながら，基準となる Basis type の形状と大きく異なる形状がグルーピングされる可能性がある．また，グルーピングを行う閾値 θ については，実験的に 0.8 とした．

オフラインでの学習により，選ばれた弱識別器の上位 10 個と全弱識別器を重ね合わせた特徴分布を図 5.5 に示す．図 5.5(a) の矩形で描画されている特徴量が EHOG であり，白線で描画されている特徴量が Edgelet である．図 5.5(b) の特徴分布は Edgelet の形状と EHOG の勾配方向を重ね合わせている．これより，特徴量は頭部付近や体の輪郭付近に多く分布していることがわかる．また，SDF の平均形状数は，3.5 個である．

従来手法および提案手法のオフライン学習で選択する弱識別器数はともに 500 個であり，オンライン時のセレクタ数も同様に 50 と同じである．また，各セレクタは 500 個の弱識別器をランダムに分けた弱識別器のグループから選択する．グループ数はセレクタ数と同じ 50 であり，各グループには 10 個の弱識別器が含まれている．これらの値およびランダム分割は，従来手法における実験と同

様である．

提案手法が，追跡物体の位置および姿勢が変化しても正しく追跡していることを確認するために，スライディングウィンドウとパーティクルフィルタの両手法を用いて実験を行う．スライディングウィンドウは，前フレームで検出した物体の幅に対して 2 倍の範囲を網羅的に探索する．一方，パーティクルフィルタは，前フレームで検出した物体の中心を基準に幅の 2 倍の範囲内にパーティクルを散布する．パーティクル数は 100 個である．

追跡性能は追跡位置の精度を比較した．評価映像は，あらかじめ人体領域を囲むような矩形を正解矩形として入力している．正解矩形の中心座標を求め，中心座標と追跡結果の中心座標の距離をもとに位置精度を算出する．位置精度 Pa は式 (5.27) のように，人体領域の幅に対する正解中心座標からの距離の割合とする．

$$Pa = Dist(Tracking, GroundTruth)/Width[\%] \tag{5.27}$$

5.3.2 追跡精度の比較

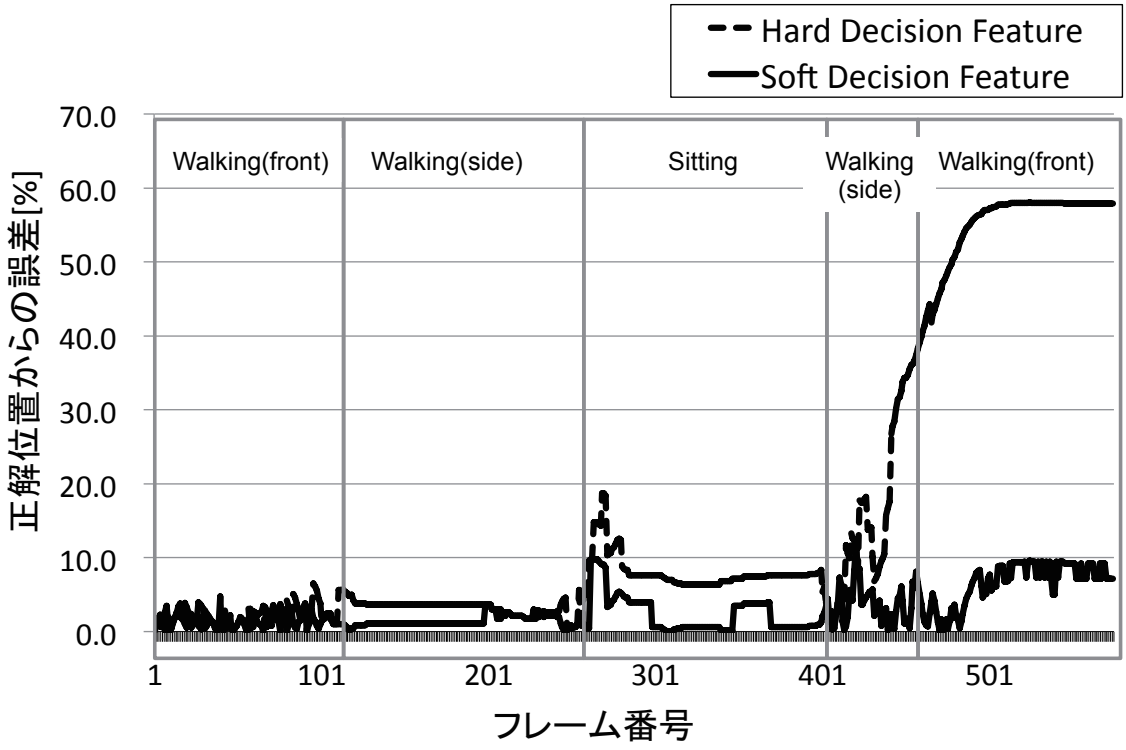


図 5.6: 各フレームでの追跡位置精度

体の向きおよび姿勢変化が生じる映像を用いて，追跡位置の精度を比較した．映像は正面向きの直立と横向きの直立，横向きの座り込みが含まれており，直立から座り込み後，直立するような映像である．

図 5.6 にスライディングウィンドウによる時系列での追跡位置精度を示す．これより，最初の正面向きに直立している最初の 100 フレームは，HDF および SDF の位置精度に大きな差はない．

横向きの直立時および座り込み時は，HDF に比べて SDF の方が位置精度がよい．さらに座り込み後の横向きおよび正面の直立時，HDF は位置精度が悪くなり追跡が失敗している．一方，SDF は座り込みから直立への姿勢変化時に位置精度にばらつきがあるものの追跡は成功している．

表 5.1: 追跡性能の比較 (スライディングウィンドウ)

	SDF		HDF	
姿勢変化	精度 Pa[%]	標準偏差	精度 Pa[%]	標準偏差
(a) 直立 (正面)	1.17	0.81	1.13	1.03
(b) 直立 (横)	1.62	1.45	2.46	2.35
(c) 座り込み	1.33	2.69	4.24	7.22
(d) 直立 (正面)	1.91	4.74	Miss	-
(e) 直立 (横)	1.90	3.89	Miss	-
全体	1.58	2.71	6.46	38.97

表 5.2: 追跡性能の比較 (パーティクルフィルタ)

	SDF		HDF	
姿勢変化	精度 Pa[%]	標準偏差	精度 Pa[%]	標準偏差
(a) 直立 (正面)	1.73	1.17	1.84	1.91
(b) 直立 (横)	1.83	4.26	3.95	9.72
(c) 座り込み	2.10	4.12	7.34	4.08
(d) 直立 (正面)	3.07	4.64	Miss	-
(e) 直立 (横)	2.70	4.19	Miss	-
全体	2.28	3.76	7.52	39.99

表 5.1 および表 5.2 にスライディングウィンドウとパーティクルフィルタの 2 つの追跡手法における各姿勢の追跡位置精度の平均と標準偏差を示す．表 5.1 より，SDF は追跡時に姿勢変化が生じても標準偏差は大きくなるものの平均位置精度に大きな差はない．一方，HDF は姿勢変化が生じると平均位置精度が悪くなり，座り込み後の直立時に追跡が失敗している．また，表 5.2 よりスライディン

グウィンドウを用いた方が追跡位置精度は良いが，パーティクルフィルタを用いた場合と大きな差はない．そのため，追跡速度を考慮する場合は，パーティクルフィルタを用いても問題ないといえる．

図 5.7: HDF による追跡結果例
(a) 直立 (正面) 時，(b) 直立 (横) 時，(c) 座り込み時，
(d) 直立 (横) 時，(e) 直立 (正面) 時の追跡結果

図 5.8: SDF による追跡結果例
(a) 直立 (正面) 時，(b) 直立 (横) 時，(c) 座り込み時，
(d) 直立 (横) 時，(e) 直立 (正面) 時の追跡結果

パーティクルフィルタを用いた場合の追跡結果例として，図 5.7 に提案手法，図 5.8 に従来手法の結果を示す．図 5.8(c) のように，SDF は座り込み時も人物の領域を正しく追跡することができていることがわかる．一方，HDF の場合，図 5.7(c) のように座り込み時に追跡位置がずれていることがわかる．また，図 5.7(d) および (e) のように座り込み後の直立時は，追跡対象を正しく捉えることができていない．これは，追跡過程における位置ずれの影響により，弱識別器の確率密度関数の更新が正しく行われていないためである．提案手法の SDF は座り込みまでの位置精度が高いため，確率密度関数の更新が正しく行われており，座り込み後の直立時も位置精度が高く，図 5.8(d) および (e) のように追跡が成功している．

これより，提案手法は学習した直立姿勢だけでなく，座り込みの姿勢に対して柔軟に対応できており，姿勢変化が生じた場合でも正しく追跡対象を追跡することが可能である．

5.3.3 姿勢変化への汎化性

前節では，座り込みを含む姿勢変化を伴うシーンにおいて，提案手法が姿勢変化にロバストに追跡できることを示した．本節では，その他の姿勢変化に対する追跡性能について実験を行った．姿勢

変化として，姿勢変化と急激な移動を伴うジャンプシーンと姿勢変化と隠れが生じるモーグルの競技シーンを対象とした．それぞれの追跡結果を図 5.9 および図 5.10 に示す．なお，本実験に用いた弱識別器は図 4.2 の実験と同じものであり，セレクタ数およびパーティクル数も同じである．

図 5.9 および図 5.10 のように，提案手法は大きな姿勢変化が伴うフレームや隠れが生じるシーンでも正しく追跡することができている．これらの実験においても，提案手法が姿勢が変化しても正しく追跡することができており，姿勢変化に対して頑健であることが分かる．

図 5.9: 姿勢変化と急激な移動を伴うシーンでの追跡結果例

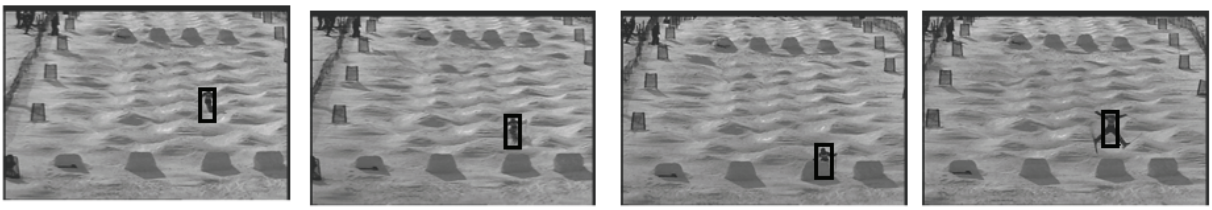


図 5.10: 姿勢変化と隠れが生じるシーンでの追跡結果例

5.4 従来手法との比較

5.4.1 選択された特徴量の比較

座り込み時における SDF と HDF の追跡結果と選択された弱識別器を重ね合わせた特徴分布を図 5.11 に示す．

図 5.11(a) は SDF による追跡結果，図 5.11(b) はこの時に選択された弱識別器の特徴分布である．また，図 5.11(c) は HDF による追跡結果，図 5.11(d) は弱識別器の特徴分布である．図 5.11(b) および (d) の線の色は弱識別器の重なりが大きいほど白色で表現されている．座り込み時，SDF は頭部から背中付近に分布する弱識別器が選択されている．また，足元や腕付近にも均等に弱識別器が配置されており，体全体を捉えることができている．一方，HDF は図 5.11(d) のように頭部付近に

