

近似計算を導入した線形識別器の早期判定による高速な識別

後藤 雄飛^{†a)} 土屋 成光^{†b)} 山内 悠嗣^{†c)}
藤吉 弘亘^{†d)}

Fast Discrimination by Early Judgment Using Linear Classifier Based on Approximation Calculation

Yuhi GOTO^{†a)}, Masamitsu TSUCHIYA^{†b)}, Yuji YAMAUCHI^{†c)},
and Hironobu FUJIYOSHI^{†d)}

あらまし 物体検出は、入力画像を網羅的にラスタスキャンさせて得られる、膨大な数の検出ウィンドウを検出対象か非検出対象に分類する。各検出ウィンドウは、抽出した局所特徴量と統計的学習手法により学習した識別器により出力値を算出し、対象のクラスに分類する。近年では、物体検出に一般的に利用されている線形 SVM により学習した識別器を、対応点追跡の問題において近似計算をして高速な識別を実現する方法がある。そこで、本研究では物体検出に対して線形 SVM の近似計算法を導入し、ラスタスキャンに伴い発生する識別処理の高速化を検討する。更に、提案手法では近似計算結果に応じて識別器の出力値を早期判定する。これにより、入力された検出ウィンドウに対して、近似計算に必要な基底数を適応的に判断して分類することで線形 SVM の高速な識別を可能にする。また、提案手法では線形 SVM を近似計算する際に用いる、HOG 特徴量をバイナリコード化した B-HOG 特徴量を共起表現して物体検出の高精度化を行う。人検出による評価実験より、提案手法は HOG 特徴量と線形 SVM による物体検出手法と比較して、約 6.1% 高精度な識別に加えて、約 17 倍高速な識別計算を実現した。

キーワード 早期判定, 線形 SVM, 内積近似計算, 人検出

1. ま え が き

物体検出は、学習サンプルから抽出した局所特徴量と統計的学習手法を用いて学習した識別器により実現されている。代表的な物体検出として、2005 年に Dalal らは HOG 特徴量と線形 SVM [1] による人検出法 [2] を提案した。HOG 特徴量と線形 SVM の組み合わせは、検出対象の隠れの推定 [3], [4] やパーツモデル [5] 等の物体検出の高精度化に関する研究においても、ベースとなる手法として用いられている。このような物体検出の実用化には識別性能の高精度化だけではなく、検出過程の高速化と省メモリ化が課題である。検出過程では、入力画像を網羅的にラスタスキャン

した際に発生する膨大な数の検出ウィンドウを識別処理する必要がある。高速な物体検出を実現するには各検出ウィンドウの処理である特徴抽出過程と識別過程をそれぞれ高速化する必要がある。特徴抽出過程の高速化としては、HOG 特徴量の算出に積分画像 [6] や GPU [7] を利用する手法が提案されている。中でも、GPU で実装された fast HOG [8] は、CPU による特徴抽出と比較して約 95 倍高速な処理を実現した。識別過程では、AdaBoost [9] をカスケード型に構成し、非検出対象を早期判定することにより高速化する手法 [10] が提案されている。しかし、物体検出に多く利用されている線形 SVM の高速化に関する検討は従来行われていない。

そこで、本研究では線形 SVM による識別結果の早期判定を導入して識別器の高速化を検討する。提案手法は、検出ウィンドウから抽出した特徴ベクトルに応じて線形 SVM を近似計算し、識別結果を早期判定することで高速な識別処理を実現する。このとき、線形 SVM の近似計算を高速に求めるために入力特徴ベク

