

# AdaBoost とパーティクルフィルタによる移動物体追跡の効率化

## Increasing Efficiency of Tracking Moving Object using AdaBoost and Particle Filtering

園田 直樹 † 岩堀 祐之 † 福井 真二 ‡ 河中 治樹 ††  
Naoki Enda Yuji Iwahori Shinji Fukui Haruki Kawanaka

### 1 はじめに

動画像における移動物体の追跡技術として、ノイズやオクルージョンに頑健な確率的手法であるパーティクルフィルタ (以下 PF とする) [1] が注目されている。PF では、追跡対象の事後確率を近似するためにパーティクル (粒子) 群を用いる。乱数を用いて粒子を分散させるため、事後確率の密な近似を得てひとつの対象を安定して追跡するためには数多くの粒子が必要となる。各粒子において尤度評価を行う必要があり、追跡対象の増加に伴い粒子数が増えるため、追跡対象の数に比例して演算量が増加し、多数の追跡対象をリアルタイムに追跡することが困難となる。この問題は、粒子の分散範囲を限定し、個々の追跡対象に使用する粒子数を減らすことにより解決することができる。粒子が分散する範囲を限定するためには、粒子を分散させる前に追跡対象領域を効率よく検出しておくことが重要である。

そこで、対象と非対象とを高速に判別することが可能な AdaBoost[2] に注目し、AdaBoost を用いた PF の効率化を行う手法が提案された [3]。AdaBoost はパターン認識の統計的手法として、ある例題に対する重み付き学習を通して、単純なしきい値演算などで分類できない複雑な問題にも対応した判別関数を生成することができる。AdaBoost は顔検出などに用いられ顕著な性能を示すことが知られている [4][5]。この判別関数による分類を移動物体領域の検出に応用することで、PF の処理効率を上げることができた [3]。しかし、文献 [3] で用いられている AdaBoost 判別特徴量は単純なエッジ情報のみであるため、複雑な背景において精度良く移動物体領域を判別することができず、粒子の分散範囲を限定できない場合があった。

そこで本論文では、より複雑な背景に対応でき、かつ処理コストが低い AdaBoost 判別特徴量を提案する。これにより、複雑な背景においても AdaBoost を用いて粒子の分散範囲を限定することができ、PF の効率化を行うことが可能となる。

### 2 PF による移動物体追跡

PF とは、追跡対象の状態を確率変数とする確率密度を粒子群  $s^{(i)} = \{x^{(i)}, \pi^{(i)}\}_{i=1, \dots, N}$  によって近似する手法である。各粒子は状態空間に状態量  $x$  を持ち、その確率は重み  $\pi$  に比例する。ある時刻  $t$  における追跡対象の状態量を  $x_t$ 、そのときの観測結果を  $z_t$  とし、時刻  $t$  までに得られる観測結果を  $Z_t = (z_1, \dots, z_t)$  とする。粒子は事前に与

えられた状態空間モデルによって推移し、時刻  $t$  までの観測結果が得られたときの状態  $x_t$  の事後確率密度  $p(x_t|Z_t)$  をベイズの法則に従って次式のように推定する。

$$p(x_t|Z_t) = \alpha p(z_t|x_t)p(x_t|Z_{t-1}) \quad (1)$$

ただし、 $\alpha$  は正規化定数である。

時刻  $t-1$  において、観測  $Z_{t-1}$  が得られたときの状態量  $x_{t-1}$  の分布  $p(x_{t-1}|Z_{t-1})$  が、 $N$  個の重みつき仮説群  $s_{t-1}^{(i)} (i = 1, \dots, N)$  で表されているとき、各仮説の重み  $\pi_{t-1}^{(i)}$  の比に従い、 $N$  個の仮説群  $s_{t-1}^{(i)}$  を選択する。