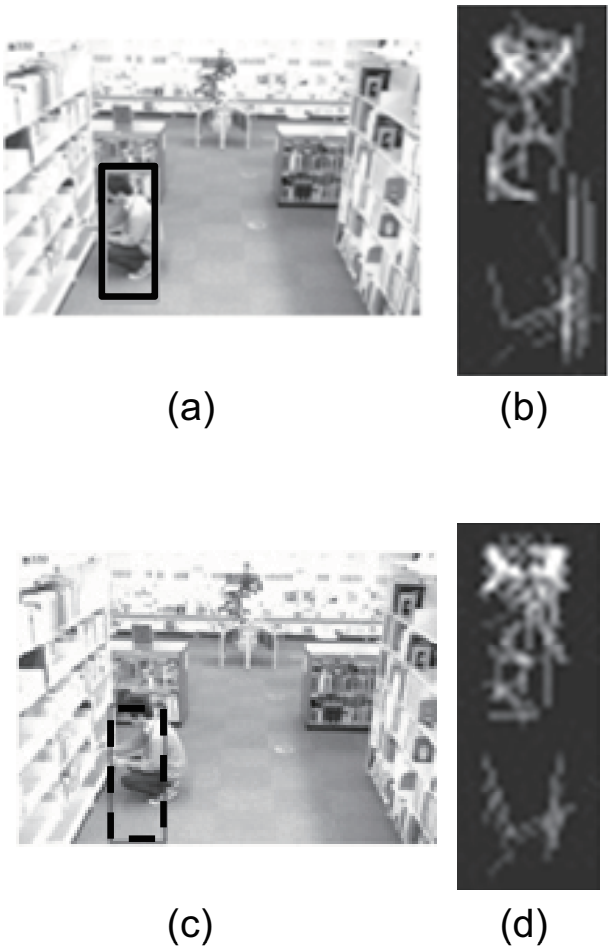


図 5.11: 選択された弱識別器
(a)SDF による座り込み時の追跡結果，(b) 座り込み時に選択された SDF による特徴分布，
(c)HDF による座り込み時の追跡結果，(d) 座り込み時に選択された HDF による特徴分布

集中的に弱識別器が選択されているため，追跡矩形が人体の下側にずれている．これは，HDF は姿勢変動に対応する形状ではなく，姿勢変化が生じていない頭部付近にある弱識別器が集中的に選択されており，集中した部分の形状が変化すると追跡することができなくなっているためである．一方，提案手法はオフライン時に形状の変形を考慮し，それに対応する形状をグルーピングできている．そのため，座り込みのような姿勢変動が生じた場合でも弱識別器が同じ場所に集中せずに分散し，識別に有効な形状に変化した弱識別器が選択されることで追跡することができている．

また，これらの違いは弱識別器の確率密度関数の更新に大きく影響する．提案手法の場合は形状の変化に合わせて最適な弱識別器が選択され，その確率密度関数が正しく更新されている．そのため，座り込みによる姿勢変化が生じた後でも正しく人体を追跡することができている．一方，HDF は，座り込み時に形状変化に合わせた弱識別器が選択されず，比較的確率の高い弱識別器を選択しており，弱識別器の確率密度関数が正しく更新されず，人体を追跡することができなくなっている．

表 5.3: 弱識別器数を変えた場合の HDF の追跡性能

	SDF		HDF(same weak no.)		HDF(2pose)	
姿勢変化	精度 Pa[%]	標準偏差	精度 Pa[%]	標準偏差	精度 Pa[%]	標準偏差
(a) 直立 (正面)	1.73	1.17	1.63	1.60	2.65	3.01
(b) 直立 (横)	1.83	4.26	2.88	5.93	4.02	4.56
(c) 座り込み	2.10	4.12	4.95	4.87	10.01	15.91
(d) 直立 (正面)	3.07	4.64	6.64	5.54	14.32	21.23
(e) 直立 (横)	2.70	4.19	6.23	5.81	17.71	17.66
全体	2.28	3.76	4.46	4.75	9.74	12.74

5.4.2 弱識別器数および対応姿勢数による性能比較

HDF の弱識別器候補数を SDF のグルーピングされた形状と同等数 (=1750 個) にした場合の追跡性能および，直立と座り込みの２つの姿勢についてオフライン学習した場合の追跡性能について，提案手法と比較する．２つの姿勢に対するオフライン学習について，座り込み姿勢は学習サンプルを 2000 枚用意し，500 個の弱識別器を学習した．直立姿勢はこれまでの実験と同じものを用いた．オンライン学習は，各姿勢の弱識別器を統合し，ランダムに分割した．パラメータについてはこれまでの実験と同様である．比較結果を表 5.3 に示す．これより，弱識別器数を増加させた場合 (=same weak no.)，追跡精度は向上するものの SDF よりは劣る．一方，2 つの姿勢に対応した学習を行った場合 (=2 pose) については性能が低下する．従来手法の弱識別器を SDF の形状と同等数にした場合および，2 つの姿勢に対応させた場合，多くの HDF は人体の形状の変化が少ない頭部付近に集中している．そのため，座り込み時に誤って直立姿勢の頭部付近の HDF が選ばれているケースがあり，縦方向に対する位置精度が低下している．一方，提案手法は形状の違いが生じる領域をグルーピングにより 1 つの弱識別器として選択することができる．そのため，形状変化が生じた場合に特定の領域に SDF が集中することがない．よって，姿勢変化が生じるシーンにおいても正しく追跡することができている．

これより，提案手法は，従来手法の弱識別器数を増加させた場合や複数の姿勢に対応させた場合と比較しても，追跡精度を向上させることができる．

5.4.3 処理速度の比較

姿勢変化に対応した SDF による追跡と姿勢ごとの HDF による追跡について，処理時間を比較する．HDF は前節と同様に，SDF がもつ全形状と同等数の場合と直立と座り込みの 2 つの姿勢に対応する場合を考える．オフラインで学習する 1 つの強識別器は 500 個の弱識別器からなる．表 5.4 に Intel Core 2 Duo 2.4GHz における処理速度を示す．提案手法の処理速度は，約 30ms とリアルタイ

ムでの追跡を実現している．

一方，従来手法の弱識別器数を SDF の形状と同等数にした場合の処理速度は 49ms となっており，オンライン学習における弱識別器選択の処理時間が増加した分，処理速度が遅くなっている．また，2 つの姿勢対応した従来手法は 42ms となっている．これより，提案手法は 2 つの姿勢に対応した従来手法と比較して，28%高速化することができている．

処理速度はセレクタ数とセレクタに含まれる弱識別器数に依存する．そのため，オフライン学習により選択される弱識別器の識別能力が高ければ，セレクタ数を削減することができ，さらに高速化することができる．

表 5.4: 処理速度の比較

	SDF	HDF(same weak no.)	HDF(2 pose)
処理速度	30ms	49ms	42ms

5.5 まとめ

本章では，追跡対象の形状変化に対して，適応的に弱識別器の形状を変えることができる SDF を用いたオフライン学習およびオンライン学習の枠組みについて提案した．提案手法は，オンラインで追跡対象の姿勢変化に合わせて弱識別器の形状を変形させることで，追跡対象の姿勢変化に対してロバストに追跡することができた．これにより，従来の HDF による追跡と比較して，姿勢変化が伴うシーンでの追跡性能を向上させることができた．また，SDF は，変形パラメータの設定範囲によって追跡対象の拡大・縮小や回転へ対応することが可能な弱識別器セットをグルーピングすることも可能である．

第6章

Boosted Randomized Treesによる人体検出と行動の同時認識

従来の行動認識は，背景差分や人検出などを用いて、人の領域を抽出した後，人の領域の動き情報を捉えて行動を認識する．そのため，人の領域を正しく検出できていないと，行動認識の精度が低下する．そこで，我々は人体を検出すると同時に姿勢の推定および行動の認識を 1 つのフレームワークで実現可能な Boosted Randomized Trees を提案する．提案手法は，Randomized Trees のノード生成時に，クラス尤度をもとにした階層的な認識クラスの定義方法および，Joint Boosting による事前特徴選択を導入することで，認識に有効な特徴を効率的に選択することが可能である．これにより，上位ノードは人体検出，下位ノードは姿勢推定または行動認識を階層的に行うことができる．評価実験では，従来の Randomized Trees に比べて，人体検出率および行動認識率が向上していることを示す．

6.1 行動認識について

人の行動を理解するための重要な要素技術として，人体検出手法，姿勢推定手法，行動認識手法がある．図 6.1 のように，1 枚の画像のみ与えられた場合，(a) および (b) は，人が歩いている，または走っていることが認識できる．一方，図 6.1(c) は人がかがみこんでいることが 1 枚の画像から認識することができるが，しゃがもうとしているのか，起き上がろうとしているのかまでは認識することができない．しかしながら，図 6.2 のように，3 枚の動画像の場合，起き上がろうとしていることが認識することができる．行動認識の過程において，人は人体の検出から行動の認識を静止画または動画像から得られる情報をもとに，より人の詳細な変化をとらえていると考えられる．すなわち，図 6.3(b) に示すように，人は人体の検出を行い，検出した人体の姿勢を推定，行動の認識を階層的に行っている．

図 6.1: 静止画における人の行動

図 6.2: 動画像における人の行動

そこで，我々は，図 6.3(b) に示すような，人体検出から行動認識までを 1 つのフレームワークで実現する．本フレームワークを用いることで，静止画を与えた場合は，姿勢推定および単純な行動