[†] 中部大学情報工学科, 春日井市

Department of Computer Science, Chubu University, 1200 Matsumoto, Kasugai-shi, 487-8501 Japan

a) E-mail: yuhi@vision.cs.chubu.ac.jp

b) E-mail: mtdoll@vision.cs.chubu.ac.jp

c) E-mail: yuu@vision.cs.chubu.ac.jp

d) E-mail: hf@cs.chubu.ac.jp

トルはバイナリコードである必要があるため、提案手法では HOG 特徴量をバイナリコード化した B-HOG 特徴量 [11] を用いる。B-HOG 特徴量は HOG 特徴量と比較して 1/8 のメモリ量で特徴ベクトルを表現することができるが、HOG 特徴量を 2 値化した際の量子化誤差の影響により、識別精度が低下するという問題がある。そこで、提案手法では B-HOG 特徴量のバイナリコードにビット演算を用いた共起表現を導入して、セル間の関係性を表現することで計算コストを最小限に抑えながら識別性能の高精度化を行う。このように、本研究では識別器の早期判定による識別処理と B-HOG 特徴量の共起表現により、高精度でかつ高速な物体検出を実現する。

2. ラスタスキャンベースの物体検出

物体検出は、局所特徴量と統計的学習手法を用いて学習した識別器を利用する手法が一般的である。物体検出を実現するには入力画像 I に対して検出ウィンドウを網羅的にラスタスキャンする。そして、ラスタスキャンして得られた全ての検出ウィンドウに対して特徴ベクトルを抽出し、事前に統計的学習手法により構築した識別器を用いて、クラス y_k を検出対象の場合 1、非検出対象の場合 -1 として判別する。

人を検出対象としたラスタスキャンベースの物体検出手法には、2005 年に Dalal らが提案した HOG 特徴量と線形 SVM による手法 [2] が多く用いられている。HOG 特徴量を小規模なハードウェアで実装するために、HOG 特徴量を 2 値化した B-HOG 特徴量 [11] が提案されている。本章では、B-HOG 特徴量、線形 SVM [1] により構築した識別器とその問題点について述べる。

2.1 B-HOG 特徴量

Histograms of Oriented Gradients (HOG) 特徴量 [2] は、検出ウィンドウからセルと呼ばれる局所領域ごとに作成した勾配方向ヒストグラムを特徴量とする。また、複数のセルで構成されるブロック領域ごとに特徴量を正規化することで、照明変化の影響を受けにくい特徴量となる。

B-HOG 特徴量 [11] は、図 1 に示すように HOG 特徴量を 2 値化したバイナリコードで物体形状を表現する。HOG 特徴量により作成した勾配方向ヒストグラム $V_c = \{v_c(1), v_c(2), \dots, v_c(N)\}$ を式 (1) に示すようにしきい値 th_n により 2 値化することによりバイナリコード $P_c = \{p_c(1), p_c(2), \dots, p_c(N)\}$ を生成する。こ

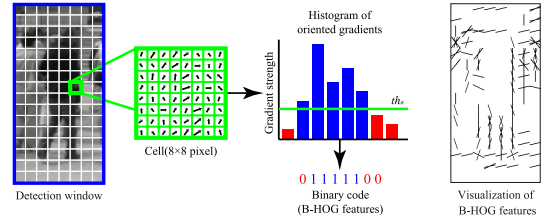


図 1 B-HOG 特徴量
Fig. 1 B-HOG features.

こで、 N はこう配方向ヒストグラムの量子化数である。

$$p_c(n) = \begin{cases} 1 & \text{if } v_c(n) > th_n \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

しきい値 th_n は学習サンプルから得られる HOG 特徴量の各次元ごとの平均値を用いる。B-HOG 特徴量は 64×128 ピクセルの検出ウィンドウに対して、セルサイズ $M = 8$ ピクセル、ブロックサイズ $R = 2$ セルの場合に $\{(64/M) - (R - 1)\} \times \{(128/M) - (R - 1)\} = 105$ 個のブロック領域により正規化処理が行われる。そのため、勾配方向ヒストグラムの量子化数 $N = 8$ のとき、一つの検出ウィンドウから得られるバイナリコードは $2 \times 2 \times 8 \times 105 = 3,360$ ビットとなり、HOG 特徴量と比べてメモリ量を 1/8 に削減することができる^(注1)。しかし、B-HOG 特徴量は HOG 特徴量と比較して、2 値化の際の量子化誤差の影響により物体検出の識別精度が低下するという問題がある。