事前に与えられた状態空間モデルによって各粒子を推移させることで確率分布  $p(x_t|Z_{t-1})$  を近似する。本論文では、追跡対象の中心画素座標  $(u, v)$ 、追跡対象を内包するであろう矩形の幅  $w$  および高さ  $h$  を状態量  $x$  とし、状態空間モデルは

$$x_t = x_{t-1} + \omega_t \quad (2)$$

として変動させる。ただし、 $\omega_t$  は正規乱数である。新しいサンプル  $s_t^{(i)}$  の重み  $\pi_t^{(i)}$  は画像の色情報に基づく尤度から推定する。重みの和は 1 になるように正規化する。その結果、時刻  $t$  における  $p(x_t|Z_t)$  の近似表現を得ることができる。また、追跡対象の最適な状態量推定には仮説群の期待値を用いる。尤度関数  $L(z_t|x_t)$  を利用して、 $\pi_t^{(i)} = L(z_t|x_t)$  により重みを計算し、Sequential Importance Sampling[6] によりリサンプリングを行う。

尤度関数には HSV 表色系を用い、現在の画像における追跡対象を内包する矩形領域の HSV の正規化色ヒストグラム  $H_u$  と予め得ておいた参照用ヒストグラム  $H_{ref}$  との類似度  $S$  を次式の Swain の Histogram Intersection[7] を用いて計算する。

$$S = \sum_{i=1}^{(\text{Tone})^3} \min(H_u, H_{ref}) \quad (3)$$

ただし、Tone は HSV の各階調を表す。この類似度  $S$  を用いて尤度関数を次式のように

$$L(z_t|x_t) = \exp(kS^2) \quad (4)$$

ただし、 $k$  は定数である。以上より、追跡対象の存在可能性が高い仮説に大きな重みを与えることができる。

### 3 AdaBoost による PF の効率化

従来の PF では時刻  $t-1$  のリサンプリング結果から得た推定状態を元に、乱数を用いて分散させ時刻  $t$  の粒子を生成する。しかし、この方法では対象を捕らえていない状態の粒子も生成されるため、効率の悪い粒子の分布になってしまっている。

† 中部大学, Chubu University

‡ 愛知教育大学, Aichi University of Education

†† 愛知県立大学, Aichi Prefectural University

そこで、本論文では PF の効率化のために、AdaBoost を用いて移動物体領域を検出し粒子の分散領域および矩形サイズの制限を行う。AdaBoost では単純な計算で高速に粒子分散範囲を判別し、その領域内で追跡対象を色ヒストグラムを用いた PF によって推定する。これにより粒子が対象を正しく捕らえ、使用する粒子数削減することが可能なため、AdaBoost による移動物体領域検出の処理時間の増加があっても全体の処理時間の短縮が期待できる。

### 3.1 AdaBoost

本論文では、二値判別問題を考えたときの AdaBoost を対象とする。

例題  $(x_i, y_i) : i = 1, \dots, n$  に対して  $\pm 1$  の値をとる弱学習器の集合  $f_j(x) : j \in 1, \dots, J$  を用意する。ここで、 $n$  は例題の個数、 $J$  は弱学習器の個数を表す。このとき、AdaBoost の学習アルゴリズムは次のように表される。

#### (1) 重みの初期化

$$\omega_1(i) = \frac{1}{n}, (i = 1, \dots, n)$$

#### (2) Boosting Round $t = 1, \dots, T$ での重み付きエラーレート

$$\epsilon_t(f) = \sum_{i=1}^n I(f_j(x_i) \neq y_i) \omega_t(i)$$

##### (2.1) エラーレート最小の弱学習器の選択

$$f_t = \arg \min \epsilon_t(f_j)$$

##### (2.2) 結合比重 (Union Weight)

$$\beta_t = \frac{1}{2} \log \frac{1 - \epsilon_t(f_t)}{\epsilon_t(f_t)}$$

##### (2.3) 重み更新

$$\omega_{t+1}(i) = \frac{\omega_t(i) \exp(-y_i \beta_t f_t(x_i))}{\sum_{k=1}^n \omega_t(k) \exp(-y_i \beta_t f_t(x_k))}$$