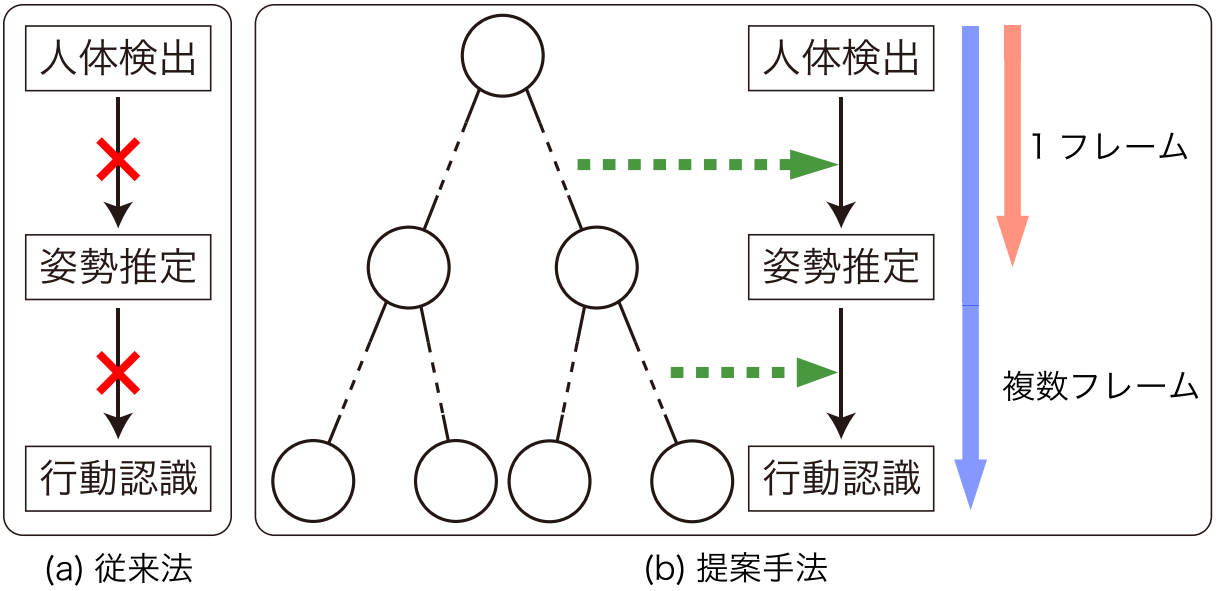


図 6.3: 提案手法と従来法の行動認識フレームワーク

が認識でき，動画像を与えた場合は，より複雑な行動を認識することができる．我々は，このようなフレームワークを実現するための階層的な構造として，Boosted Randomized Trees を提案する．提案手法は，複数の姿勢の人体を検出するために，マルチクラスの識別手法として注目されている Randomized Trees[42] をベースとしている．Randomized Trees は，各決定木の生成時にランダム選択を用いているため，有効な特徴が選択されない場合が生じる．提案手法は，クラス尤度をもとにした認識クラスの定義方法および Joint Boosting[43] による事前特徴選択を導入することで，効率的な特徴選択を実現する．これにより，従来の Randomized Trees の課題を解決することができる．

6.2 従来の行動認識手法について

人の行動を理解するための人体検出と行動認識は，ほとんどの場合が独立して研究されている．人体検出は，Dalal らが勾配方向ヒストグラムを利用した HOG 特徴量と SVM による手法 [20]，Zhu らは HOG 特徴量と AdaBoost による高速な手法 [26] を提案している．Hou らは HOG 特徴量を簡略化した EHOG と木構造の Vector Boosting による手法 [27] により，人体の検出と直立した人体の向きを推定している．また，Gall ら [44] は局所特徴量と Randomized Trees を用いて局所特徴と重心の位置関係を抽出し，Hough 変換により人の検出する手法を提案している．これらの手法は，直立した人体を前提としており，直立姿勢以外の姿勢を検出することができない．

姿勢推定は，シルエット画像のサンプルベースによる手法や人体の手や足などのパーツの関連性をもとにした手法などが提案されている．サンプル画像を利用した代表的な手法として，Shakhnarovich[45]

の手法がある．これは大量のシルエット画像のサンプルからハッシュを利用した最近傍探索による類似画像探索をベースとした姿勢推定である．Fathi らは，動画像における人の動きや見えの変化から人の姿勢を推定している [46]．これらは，大量のサンプル画像が必要となり，それらはあらかじめ正規化しておかなければならず，その切り出し精度により性能が大きく左右される．一方，Bissacco[47] の見えと動きを利用した手足のパーツベースの手法も同様に，人体の位置情報を必要としており，その位置精度が性能を左右する．このように，姿勢を推定するためには，人体の位置があらかじめ分かっていることが前提である．

行動認識は，姿勢の変化に着目した手法やフレーム間での動きや見えの変化をもとにした手法が提案されている．Blank らは，局所領域での時空間変化をとらえる ST-Patch を用いた手法 [11]，Niebles らは動画像における動きや見えの変化を階層的なモデルに利用した手法 [48] を提案している．提案されている多くの手法は，フレーム間での変化をもとにしている．しかしながら，人の行動によっては，1 フレームにおける姿勢から大まかな行動を推定することができる．Schindler ら [49] は，行動認識を行うために必要なフレーム数について実験しており，単純な行動の場合，1 フレームで認識できている．また，その他の行動については，3 から 5 フレームで行動を認識できると報告されている．

人体の検出から姿勢推定までを 1 つのフレームワークで実現している手法もある [50][51]．Bissacco ら [51] は，直立した人体における詳細な姿勢をとらえることを対象とし，サンプル画像に類似する姿勢を推定する方法を提案している．Rogez ら [50] の手法は，提案手法と同様に Randomized Trees を利用した姿勢推定を行っているが，直立姿勢における詳細な手足の位置を推定しスケルトンモデルとして姿勢を推定している．人の行動を認識する場合，これらのような詳細な姿勢情報は必要とせず，大まかな姿勢が分かれば行動を認識することができる．これらの手法は姿勢を推定するのみであり，行動を認識することはできない．

一方で，提案手法のように階層的な構造にブースティングを導入した方法がある [52]．これは，ブースティングで学習した強識別器を木構造のノードとして扱うため，1 つの木で多クラス認識が可能であるが，過学習が生じやすい．提案手法は，過学習を防ぐために，ノードは 1 つの弱識別器をブースティングにより学習している．これにより，木構造のランダム性を維持したまま，より識別に有効な特徴量をブースティングで選択することが可能である．また，木構造を利用して階層的に局所領域の多クラス識別と物体の位置推定を行う手法がある [44][53]．これは，物体の局所領域を抽出し，その局所領域が属するクラスを識別した後，複数のクラスの位置関係から物体の位置を推定する．これらの手法は，局所領域から向きや姿勢などを推定した後，その推定結果をもとに物体の位置を推定しているといえる．提案手法は，物体を 1 つの領域として捉えるため，位置の推定をする必要はなく，1 つのフレームワークで位置推定と同時に行動を認識することができる．

6.3 提案手法

Randomized Trees を構成する決定木は，階層的な構造であり，上位ノードは複数のクラスに共通する特徴，下位ノードは特定のクラスに対応する特徴を選ぶ必要がある．また，人体の検出から行動認識を行うためには，上位ノードは複数の姿勢と背景を識別するのに有効な特徴，下位ノードは

特定の姿勢を認識するのに有効な特徴を選択する必要がある．しかしながら，ランダムに特徴候補を用意するため，有効な特徴がその中に含まれているとは限らない．そこで，我々は各ノード生成において，認識対象クラスに適した特徴を効率的に特徴選択する Boosted Randomized Trees を提案する．Boosted Randomized Trees は，ノード生成時に認識対象クラスをクラス尤度をもとに絞り込み，そのクラス集合に適した特徴を Joint Boosintg により選択する．これにより，上位ノードでは複数クラスに対応した特徴，下位ノードでは特定クラスに有効な特徴を選択することが可能である．

本章では，従来の Randomized Trees および Joint Boosting について述べた後，提案する Boosted Randomized Trees について述べる．

6.3.1 Randomized Trees

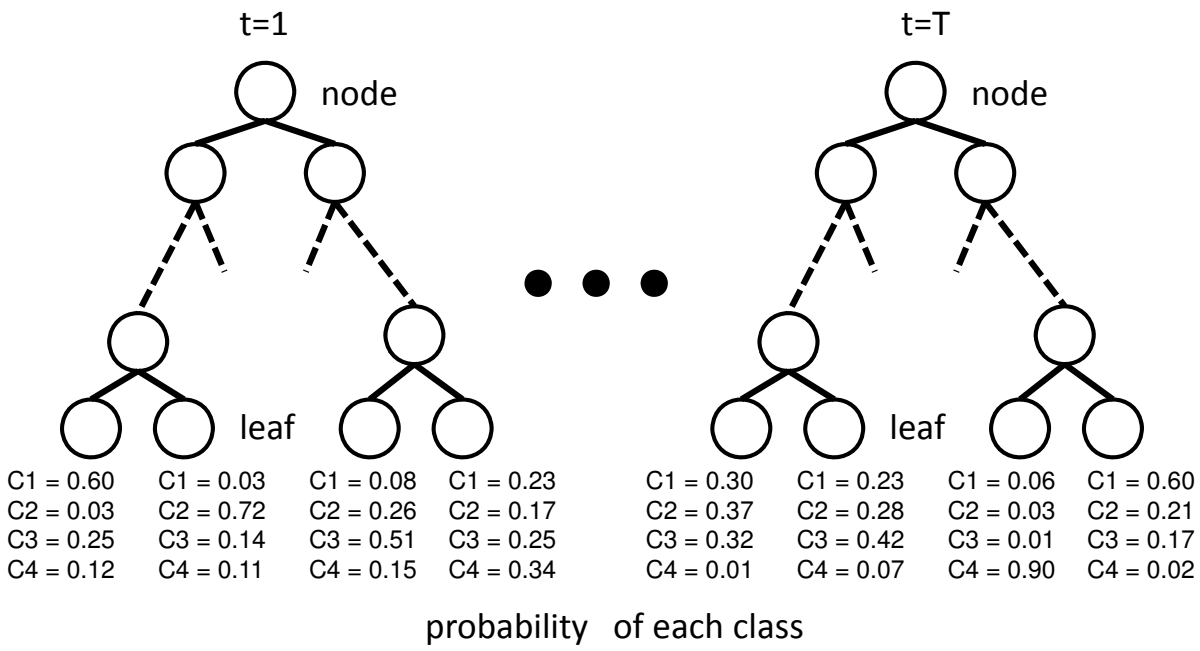


図 6.4: Randomized Trees の構造

Randomized Trees[42] は，マルチクラスの認識に対応したアンサンブル学習手法の一つであり，キーポイントの検出 [54] やセグメンテーション [55] など様々な分野に応用されている．Randomized Trees は，学習サンプルに含まれるノイズに対してロバスト性があり，認識対象によっては，Boosting よりも精度が高い場合がある．また，Randomized Trees を構成する各決定木は独立しており，学習および評価の処理を決定木ごとに並列に行うことができるため高速である．