2.2 線形 SVM による識別器

Support Vector Machine (SVM) [1] は、パターン認識手法において優れた認識性能をもつ統計的学習手法の一つである。コンピュータビジョンの分野においても非常に多くの研究に用いられ、物体検出においては線形 SVM が用いられることが多い。

線形 SVM における識別関数は式 (2) のように定義される。

$$F(\mathbf{x}) = \mathbf{w}^T \mathbf{x} = \sum_{i=1}^D w_i x_i \quad (2)$$

ここで、 $\mathbf{x}$ は、 D 次元の特徴ベクトル、 $\mathbf{w}$ は特徴ベク

(注1)：予備実験より倍精度浮動小数点型で表現した HOG 特徴量と符号なし 8 ビット整数型で表現した HOG 特徴量の人検出性能は同等であった。本研究では、より実用的な比較として HOG 特徴量を符号なし 8 ビット整数型で表現するため、B-HOG 特徴量は HOG 特徴量よりも 1/8 のメモリ量で表現される。

トルに対する重みを表す．物体検出を行うためには，ラスタスキャンにより得られた検出ウィンドウから抽出した特徴ベクトル $\mathbf{x}$ に対して，識別器 $F(\mathbf{x})$ の出力値を求め，しきい値処理することでクラスを判定する．

2.3 問題点

VGA サイズ (640×480 ピクセル) の入力画像 I に対して横幅 l_w ，縦幅 l_h の検出ウィンドウをスケール l_s ，ずらし幅 l_m にてラスタスキャンしたとき，発生する検出ウィンドウの数 K は次式により求めることができる．

$$K = \sum_{l_s=1}^S \text{floor}((640-l_w \times l_s - 1) / l_m) + \sum_{l_s=1}^S \text{floor}((480-l_h \times l_s - 1) / l_m) \quad (3)$$

ここで， $\text{floor}()$ は床関数であり，入力された実数の小数点以下を切り捨てた整数を出力する．例えば，検出ウィンドウのサイズ $(l_w, l_h) = (64, 128)$ をマルチスケール $S = 2$ ($l_s = 1, 2$) でずらし幅 $l_m = 4$ としてラスタスキャンした場合，検出ウィンドウは約 2 万個となる．これら全ての検出ウィンドウから特徴抽出を行い，識別器 $F(\mathbf{x}_k)$ の出力値を求める必要があり，物体検出をリアルタイムに実現するためには，特徴抽出過程と識別過程をそれぞれ高速化する必要がある．特徴抽出過程の高速化としては，積分画像を用いた HOG 特徴量の高速化 [6] が提案されている．識別過程では，膨大な数の検出ウィンドウを処理する際に，線形 SVM の実数ベクトル間の内積にかかる計算コストが高いという問題がある．そこで，GPU を用いた線形 SVM における内積計算の並列処理による高速化手法が提案されている [8]．しかし，この高速化手法は GPU 等の数百個規模の並列演算ユニットを別途必要とする．

3. 提案手法

本研究では，識別処理の高速化として線形 SVM に近似計算による早期判定を導入する．また，線形 SVM の近似計算に基づく高速化に必要な入力特徴ベクトルのバイナリ化に伴う性能低下を解決するために，HOG 特徴量をバイナリコード化した B-HOG 特徴量の共起性を表現することで，物体検出の高精度化を行う．以下に提案手法による識別処理の高速化とバイナリコードにおける共起表現について述べる．

3.1 線形 SVM の近似計算

線形 SVM により学習した識別器 $F(\mathbf{x})$ の計算は，

Algorithm 1 Decomposition of the weight vector $\mathbf{w}$ [12].