#### (3) 重み付き多数決

$$f = \text{sign}(F_t(x)), F_t = \sum_{t=1}^T \beta_t f_t(x)$$

ただし、 $I$  は弱学習器  $f_j$  の出力と例題の答えが異なる場合には 1 を、同じ場合には 0 を返す関数である。(1) で各例題の重みの初期値として一様な値を与える。次に、(2) で各弱学習器において、例題との正誤に応じてエラーレートを求め、エラーレートが最小となる弱学習器を選択し、信頼性を表す結合比重の値を求める。このとき、エラーレートが 0.5 以上になる場合には符号を変え  $-f_t$  にすることで 0.5 以下にする。また、選ばれた弱学習器が Boosting Round  $t+1$  において最も不利になるように各例題の重みを更新する。そして、(3) において、学習で得られた  $T$  個の弱学習器  $f_t$  を結合比重  $\beta_t$  に基づいて重み付き多数決をとることにより強学習器  $f$  が得られる。

### 3.2 AdaBoost での学習

AdaBoost によって追跡対象領域と非追跡対象領域を判別するためには、判別の根拠となる特徴量が必要となる。本研究では、PF の効率化による処理時間の軽減を目的とするため、毎フレーム行う AdaBoost による判別は高速であるべきであり、未知データの特徴量取得においても計算量は可能な限り少なくしたい。

そこで、計算コストが少なく、画像中の追跡対象領域情報を得ることができる特徴量を使用する。まず、RGB 背景差分を用いる。判別領域に各色  $c \in \{R, G, B\}$  について背景差分を適用することで得られる値の平均値を求め、3次元の特徴量とする。現フレーム画像を  $I_c(x, y)$ 、背景画像を  $B_c(x, y)$ 、判別領域を  $X \times Y$  の窓範囲とすると次式か

ら特徴量  $C_c$  を求める。

$$C_c = \frac{\sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} |I_c(x+i, y+j) - B_c(x+i, y+j)|}{X \times Y} \quad (5)$$

次に、フレーム間差分を用いる。フレーム間差分値を求める際には、RGB 値からモノクロ濃淡値に変換したものをを用いる。判別領域にフレーム間差分を適用することで得られる値の平均値を求め、1次元の特徴量とする。入力画像を  $G(x, y)$  とすると、次式から特徴量  $C_f$  を求める。

$$C_f = \frac{\sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} |G^t(x+i, y+j) - G^{t-1}(x+i, y+j)|}{X \times Y} \quad (6)$$

最後に、エッジ背景差分を用いる。判別領域に四方向(垂直、水平、右上がり、左上がり)のエッジ背景差分を適用することで得られる値の平均値を求め、4次元の特徴量とする。このとき、各方向  $v \in \{V, H, R, L\}$  のエッジ検出には次式のエッジオペレータを使用する。

$$h_V = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h_H = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$h_R = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad h_L = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

各方向  $v$  のオペレータを適用することで、各方向  $v$  のエッジを得ることができるが、判別領域全てのエッジを求めるには処理に時間がかかってしまう。そこで、処理時間を軽減させるために、AdaBoost による判別範囲を低解像度化することにより、オペレータを適用させる回数を少なくすることで、処理時間を軽減させる。判別領域  $X \times Y$  画素から  $\frac{X}{5} \times \frac{Y}{5}$  画素の大きさへと解像度を下げる。エッジ背景差分には RGB 値からモノクロ濃淡値に変換したものをを使用する。入力画像  $G(x, y)$  とすると各方向  $v$  のエッジ強度  $e_v$  は次式のように求められる。

$$e_v(x, y) = \left| \sum_{m=0}^2 \sum_{n=0}^2 G(x+m, y+n) h_v(m, n) \right| \quad (7)$$

現フレーム画像と、背景画像から求められた各方向  $v$  のエッジ強度  $e_v$  を用いて次式から特徴量  $E_v$  を求める。

$$E_v = \frac{\sum_{i=0}^{X/5-1} \sum_{j=0}^{Y/5-1} |e_v^n(x+i, y+j) - e_v^b(x+i, y+j)|}{\frac{X \times Y}{25}} \quad (8)$$