図 6.4 に Randomized Trees の構成を示す．図 6.4 のように，複数の決定木 T から構成されている．各決定木には，分岐ノードと木の終端のリーフがある．C 個のクラスを認識する場合，各リーフは，各クラス $c = (1, 2, ..., C)$ に対する確率分布を有する．また，決定木の各ノードは，スプリット関数により分岐処理を行う．スプリット関数は，式 (6.1) のように，特徴量 $I(x)$ が閾値 θ より小さい場合は左側の子ノード，大きい場合は右側の子ノードに進む．

$$I(x) = \begin{cases} < \theta & go\ left\ child\ node \\ \geq \theta & go\ right\ child\ node \end{cases} \tag{6.1}$$

Randomized Trees の学習は，サブセットの作成，ノードの生成，サブセットの分割の 3 つのプロセスからなる．

まず，各決定木を学習するために，学習サンプル $X = \{x_i, c_j; i \in [1, N], j \in [1, C]\}$ からサブセット X_s を作成する．サブセットは，ランダムに選択された S 枚のサンプル画像からなる．

次に，ノードを生成する．ノードは，スプリット関数とその関数に用いる特徴と閾値から構成される．ノードの生成では，スプリット関数に用いる特徴と閾値を決める．特徴および閾値の選択は，Extremely Randomized Trees のようにランダムに用意された候補の中から最適な組み合わせを選択する方法 [56] とノード生成時に用いる学習サンプルに対して，最もエラー率が小さくなる特徴を選択する方法 [50] がある．ここでは，Extremely Randomized Trees を用いた場合について説明する．

ランダムに用意された特徴 $f_m; m \in [1, M]$ に対して，閾値 $\theta_{m,k}; k \in [1, K]$ をランダムに用意する．すべての特徴に対して，ランダムに閾値を用意し，最も最適な組み合わせとなる特徴と閾値を選択する．最適なものを選択する際の選択基準として，式 (6.2) のような情報量を用いる．

$$\triangle E = -\frac{|I_l|}{|I|} E(I_l) - \frac{|I_r|}{|I|} E(I_r) \tag{6.2}$$

ここで，$E(I_l)$ および $E(I_r)$ はある特徴と閾値の組み合わせを用いた場合に，右側 r または左側 l に分岐した各クラスのサンプルに対するシャノン情報量である．シャノン情報量は式 (6.3) のように算出する．

$$E(I) = -\sum_{j=1}^{C} P(c_j) \log P(c_j) \tag{6.3}$$

$P(c_j)$ はノードにおけるクラス c_j の確率分布である．

サブセットの分割は，上記の特徴量選択により得られた特徴を用いて，式 (6.1) のように分割する．閾値以下の場合は左の子ノードを生成するためのサブセット，閾値以上の場合は右側の子ノードを生成するためのサブセットとなる．

上記の処理をそれぞれの子ノードに対して，分割した新たなサブセットを用いて繰り返し行う．到達した学習サンプルが一定枚数より少なくなった場合，1 つのクラスの学習サンプルのみの場合，または一定のノードの深さになった場合にノード生成の処理を打ち切る．終端ノードはリーフ l として，各クラスの確率分布 P(c) を持っている．クラス c_j の確率分布は式 (6.4) のように求める．

$$P(c_j|l) = \frac{|I_{c_j}|}{|I|} \tag{6.4}$$

アルゴリズム 1 Joint Boosting による学習アルゴリズム

初期化:

1. 学習サンプルの重み w_i^c を初期化

$For\ i = 1..N$ //サンプル数

  $For\ c = 1..C$ //クラス数

    学習サンプルの重み w_i^c を初期化

学習:

2. $m = 1, 2, .., M$ //選択する弱識別器数

  (a) $n = 1, 2, .., 2^C - 1$ //クラスの全組み合わせ回数

    (i) 全弱識別器候補に対するエラーを算出

  (b) 最小エラーとなるクラスの組合せ n と弱識別器候補 $h_m^n(v, c)$ を選択

  (c) 重み w_i^c を更新

3. 選択した弱識別器を組合せ n ごとに統合し，$G^{S(n)}(v)$ を得る

$|I|$ は全クラスのサンプル数，$|I_{c_j}|$ はクラス c_j のサンプル数である．

評価は，生成されたすべての決定木から得られる各クラスの確率を求める．入力された画像は各決定木の 1 つのリーフへ到達する．到達したリーフ $L = \{L_t; t \in [1, T]\}$ が持つ確率分布 $P(C|L_t)$ を式 (6.5) のようにクラスごとに累積して平均を求める．

$$P(C|L) = \frac{1}{T} \sum_{t=1}^{T} P(C|L_t) \tag{6.5}$$

式 (6.5) の平均確率が最大となるクラスを認識結果として出力する．

6.3.2 Joint Boosting

Joint Boosting[43] は，マルチクラスの Boosting であり，複数のクラスに共通する特徴を選択することができる．Joint Boosting は，式 (6.6) に示すように，全クラスの部分集合 $S(n)$ に対する強識別器を学習する．

$$G^{S(n)}(v) = \sum_{m=1}^{M} h_m^n(v) \tag{6.6}$$

ここで，$h_m^n(v)$ は m 番目の弱識別器であり，v は特徴ベクトルである．アルゴリズム 1 に Joint Boosting の処理手順を示す．Joint Boosting の学習は，正解クラスの組み合せを変えながら最適な弱識別器を選択する．すなわち，$2^C - 1$ の組み合わせの中からエラーが最小となるクラス集合 $S(n)$ に対応した弱識別器 $h_m^n(v)$ を得る．3 クラスの場合，組み合わせ数は 7 通りである．Joint Boosting では，対象とするクラス集合 $S(n)$ に共通する弱識別器を統合し，弱識別器の集合 $G^{S(n)}$ を得る．3 クラスを Joint Boosting で学習する場合の学習例を図 6.5 に示す．図 6.5 のクラス 1，クラス 2，クラス 3 は正解クラスであり，neg は非正解クラスである．Joint Boosting では，非正解クラスの学習

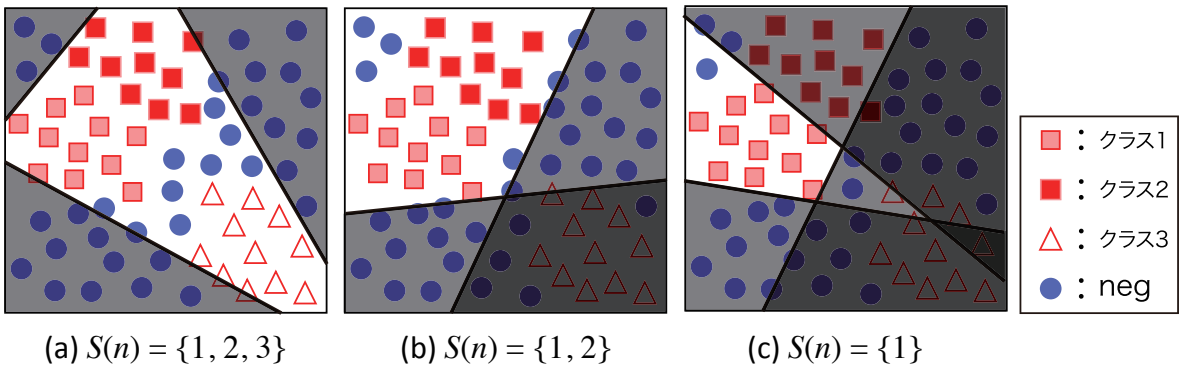


図 6.5: Joint Boosting による学習例

サンプルをあらかじめ別途用意している．$S(n) = \{1, 2, 3\}$ の場合，図 6.5(a) のようにすべてのクラスを正解クラスとするように弱識別器を学習することができる．一方，$S(n) = \{1, 2\}$ の場合は，図 6.5(b) クラス 1 とクラス 2 を正解するように弱識別器を学習する．また，$S(n) = \{1\}$ の場合は，図 6.5(c) のようにクラス 1 を正解するように弱識別器を学習することができる．

6.3.3 Boosted Randomized Trees

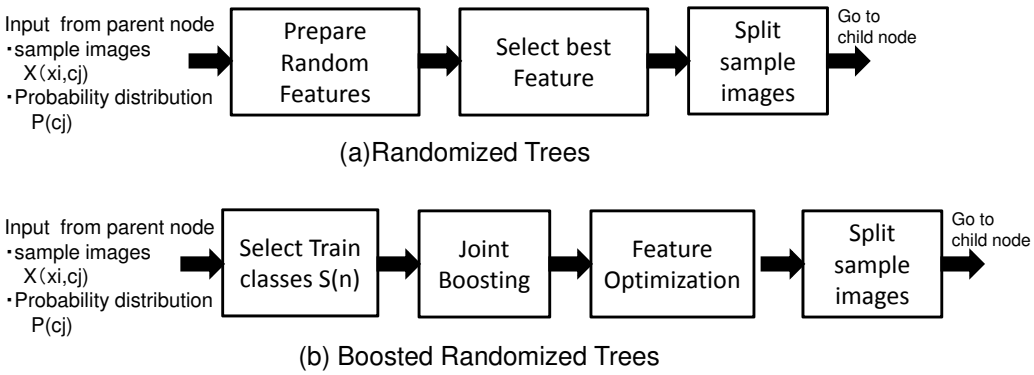


図 6.6: ノードの学習方法

従来の Randomized Trees および Boosted Randomized Trees におけるノードの生成方法を図 6.6 に示す．図 6.6(a) に示すように，従来の Randomized Trees のノード生成は，特徴と閾値をランダ

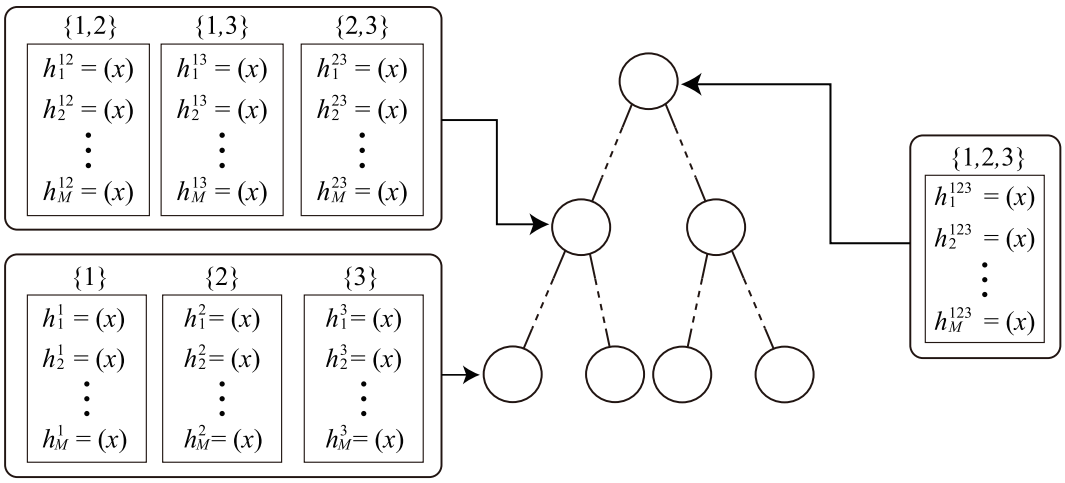


図 6.7: Joint Boosting による事前特徴選択

ムに用意する処理，情報利得が最大となる特徴および閾値の選択，サンプル画像の評価と子ノードのサブセット作成の 3 つのプロセスからなる．一方，提案する Boosted Randomized Trees は，クラス集合の定義，Joint Boosting による事前特徴選択，特徴の最適化，子ノードのサブセット作成の 4 つのプロセスから構成される．ここでは，提案手法の各プロセスについて述べる．

■ クラス集合の定義

Joint Boosting は，クラスの全組み合わせの中から特定のクラス集合に対する弱識別器を選択する．そのため，クラス集合をあらかじめ決めることができない．Boosted Randomized Trees を構成する決定木は階層的な構造であり，上位ノードは複数のクラス，下位ノードは特定のクラスに対応している．また，人体検出から行動認識までを階層的に行う場合，上位ノードは人体と背景，下位ノードは特定の姿勢または行動を区別する特徴を選択することができれば，効率的な決定木を学習することができる．すなわち，上位ノードは複数のクラスのクラス集合，下位ノードは特定のクラスのクラス集合に対する弱識別器を学習する方がよい．そこで，ノードにおける最適なクラス集合をクラス尤度をもとに定義する．各クラスの尤度は式 (6.4) で示したように，各クラスに対する確率から求める．あるノードにおいて，クラスの組合せのうち，式 (6.7) のようにクラス尤度の合計が閾値 τ より大きくなる組み合わせをクラス集合 $S(n)$ として定義する．

$$S(n) = id \in \mathcal{I} : L(id) > \tau \tag{6.7}$$

id は組み合わせの番号，$\mathcal{I}$ は全組み合わせ，$L(id)$ はクラス尤度の合計，τ はしきい値を表わす．その際，クラス数が最も少なくなる組合せを選ぶ．上位ノードは，多数のクラスが存在しており，各クラスの尤度は低くなる．そのため，多くのクラスを組み合わせたクラス集合を定義することがで

きる．一方，下位ノードは，特定のクラスに偏っているため，特定クラスの尤度が高くなる．そのため，尤度の低いクラスを認識対象としてみなす必要がなく，クラス集合は特定クラスのみになる．これにより，特定のクラスの組合せにのみ Joint Boosting により学習することができる．

■ Joint Boosting による事前特徴選択

3.2 節で述べた Joint Boosting は，与えられたクラスの全組合せの中から最適な特徴を選択する．そのため，上位ノードにおいて，複数のクラスに共通した特徴が選択されず，特定のクラスにのみ特化した特徴が選択される場合がある．そこで，提案手法では，3.3 節で述べたクラス尤度をもとにしたクラス集合を利用して，クラスの組合せを限定する．すなわち，アルゴリズム 1 で示す学習手順において，ループ (a) を全組み合わせでなく，限定した組合せにのみ行う．図 6.7 に示すように，多くのクラスを含む上位ノードでは，クラス集合に多くのクラスが含まれており，それに共通する特徴が選択される ($S(n) = \{1, 2, 3\}$)．一方，下位ノードは，特定のクラスのみクラス集合に含まれているため，特定のクラスにのみ適した特徴が選択される ($S(n) = \{1\}$)．このように，クラス尤度を用いたクラス集合を用いて，Joint Boosting の特徴選択方法を限定することで，階層的な特徴を効率的に選択することが可能である．

■ 特徴量

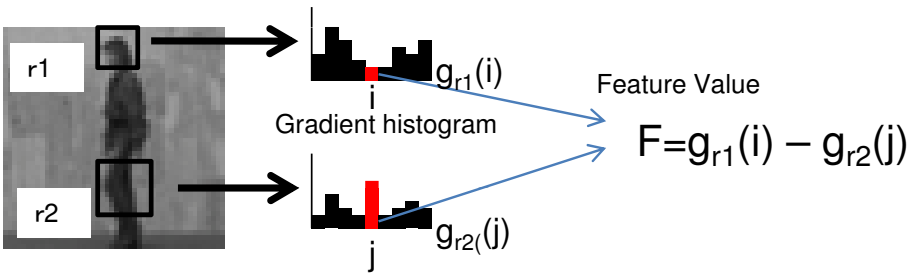


図 6.8: 本手法に用いる特徴量

本稿では，人体検出に有効な勾配方向ヒストグラムをベースとした特徴量を用いる．特徴量の概要を図 6.8 に示す．特徴量は式 (6.8) に示すように，2 つの局所領域 r_1,R_2 における勾配方向ヒストグラムの 1 つのビンの値 $g_{r1}(i)$，$g_{r2}(j)$ の差である．本稿では勾配方向数を 9 としている．

$$F = g_{r1}(i) - g_{r2}(j) \tag{6.8}$$

ここで，i および j は勾配ヒストグラムのビンの位置である．勾配ヒストグラムの差をとらえることで，局所領域におけるエッジの関連性をとらえることができる．

■ 特徴量の最適化

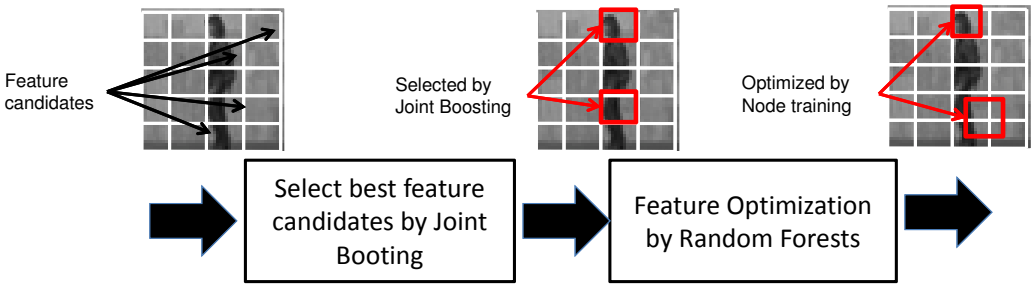


図 6.9: 特徴量の最適化方法

本稿で用いる特徴量は，２つの領域間における差に着目しており，領域の組み合わせは非常に膨大である．そこで，本稿では，図 6.9 に示すような Joint Boosting での事前特徴選択とノード内における特徴最適化の 2 段階の特徴選択を行う．まず，Joint Boosting で選択する特徴候補は，領域の位置および大きさを一定の間隔でサンプリングされた領域を特徴候補として与える．Joint Boosting では，これらの領域の組み合わせのうち，最適な２つの領域の組み合わせを特徴として選択する．次に，各領域をサンプリング間隔の範囲で位置および大きさをずらし，その中で最適な組み合わせを最終的な特徴として選択する．スプリット関数の閾値は，Joint Boosting により得られた閾値をもとに，新たな組み合わせに対して最適な閾値を求める．特徴および閾値は，式 (6.3) を用いて，情報利得が最大となるものを選択する．

6.3.4　特徴量の行動認識への拡張

フレーム間での特徴の変化は，行動認識において重要な特徴である．そこで，行動認識を行うために，特徴量を図 6.10 のように，複数フレーム間における勾配方向ヒストグラムの差を捉えることができるように拡張する．

$$F = g_{r1,t1}(i) - g_{r2,t2}(j) \tag{6.9}$$

ここで，t_1 および t_2 は着目するフレームである．このように複数フレーム間に拡張することで，同一フレーム内における勾配方向の違いをとらえるだけでなく，異なるフレーム間での勾配方向の変化をとらえることができる．

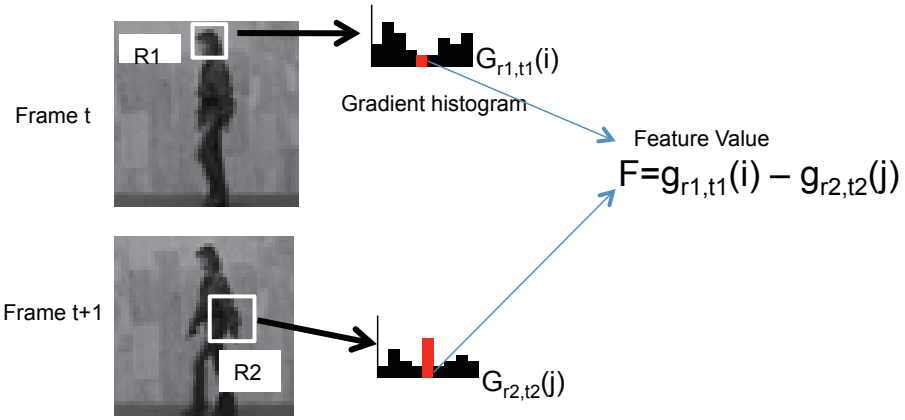


図 6.10: 動画像に拡張した特徴量

6.4 実験

6.4.1 実験概要

本提案手法の有効性を示すために，人体検出率と行動認識率の性能を Joint Boosting および Randomized Trees を用いた場合との比較実験を行う．入力として 1 枚の静止画を与えた場合，動き情報はとらえないため，行動認識は大まかな姿勢推定とみなすことができる．本稿では，1 枚の静止画を入力した場合の行動認識を姿勢推定としてみなし，実験を行う．提案手法は，直立した人体だけでなく様々な姿勢の人体検出から行動認識までを 1 つのフレームワークで行うことができる．そこで，姿勢変化を含む評価データセットとして，行動認識の研究で一般的に用いられるデータセット [11] を用いる．評価データセットは，行動ごとに 9 名分の被験者の動画像から構成されている．本稿では，姿勢の異なる行動シーンにおける人体検出および姿勢推定の評価を行うために，図 6.11 に示すような“walk”，“bend”，“wave”の 3 つの行動シーンを用いる．行動認識は，“bend”および“wave”を構成する行動を細分化して認識する．“bend”はしゃがむ、起き上がるの 2 つのどちらであるかを認識する．“wave”は手を挙げるまたは手を下げるのどちらであるかを認識する．評価は 9 人のうち 1 人を評価データとし，残り 8 人を学習データとして用いる．これらの評価をすべての人物に対して行い，平均精度を求めた．各動画像のフレーム数は一定ではないため， 評価対象フレームはフレーム内に人物が映っている範囲のうち 30 フレームとした．人体の検出は，スライディングウィンドウ方式により網羅的に探索する．そして，式 (6.5) から得られるクラスの平均確率が閾値より大きい領域を人体領域として出力する．本実験では，閾値を 7.5 としている．