AdaBoost 判別には RGB 背景差分値 (3次元)、フレーム間差分値 (1次元)、エッジ背景差分値 (4次元) を合わせた 8次元の特徴量を用いる。AdaBoost に与える判別特徴量は次式のような 8次元ベクトル  $V$  で構成される。

$$V = [C_R \ C_G \ C_B \ C_f \ E_V \ E_H \ E_R \ E_L]^T \quad (9)$$

このようにして得られたベクトル  $V$  を 8次元空間で AdaBoost によって識別することで、追跡対象領域と非追跡対象領域を判別する。学習データは追跡対象領域の特徴量と非追跡対象領域の特徴量それぞれに対して答えを



図1 シーン1

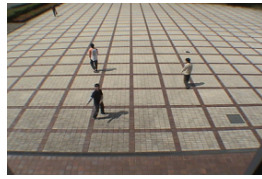


図2 シーン2



図3 シーン3



図4 シーン4

合わせて用意する必要がある．AdaBoost では特徴量に対する答えとして  $\pm 1$  をとるが，本論文では  $+1$  を追跡対象領域， $-1$  を非追跡対象領域とする．

8次元の特徴量においては，弱学習器の判別の種類は16種類となる．重み付きされた学習データに対してこの16種類の判別を行い，それぞれしきい値を変動させて最もエラーの少なかった判別をその Boosting Round  $t$  における判別とする．結合比重  $\beta_t$  は判別時のエラーに基づいた弱学習器の精度となる．ただし，判別を単純なしきい値演算の繰り返しにするため，Boosting Round 1回につき1つの特徴量に対するしきい値演算の判別を行うようにした．その Boosting Round の重み付き学習データに対するしきい値演算を繰り返すことで，単純なしきい値演算が重み付き多数決時には複雑な関数として機能する．AdaBoost は問題に対してエラーの少ない重要な判別から行うため，結合比重は指数的に下がる．本研究では結合比重は初期値の1%を下回った時点で Boosting を終了することとした．

### 3.3 粒子分散範囲の制限

前フレームの処理結果として得られた追跡対象の中心から粒子の分散する範囲を AdaBoost での判別範囲とし，その範囲内に重ならないように設置された  $X \times Y$  画素からなる窓それぞれから8次元の特徴量  $V$  を得る．この未知データに対して，事前学習で得た AdaBoost の判別関数を用いて追跡対象領域および非追跡対象領域の判別を行う．弱学習器の重み付きの判別結果を全弱学習器分足し合わせた結果，強学習器の出力が  $f \geq 0$  なら追跡対象領域， $f < 0$  なら非追跡対象領域となる．このように，AdaBoost の判別により求められた追跡対象領域に各粒子の分散が収まるように制限する．

### 3.4 矩形サイズの制限

粒子の分散領域の制限の有無にかかわらず，追跡対象を内包すべき矩形の大きさを従来のように乱数を用いて変動させた時には次のような問題が生じる．まず，追跡対象が物体や別の対象に隠れたときに矩形が小さくなり，その後も体の一部分のヒストグラムの類似度が高かった場合，矩形の大きさが対象を捕らえないまま追跡が行われることがある．また，カメラとの距離の変化などから画像内での追跡対象の大きさが変化する場合にも，同様の理由により対象を矩形内に捕らえないまま追跡が行われることがある．さらに，粒子の分散領域を制限するだけでは，矩形内の大きさについてのサンプリング量が少



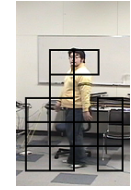
(a) シーン1



(b) シーン2

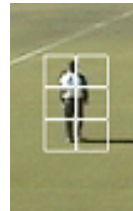


(c) シーン3



(d) シーン4

図5 検出結果(文献[3]の手法)



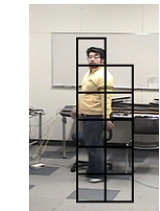
(a) シーン1



(b) シーン2



(c) シーン3



(d) シーン4

図6 検出結果(提案手法)