(a) walk

(b) bend

(c) wave

図 6.11: 実験画像例

6.4.2 Boosted Randomized Trees の学習

Randomized Trees および Boosted Randomized Trees で学習する決定木数は 10 とし，決定木の最大深さは 15 階層とする．Randomized Trees のノード生成時に用意する特徴候補はランダムに 100 個用意し，各候補に対して 100 個の閾値を同様にランダムに用意する．これらの組み合わせの中から最適なノードを選択する．一方，Boosted Randomized Trees は Joint Boosting により 10 個の特徴候補を事前学習する．Joint Boosting に与える特徴候補は，位置を 4 ピクセル間隔でサンプリングして限定している．Joint Boositng により得られた各候補に対して，位置を 4 サンプリングで間引かれた 4 ピクセルの範囲で変えて最適な組み合わせを選択する．各決定木の学習には，48x48 ピクセルに正規化した各姿勢のサンプル画像を 240 枚用いる．サンプル画像は全サンプル画像からランダムに選択する．非人体画像は，背景データセットから 1 万枚の正規化した画像を用意してランダムに 2000 枚を選択する．行動認識の学習は，3 フレームを 1 つのサンプル動画像とし，ランダムに動画像を 240 セット選択する．決定木数およびその他の設定は，静止画における実験と同様である．

6.4.3 実験結果

■ 人体検出結果

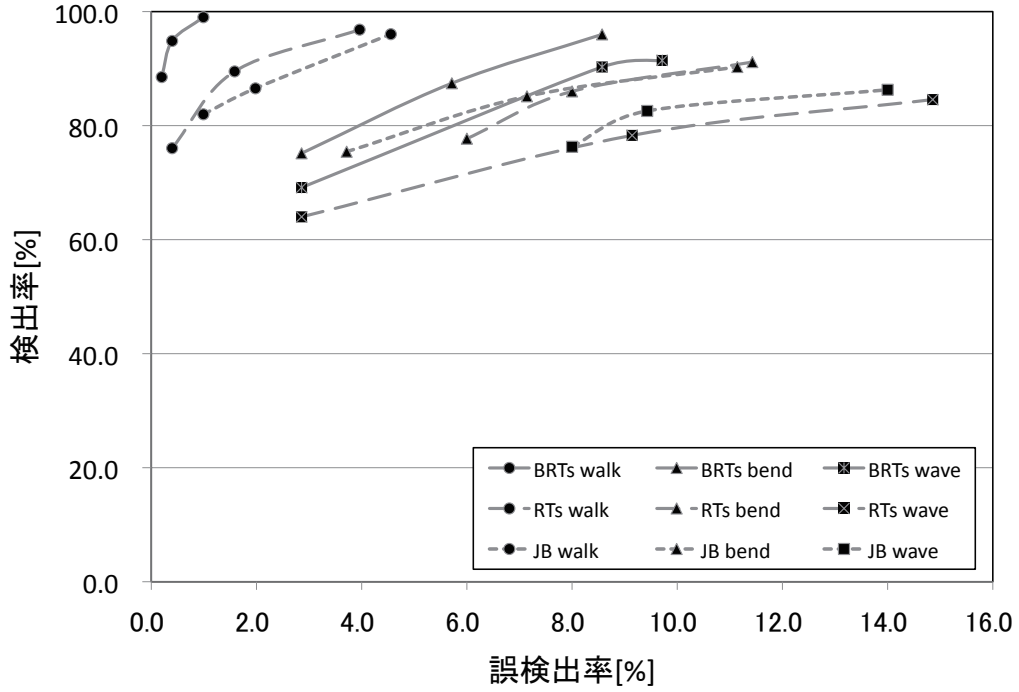


図 6.12: 人体検出率の ROC カーブ

図 6.12 に Joint Boosting(JB)，Randomized Trees(RTs) および Boosted Randomized Trees(BRTs) の人体検出率を示す．また，図 6.13 に各姿勢における提案手法の検出結果を示す．図 6.13 の (a) は walk データセット，(b) は bend データセット，(c) は wave データセットにおける検出結果である．Boosted Randomized Trees は，Joint Boosting および Randomized Trees を用いた場合に比べて，全姿勢において検出率が向上していることが分かる．walk データセットは直立姿勢の人体であり，体型による個人差はあるものの姿勢のバリエーションが少ないため，検出率は高い．そのため，すべての手法の検出率に大きな差はない．一方，bend はしゃがみこむ角度により様々な形状があり，姿勢のバリエーションが多い．wave も同様に，手の挙げ方や挙げている行動過程における手の角度が様々であり，バリエーションが多い．Joint Boosting はクラス間またはクラス内に共通するバリエーションが少ない特徴をとらえる．また，クラスの組み合わせは事前に定義できないため，特定のクラスに偏る場合もある．そのため，これらの姿勢における検出率が低下している．Randomized Trees はランダムに特徴を選択するため，これらの姿勢をとらえるのに有効な特徴が選択されない場合が多い．一方，提案手法は，ノードに最適なクラスの組み合わせを定義するため，複数のクラス

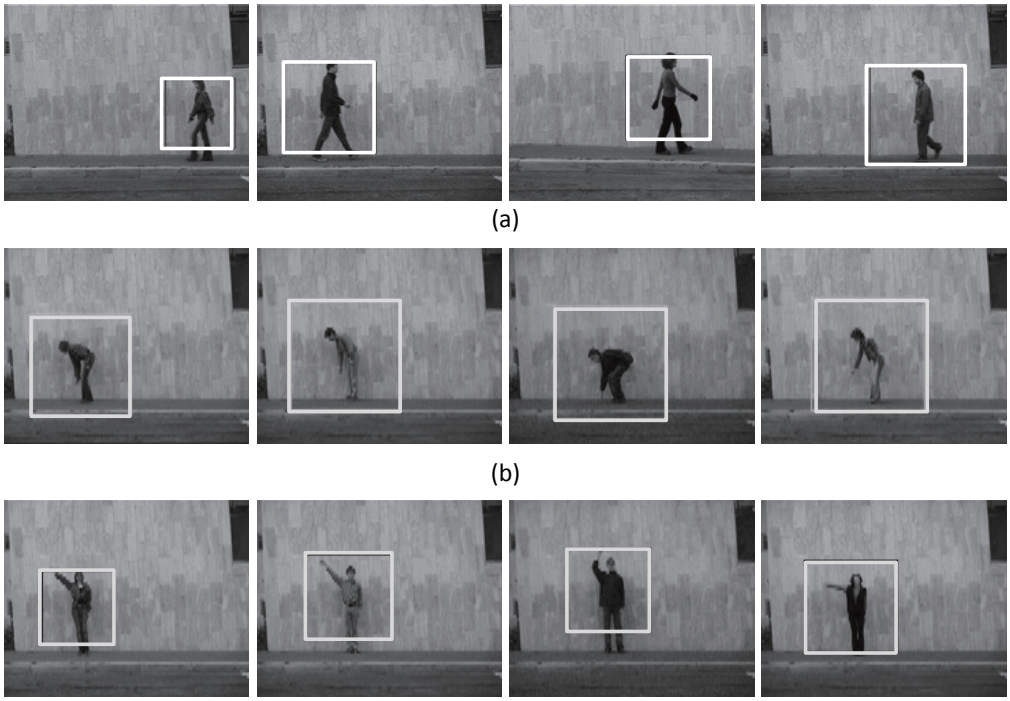


図 6.13: 人体検出結果例

に共通する有効な特徴を事前に選択することができる．また，木構造のため，同一クラスに対して複数のリーフが存在する．そのため，クラス内のバリエーションに対応するリーフを生成することができる．これにより，提案手法は bend および wave データセットにおいて，Joint Boosting および Randomized Trees より検出率を向上させることができている．

■ 静止画における行動認識結果

前節での実験結果に対して，検出結果がどの行動クラスに属していたかを調べた．静止画における行動認識は，大まかな姿勢推定とみなすことができる．表 6.1 は，検出した人体に対する姿勢推定率であり，かっこ内は，全評価画像に対する姿勢推定率である．walk データセットにおいて，検出された人体に対する姿勢推定率は Joint Boosting および Randomize Trees と差はない．しかし，bend および wave データセットでは，提案手法は Joint Boosting および Randomized Trees に比べて，姿勢推定率は向上している．これより，クラス内でのバリエーションが多い各姿勢に対して有効な特徴が選択することができていることが分かる．

表 6.1: データセットごとの姿勢推定率

dataset	JB[%]	RTs[%]	BRTs[%]
walk	98.7(95.6)	98.1(96.4)	98.4(97.6)
bend	88.3(77.7)	91.3(71.6)	94.5(84.9)
wave	68.3(61.5)	65.3(55.0)	70.0(63.2)
average	85.1(78.3)	84.9(74.3)	87.6(81.9)

■ 動画像における行動認識結果

動画像に対する行動認識性能の比較実験を行った．その結果を表 6.2 および表 6.3 に示す．Boosted Random Trees は Randomized Trees を用いた場合に比べて，行動認識率が向上していることがわかる．従来法は，誤った行動として認識するケースがあるが，提案手法は，行動認識ができなかったものは検出できない場合であり，誤認識がない．提案手法は，動画像において，フレーム間の変化をとらえる特徴を事前に効率的に選択できているため，認識精度向上につながったといえる．これより，提案手法は動画像を入力とした場合に，人体検出から行動認識までを 1 つのフレームワークで実現することができている．

表 6.2: RTs の行動認識精度

	walk	bend-up	bend-down	Wave-up	Wave-down	False Negative
walk	95.5			2.0		2.5
bend-up		91.1	1.0	3.2		4.7
Bend-down		2.1	89.5		3.0	5.4
wave-up		5.5		80.1	1.3	13.1
wave-down			4.2	1.8	84.7	9.3
average	89.4%					

表 6.3: BRTs の行動認識精度

	walk	bend-up	bend-down	Wave-up	Wave-down	False Negative
walk	99.5					0.5
bend-up		95.1				4.9
Bend-down			95.6			4.4
wave-up				85.1		14.9
wave-down					90.3	9.7
average	93.9%					

6.5 考察

クラス尤度をもとにしたクラス集合の定義と Joint Boosting による事前特徴選択の有効性を示すために，各ノードにおけるクラスの分割結果を図 6.14 に示す．図 6.14 のルートノードでは，Joint Boosting により 3 つの姿勢に共通する足元をとらえる特徴が選択されている (a)．ルートノードで選択された特徴により学習サンプルは 2 つのサブセットに分割される (b)．分割された一方は bend と背景クラス，もう一方は walk と wave を主に含んでいる．ルートノードにおいて，形状の類似している walk と wave，それ以外のクラスに分けることができている．1 階層目の左側のノード (c) では，

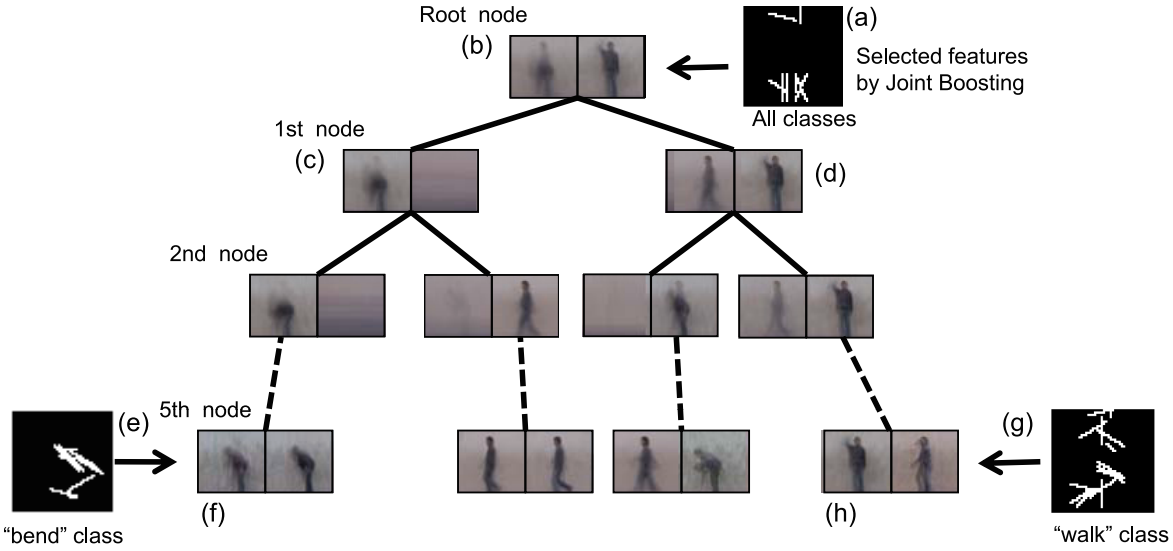


図 6.14: 各ノードにおける事前選択特徴とクラス分割結果例

bend と背景クラス，右側のノード (d) では walk と wave を識別しており，2 層目では，各クラスと背景を主に識別している．このように，最初の数層で様々な姿勢の人体と背景クラスを識別することができており，人体検出に相当している．また，各姿勢ごとに分割することもできており，人体検出と同時に大まかな姿勢推定も行うことができている．さらに，学習を数層行うことで，より詳細な行動クラスに分割している．(f) では bend を多く含むノードになっており，bend クラスに共通するしゃがんだ背中付近に相当する特徴が Joint Boosting により事前学習されている (e)．(h) では walk を多く含むノードになっており，walk に共通する手や足の位置に相当する特徴が事前学習されており (g)，walk と wave を識別している．このように，下位ノードでは各行動クラスを認識するための特徴が多く選択されている．

提案手法で導入した事前特徴選択により，上位ノードは多数のクラスに共通する特徴，下位ノードは特定のクラスに対応する特徴を階層的に選択することができている．これにより，1 つのフレームワークで階層的に人体検出から姿勢推定，行動認識を実現している．

6.6 まとめ

本章は，人の行動を理解するために，人体検出から行動認識までを 1 つのフレームワークで実現する Boosted Randomized Trees を提案した．提案手法は，決定木のノード生成時に各クラスの尤度をもとに認識対象となるクラスを定義し，Joint Boosting により効率的に特徴を選択することで，上位ノードは多数のクラスに対応した特徴，下位ノードは特定のクラスに対応した特徴を階層的に選

択することが可能である．これにより，Boosted Randomized Trees を用いることで，Randomized Trees より行動認識性能を向上させることができる．

第7章

結論

本論文では，カメラ映像からの人の行動理解を実現するために，各要素技術について新たな枠組みを提案した．各章のまとめは次の通りである．2章では，物体検出に用いられる特徴量とその学習方法について述べた．検出対象ごとに適した特徴量について調査した結果，顔検出は輝度に着目する特徴量，人検出はエッジに着目する特徴量が有効であることが分かった．

3章では，距離情報による統合処理を導入した物体検出の高精度化について提案した．人の重なりが生じるシーンにおいて，人を高精度に検出するためには，パーツベースの方法が有効である．パーツベースの手法の場合，パーツ間の統合を高精度に行う必要がある．そこで，提案手法は，パーツベースの検出器として，顔検出と人検出を用い，検出結果の統合に mean shift 法をベースとした距離情報を用いた統合手法を導入している．比較実験では，距離情報を用いない場合に比べて，高い検出率を得ることができている．提案手法を用いることで，混雑環境下においても異なる距離に存在する検出結果を誤って統合することなく，高精度に人物の検出が可能である．

4章では，物体検出を応用したオンライン学習による物体追跡手法を提案した．提案手法は，物体らしさを表す確率密度関数をオンライン学習に導入することで，追跡対象への過適合を抑制し，顔の向変化による形状変化や照明変化が生じる場合でも途切れなく追跡することができる．比較実験から従来のオンライン学習手法や Mean Shift 法と比較して，追跡精度が向上していることを確認した．また，顔だけでなく，人体へも応用可能なことを示した．これより，提案手法は，顔向き変化が生じるシーンや照明変化が生じるシーンにおいて，顔や人体の追跡を高精度に行うことが可能である．

5章では，検出器を応用したオンライン学習による物体追跡において，追跡対象の形状変化に追従するための弱識別器の構造について提案した．従来の検出器を用いたオンライン学習は，オフラインで学習した弱識別器から最適なものをオンラインで選択する．そのため，オフライン学習時に様々な形状を考慮しておく必要がある．しかしながら，人体のように形状変化が生じる物体の様々な形状に対応した弱識別器を用意することは困難である．そこで，提案手法では，1つの弱識別器が複数の形状に対応可能な弱識別器の構造を導入した．提案する弱識別器は，物体らしさを表す確率密度関数が類似する形状を1つのグループとしてグルーピングする．追跡時に，弱識別器の選択を行うとともに，最適な形状の選択も行う．比較実験において，弱識別器が1つの形状しか持たない場合と比べて，追跡精度が向上していることを確認した．これにより，提案手法は，追跡対象の形状が変化する場合でも，追跡を高精度に行うことが可能である．

6章では，物体検出と同時に行動認識が可能なフレームワークについて提案した．従来の行動認識は，物体検出で得られた領域を正規化し，行動認識手法により識別する．そのため，検出の位置精

度が性能に大きく影響する．また，時系列データでのフレーム間の動き変化をもとにする手法の場合，背景が同時に動くと認識対象と背景の動きの区別が困難なため，特定の環境下でしか認識できない．そこで，提案手法では検出結果の位置精度に依存せず、複雑背景下でも認識可能な行動認識手法を提案した．提案手法は，Randomized Trees のノード生成時に，クラス尤度をもとにした階層的な認識クラスの定義方法および，Joint Boosting による事前特徴選択を導入することで，認識に有効な特徴を効率的に選択することが可能である．これにより，上位ノードは人体検出，下位ノードは姿勢推定または行動認識を階層的に行うことができる．また，提案手法は，人体を検出すると同時に姿勢の推定および行動の認識を 1 つのフレームワークで実現している．比較実験において，提案手法は，従来の行動認識手法と比較して，認識精度が向上していることを確認した．これにより，提案手法は，物体の検出とともに行動を高精度に認識することが可能である．

謝　辞

本研究を行うにあたり，終始懇切なご指導を賜りました中部大学工学部情報工学科 藤吉弘亘教授に深く感謝の意を表します.

本論文をまとめるにあたり，有益なご討論，ご助言を賜りました中部大学工学部情報工学科　岩堀祐之教授，平田豊教授，中野良平教授に謹んで深謝します.

本論文をまとめるにあたり，有益なご助言を賜りました名古屋工業大学院産業戦略工学専攻 梅崎太造教授に心から厚く御礼申し上げます.

最後に，本研究で用いるデータ撮影やディスカッションなど協力して頂いた藤吉研究室の皆様に感謝致します.

参考文献

[1] ObjectVideo. http://www.objectvideo.com/.

[2] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa: "A System for Video Surveillance and Monitoring", In *technical report CMU-RI-TR-00-12*, 2000.

[3] 細井聖, 川出雅人, 森本勝: "徘徊者保護支援システム–歩行中の人物を顔認識する技術について", *Omron technics*, Vol. 39, No. 2, pp. 108–113, 1999.

[4] 三菱重工. http://www.mhi.co.jp/products/detail/wakamaru.html.

[5] A. Mohan, C. Papageorrgiou, and T. Poggio: "Example-Based Object Detection in Images by Components", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 4, pp. 349–361, 2001.

[6] K. Mikolajczyk, C. Schmid, and A. Zisserman: "Human Detection Based on a Probabilistic Assembly of Robust Part Detectors", In *European Conference on Computer Vision*, pp. 69–81, 2004.

[7] B. Wu and R. Nevatia: "Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Based Part Detectors", *International Journal of Computer Vision*, Vol. 75, pp. 247–266, 2007.

[8] S. Avidan: "Ensemble Tracking", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 2, pp. 261–271, 2007.

[9] H. Grabner and H.Bischof: "Online bagging and boosting", In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 260–267, 2006.

[10] A. A. Efros, A. C. Berg, G. Mori, and J. Malik: "Recognizing Action at a Distance", In *IEEE International Conference on Computer Vision*, Vol. 2, pp. 726–733, 2003.

[11] M. Blank, L. Gorelick, E. Shechtman, and a. R. M. Irani: "Actions as space-time shapes", In *IEEE International Conference on Computer Vision*, Vol. 2, pp. 1395–1402, 2005.

[12] T. Sakai, M. Nagao, and S. Fujibayashi: “Line Extraction and Pattern Recognition in a Photograph”, *Pattern Recognition*, Vol. 1, No. 3, pp. 233–236, 1969.

[13] D. Maio and D. Maltoni: “Real-time Face Location on Gray-Scale Static Images”, *Pattern Recognition*, Vol. 33, No. 9, pp. 1525–1539, 2000.

[14] P. Viola and M. Jones: “Robust Real-Time Face Detection”, *International Journal of Computer Vision*, Vol. 57, No. 2, pp. 137–154, 2004.

[15] T. K. Leung, M. C. Burl, and P. Perona: “Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching”, In *IEEE International Conference on Computer Vision*, pp. 637–644, 1995.

[16] H. A. Rowley, S. Balauja, and T. Kanade: “Neural Network-Based Face Detection”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, pp. 23–38, 1998.

[17] C. Papageorgiou and T. Poggio: “A Trainable System for Object Detection”, *International Journal of Computer Vision*, Vol. 38, No. 1, pp. 15–33, 2000.