なくなり，対象を捕らえるのに十分な大きさの矩形を持つ粒子が発生しない可能性がある．

そこで，時刻  $t$  における未知データに対する AdaBoost の判別結果から，PF の矩形の幅および高さそれぞれの最大値および最小値を与えることにする．各追跡対象ごとに追跡対象領域と判断されたブロック群の横の総画素数を矩形の横幅の最大値とする．また，最小値はブロック群の横の総画素数から両端のブロックの画素数を除いた長さとする．すなわち，ブロック群の横の個数を  $n$ ，1ブロックの総画素数を  $p$  としたとき， $n > 2$  の場合には最大値を  $n \times p$ ，最小値を  $(n-2) \times p$  とし， $n \leq 2$  の場合には最大値を  $n \times p$ ，最小値  $(n-1) \times p$  とする．矩形の高さの最大値および最小値も同様に求める．

以上のようにして，AdaBoost による分散する粒子の位置および矩形の大きさの制限を行って追跡を行う．制限を行うことで矩形は対象を捕らえる大きさで変動しており，より正確な対象位置を推定することができる．

## 4 実験

本実験で用いた映像は  $720 \times 480$  画素，計算機は Core 2 Duo E6600，メモリ 1024MByte である．学習データの数は判別関数を作るのに十分と考えられる 20,000 個を与えた．そのうち追跡対象領域が 10,000 個，非追跡対象領域が 10,000 個である．重み付き学習で生成された，判別の種類，しきい値，結合比重を持った弱学習器を用いて，未知データに対する重み付き多数決を用いた判別関数によって追跡対象領域の検出を行った．

### 4.1 AdaBoost による物体検出に対する検証

文献 [3] の手法と提案手法での追跡対象領域および非追跡対象領域の判別について検証を行った．検証は，図 1, 2, 3, 4 に示すような動画を用いて 50 フレーム中から行った．また，判別を行う窓範囲の大きさは動画像中に映っている対象物の大きさから，シーン 1 は  $20 \times 20$  画素，シーン 2, 3 は  $30 \times 30$  画素，シーン 4 は  $40 \times 40$  画素とした．これらの動画における文献 [3] の手法と提案手法による検出率，未検出率および誤検出率を求めた結果

表1 検出率(文献[3]の手法)

シーン	検出率 [%]	未検出率 [%]	誤検出率 [%]
1	86.71	13.29	1.76
2	81.04	18.96	31.75
3	83.30	16.70	44.91
4	70.10	29.90	42.23

表2 検出率(提案手法)

シーン	検出率 [%]	未検出率 [%]	誤検出率 [%]
1	91.03	8.97	3.03
2	98.24	1.76	8.50
3	98.80	1.20	5.56
4	98.25	1.75	10.43

表3 追跡処理時間 [ms]

対象数	PFのみ	AdaBoost併用
1	18.829	4.400
2	25.790	6.075

を表1, 2に, AdaBoostによる検出の様子を図5, 6に示す. 目視によって理想的な検出数を求め, それに対してAdaBoostにての検出結果を比較して, 未検出率および誤検出率を求めた. 表1, 2より, シーン1においては文献[3]の手法, 提案手法どちらの場合においても精度良く判別されていることが確認できる. また, 文献[3]の手法では精度良く判別できない背景の動画であるシーン2, 3, 4において, 提案手法では精度良く判別されていることが確認できる. シーン2, 3, 4の誤検出率は未検出率と比べると大きいことがわかる. これは, シーン2, 4においては影領域の誤検出, シーン3においては, 追跡対象が車両に映り込んでしまったことが原因である. 文献[3]の手法では, シーン2, 3, 4において判別精度が悪いため粒子の分散位置が追跡対象以外の場所に多く配置され, かつ矩形サイズの制限がうまくいかないため粒子数を少なくすると追跡が破綻してしまうことがある. しかし, 提案手法により判別精度が向上したため粒子数を少なくしても安定して追跡することができた.