[18] 佐部浩太郎, 日台健一: “ピクセル差分特徴を用いた実時間任意姿勢顔検出器の学習”, 画像センシングシンポジウム (SSII), pp. 547–552, 2004.

[19] K. Levi and Y. Weiss: “Learning Object Detection from a Small Number of Examples: the Importance of Good Features”, In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 53–60, 2004.

[20] N. Dalal and B. Triggs: “Histograms of Oriented Gradients for Human Detection”, In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886–893, 2005.

[21] R. Lienhart and J. Maydt: “An Extended Set of Haar-like Features for Rapid Object Detection”, In *IEEE International Conference on Image Processing*, Vol. 1, pp. 900–903, 2002.

[22] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan: “Real Time Face Detection and Facial Expression Recognition: Development and Application to Human Computer Interaction”, In *Conference on Computer Vision and Pattern Recognition WorkShop*, pp. 139–157, 2003.

[23] B. Wu, H. Ai, C. Huang, and S. Lao: “Fast Rotation Invariant Multi-View Face Detection Based on Real AdaBoost”, In *International Conference on Automatic Face and Gesture Recognition*, pp. 79–84, 2004.

[24] T. Mita, T. Kaneko, B. Stenger, and O. Hori: “Discriminative Feature Co-occurrence Selection for Object Detection”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 7, pp. 1257–1269, 2008.

[25] C. Huang, H. Ai, T. Yamashita, S. Lao, and M. Kawade: “Incremental Learning of Boosted Face Detector”, In *IEEE International Conference on Computer Vision*, pp. 1–8, 2007.

[26] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng: “Fast Human Detection Using a Cascade of Histograms of Oriented Gradients”, In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 1491–1498, 2006.

[27] C. Hou, H. Ai, and S. Lao: “Multiview Pedestrian Detection Based on Vector Boosting”, In *IEEE Asian Conference on Computer Vision*, Vol. 4843, pp. 210–219, 2007.

[28] Y. Freund and R. E. Schapire: “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”, *Journal of Computer and System Sciences*, Vol. 55, No. 1, pp. 119–139, 1997.

[29] R. E. Schapire and Y. Singer: “Improved Boosting Algorithms Using Confidence-rated Predictions”, *Machine Learning*, Vol. 37, No. 3, pp. 297–336, 1999.

[30] “MIT+CMU Frontal Face Image”. http://vasc.ri.cmu.edu/idb/html/face/frontal_images/index.html.

[31] “UIUC Image Database for Car Detection”. http://l2r.cs.uiuc.edu/ cogcomp/Data/Car/.

[32] “INRIA Person Dataset”. http://pascal.inrialpes.fr/data/human/.

[33] “USC Pedestrian Detection Test Set”. http://iris.usc.edu/ bowu/DatasetWebpage/dataset.html/.

[34] M. Okutomi and T. Kanade: “A Multiple-Baseline Stereo”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 4, pp. 353–363, 1993.

[35] A. Ess, B. Leibe, and L. Gool: “Depth and Appearance for Mobile Scene Analysis”, In *IEEE International Conference on Computer Vision*, pp. 1–8, 2007.

[36] 池村翔, 藤吉弘亘: “距離情報に基づく局所特徴量によるリアルタイム人検出”, 電子情報通信学会論文誌, Vol. J93-D, No. 3, pp. 355–364, 2009.

[37] K. Hayashi, T. Hirai, K. Sumi, and K. Sasagawa: “Multiple-Person Tracking Using a Plan-View Map with Error Estimation”, In *IEEE Asian Conference on Computer Vision*, Vol. 3851, pp. 359–368, 2006.

[38] D. Comaniciu and P. Meer: "Mean Shift: A Robust Approach toward Feature Space Analysis", *IEEE Transaction on Pattern Analysis and Image Intelligence*, Vol. 25, No. 5, pp. 603–619, 2002.

[39] M. Isard and A. Blake: "Condensation - Conditional Density Propagation for Visual Tracking", *International Journal of Computer Vision,*, Vol. 29, No. 1, pp. 5–28, 1998.

[40] N. Oza and S. Rusell: "Online bagging and boosting", In *Artificial Intelligence and Statistics*, pp. 105–112, 2001.

[41] D. F. A.D. Jepson and T. F. El-Maraghi: "Robust Online Appearance Models for Visual Tracking", *International Journal of Computer Vision*, Vol. 25, No. 1, p. 10, 2003.

[42] L.Breiman: "Random Forests", *Machine Learning*, Vol. 45, No. 1, pp. 3–32, 2001.

[43] A. Torralba, K. P. Murphy, and W. T. Freeman: "Sharing features: efficient boosting procedures for multiclass object detection", In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 762–769, 2004.

[44] J. Gall and V.Lempitsky: "Class-specific hough forests for object detection", In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1022–1029, 2009.

[45] G. Shakhnarovich, P. Viola, and D. Trevor: "Fast pose estimation with parameter-sensitive hashing", In *IEEE International Conference on Computer Vision*, Vol. 2, pp. 750–757, 2003.

[46] A. Fathi and G. Mori: "Human pose estimation using motion exemplars", In *IEEE International Conference on Computer Vision*, pp. 1–8, 2007.

[47] A. Bissacco, M.-H. Yang, and S. Soatto: "Fast human pose estimation using appearance and motion via multi-dimensional boosting regression", In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[48] J. C. Niebles and L. Fei-Fei: "A hierarchical model of shape and appearance for human action classification", In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[49] K. Schindler and L. V. Gool: "Action snippets: How many frames does human action recognition require?", In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[50] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. H. Torr: "Randomized Trees for Human Pose Detection", In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[51] A. Bissacco, M.-H.Yang, and S. Soatto: "Detecting humans via their pose", In *Advances in Neural Information Processing Systems*, pp. 169–176, 2006.

[52] Z. Tu: "Probabilistic boosting-tree: learning discriminative models for classification, recognition and clustering", In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 1589–1596, 2005.

[53] R. Okada: "Discriminative and Generalized Hough Transform for Object Detection", In *IEEE International Conference on Computer Vision*, pp. 2000–2005, 2009.

[54] V. Lepetit, P. Lagger, and P. Fua: "Randomized trees for real-time keypoint recognition", In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 775–781, 2005.

[55] J. Shotton, M. Johnson, and R. Cipolla: "Semantic Texton Forests for Image Categorization and Segmentation", In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[56] P. Geurts, D. Ernst, and L. Wehenkel: "Extremely randomized trees", *Machine Learning*, Vol. 36, No. 1, pp. 3–42, 2006.

研究業績一覧

学術論文

[1] 山下隆義, 池村翔, 藤吉弘亘, 岩堀祐之: “距離情報を考慮したパーツ統合による人物検出の高精度化”, 電気学会論文誌, Vol. 131, No. 4, Sec. D, 2011(採録済み).

[2] 山下隆義, 藤吉弘亘: “Soft Decision Feature による姿勢変化に対応した人物追跡”, 情報処理学会論文誌, Vol. 51, No. 9, pp.1940–1950, 2010.

[3] 山下隆義, 労世紅，川出 雅人: “Online Real Boosting による人物追跡”, 情報処理学会論文誌コンピュータビジョンとイメージメディア (CVIM), Vol.1, No.1, pp.73–82, 2008.

[4] Y. Li, H. Ai, S. Lao, T. Yamashita, M. Kawade: “Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Life Spans”, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.30, No.10, pp.1728–1740, 2008.

国際会議発表論文 (査読あり)

[1] H. Wei, T. Yamashita, H. Lu, S. Lao: “SURF Tracking”, IEEE International Conference on Computer Vision, 2010.

[2] T. Yamashita, H. Fujiyoshi,S. Lao, M. Kawade: “Human tracking based on Soft Decision Feature and online real boosting”, International Conference on Pattern Recognition, 2008.

[3] Y. Li, H. Ai, S. Lao, T. Yamashita, M. Kawade: “Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Life spans”, IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[4] Xu. Li, T. Yamashita, S. Lao, M. Kawade,: “Online Real Boosting for Object Tracking Under Severe Appearance Changes and Occlusion”, IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.925–928, 2007.

学会口頭発表 (査読あり)

[1] 山下隆義, 山内悠嗣, 藤吉弘亘: “Boosted Randomized Trees による人物検出と行動の同時認識”, 第 13 回画像の認識・理解シンポジウム (MIRU2010), 2010.

[2] 松島千佳, 山内悠嗣, 山下隆義. 藤吉弘亘: “Relational Binarized HOG 特徴量と Real AdaBoost によるバイナリ選択を用いた物体検出”, 第 13 回画像の認識・理解シンポジウム (MIRU2010), 2010.

[3] 山下隆義, 藤吉弘亘, 労世紅, 川出雅人: “姿勢変化に対応した Soft Decision Feature と Online Real Boosting による人物追跡”, 第 11 回画像の認識・理解シンポジウム (MIRU2008), 2008.

学会口頭発表 (査読なし)

[1] 山下隆義, 野田隆史, 労世紅, 川出 雅人: “Real AdaBoost を利用した Online Boosting による人物追跡”, 情報処理学会　研究報告 (CVIM), No.158, pp.85–92, 2007.

解説記事

[1] 山下隆義: “統計的学習手法を用いた物体認識における特徴量の進化”，日本工業出版，画像ラボ Vol.20, No.1, pp.53-58, 2009.

チュートリアル/講演

[1] 山下隆義: “物体認識における特徴量の発展 〜今からでも遅くない！物体認識〜”，第 15 回画像センシングシンポジウム (SSII09), 2009.

[2] 山下隆義, 藤吉弘亘: “特定物体認識に有効な特徴量”, 情報処理学会 研究報告 CVIM No.165, pp. 221-236, 2008.

[3] 山下隆義: “統計的学習手法を用いた物体認識における特徴量の進化”, 第 14 回画像センシングシンポジウム (SSII08), オーガナイズドセッション (OS-04), 2008.

著書

[1] 藤吉弘亘, 山下隆義, 岡田和典, 前田英作, ノジク・ヴァンソン, 石川尋代, ドゥソルビエ・フランソワ: “コンピュータビジョン最先端ガイド 2”, ISBN 978-4915851-37-7, 1 章担当, 2010.

[2] 山下隆義 (5 章担当): “映像情報メディア工学大事典 基礎編　第 4 部門　画像処理”, オーム社, ISBN:978-4274208-69-0, 2010.

学術表彰

[1] MIRU2010　ベストインタラクティブセッション賞
題目「Boosted Randomized Trees による人物検出と行動の同時認識」

[2] MIRU2010 優秀学生発表賞
題目「Relational Binarized HOG 特徴量と Real AdaBoost によるバイナリ選択を用いた物体検出」

[3] SSII09　高木賞
題目「顔画像センシング技術　OKAO Vision」

[4] CVPR2007　Best Student Paper Award
題目「Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Lifespans」