#### 4.2 PFの効率化による処理時間の検証

これまでの実験と同様の条件下において, PFの場合とAdaBoostとPFを用いた場合の処理時間について検証を行い, AdaBoostによる粒子数の削除の効果を確認を行った. AdaBoostを使用しないPFでは, 安定した追跡を行うために1対象につき少なくとも30個の粒子を必要とする. 本来は十分な追跡精度を確保するためにより多数の粒子を使用すべきであるが, ここでは処理速度の比較対象として追跡が破綻しない最低数の30個に設定した. また, AdaBoostを用いてPFの効率化を行った場合, 追跡が破綻しない最低数として1対象につき粒子数を5個とした. 追跡対象の数による処理時間の変化をみるために, 1人および2人の追跡における実験を行った. 実験には図3のように追跡対象が約145×60画素の大きさで映っている動画を用いた. 実験では, 400フレームでの平均処理時間を計測した. AdaBoostとPFを併用した場合の処理時間はAdaBoostによる検出とPFでの追跡処理を合計した時間である.

表3の結果より, PFのみを用いた場合より, AdaBoostとPFを併用した場合の方が高速に処理できたことが確認できる. 対象の大きさにより矩形の大きさが変わること, 色ヒストグラムの類似度の計算量に違いが生じるため, 粒子数削除の効果に違いがあるが, どちらの場合においてもAdaBoostでの判別処理が追加され処理時間が増えたにもかかわらず, 処理時間の短縮が行われていることがわかる. 対象が増えるにつれて粒子数削除の効果が高まると考えられるため, 多人数の追跡時に大きな効果が期待できる.

## 5 おわりに

本論文では, PFにAdaBoostを組み入れることで複雑な背景においても頑健に追跡可能な手法を新たに提案した. 提案手法においてはいくつかの特徴量を新たに導入することにより, 複雑な背景におけるAdaBoostの判別精度の向上を図った. 文献[3]の手法では複雑な背景になると判別精度が悪く, 追跡対象領域のみに粒子の分散を制限できず, 粒子数を減らすとPFによる追跡が破綻してしまう問題点があった. 他方, 本手法では複雑な背景においても精度良く判別できるため, 粒子の分散を効率よくでき, 粒子数を減らしても破綻することなく追跡が可能となった. また, 本手法でのAdaBoostによる判別に加え, PFによる追跡処理を行う際には, PFのみを用いた場合と比べても処理時間短縮を行うことができた.

今後の課題としては, 影領域, 映りこみにおいて生じる誤検出の改善や, AdaBoostでの判別を行う窓範囲の動的な決定方法などがあげられる.

## 謝辞

本研究の一部は中部大学研究費及び堀情報科学振興財団のサポートによる. ここに感謝申し上げる.

## 参考文献

- [1] Michael Isard and Andrew Blacke, "CONDENSATION - conditional density propagation for visual tracking -", *Intl. J. of Computer Vision*, **29**(1), 5-28, 1998.
- [2] Y.Freund and R.E.Schapiro, "A decision-theoretic generalization of on-line learning and an application to boosting", *J. of Computer and System Sciences*, **55**, pp.119-139, 1997.
- [3] 河中治樹, 大西賢, 岩堀祐之, 福井真二, "Adaboostを用いたパーティクルフィルタの効率化", 画像の認識・理解シンポジウム2006(MIRU2006) 論文集, IS1-5.
- [4] 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の学習", 画像センシングシンポジウム(SSII04), 2004.
- [5] 勞世, 山下隆義, 岡本卓也, 川出雅人, "高速全方向顔検出", 画像の認識・理解シンポジウム(MIRU2004), II, pp.271-276, July, 2004.
- [6] A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering", *Statistics and Computing*, Vol.10, pp.197-208, 2000.
- [7] M. J. Swain and D. H. Ballard, "Color Indexing", *Intl. J. Computer Vision*, No.1, Vol.7, pp.11-32, 1991.