Require: weight vector $\mathbf{w}$, number of bases, N_b

1. Initialization: Store the weight vector $\mathbf{w}$ of the linear SVM in real vector $\mathbf{r}$.
 $\mathbf{r} = \mathbf{w}$
2. Analyze $\mathbf{w}$ into a real vector $\{\beta_i\}_{i=1}^{N_b}$ and binary code $\{\mathbf{b}_i\}_{i=1}^{N_b}$
for $i = 1$ to N_b **do** // N_b : number of bases
 - 2.1 Quantize real vector $\mathbf{r}$ to $\{-1, 1\}$ and calculate base binary code $\mathbf{b}_i$.
 $\mathbf{b}_i = \text{sign}(\mathbf{r})$
 - 2.2 Take the inner product of real vector $\mathbf{r}$ and base binary code $\mathbf{b}_i$.
 $\beta_i = \frac{\langle \mathbf{r}, \mathbf{b}_i \rangle}{\|\mathbf{b}_i\|^2}$
 - 2.3 Substitute the residual from the approximation into the real vector $\mathbf{r}$.
 $\mathbf{r} \leftarrow \mathbf{r} - \beta_i \mathbf{b}_i$

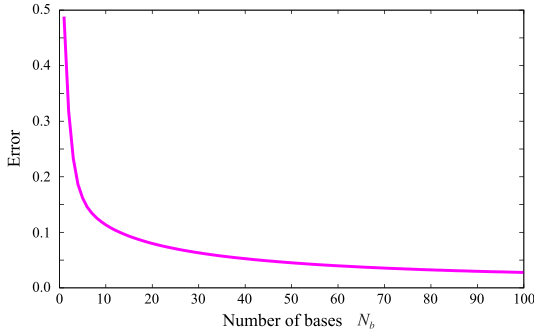
end for
return $\{\beta_i\}_{i=1}^{N_b}, \{\mathbf{b}_i\}_{i=1}^{N_b}$

式 (2) に示すように特徴ベクトル $\mathbf{x}$ と重み $\mathbf{w}$ の内積であり，膨大な数の検出ウィンドウを処理すると時間を要するという問題がある．そこで，本研究では Hare らによって提案された近似計算法 [12] を利用して，実数ベクトル間の内積をバイナリコード間の内積に置き換えることで，線形 SVM の高速な識別処理を実現する．

線形 SVM の近似計算法では，重みベクトル $\mathbf{w}$ を実数ベクトル $\boldsymbol{\beta}$ と基底バイナリコード $\mathbf{b} \in \{-1, 1\}^D$ に分解する．Algorithm 1 に重みベクトル $\mathbf{w}$ の分解アルゴリズム [12] を示す．線形 SVM による識別器 $F(\mathbf{x})$ は，重みベクトル $\mathbf{w}$ を分解して求めた実数ベクトル $\boldsymbol{\beta}$ と基底バイナリコード $\mathbf{b}$ を用いることで， $F(\mathbf{x}) \approx f(\mathbf{x}) = \sum_{i=1}^{N_b} \beta_i \mathbf{b}_i^T \mathbf{x}$ と近似計算することができる．ここで，基底バイナリコード $\mathbf{b}$ を $\mathbf{b}^+ \in \{0, 1\}^D$ と $\bar{\mathbf{b}}^+$ に分解 ($\mathbf{b} = \mathbf{b}^+ - \bar{\mathbf{b}}^+$) することで，式 (4) に示すように線形 SVM の近似計算を内積 $\langle \mathbf{b}_i^+, \mathbf{x} \rangle$ とノルム $|\mathbf{x}|$ により計算することができる．

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{N_b} \beta_i \mathbf{b}_i^T \mathbf{x} \\ &= \sum_{i=1}^{N_b} \beta_i (\langle \mathbf{b}_i^+, \mathbf{x} \rangle - \langle \bar{\mathbf{b}}_i^+, \mathbf{x} \rangle) \\ &= \sum_{i=1}^{N_b} \beta_i (2 \langle \mathbf{b}_i^+, \mathbf{x} \rangle - |\mathbf{x}|) \end{aligned} \quad (4)$$

ここで，入力特徴ベクトル $\mathbf{x}$ が B-HOG 特徴量のよ

図 2 基底数 N_b による w の近似精度Fig. 2 Approximation accuracy of w by number of bases N_b .

うなバイナリコード $x \in \{0, 1\}^D$ である場合、内積 $\langle b_i^+, x \rangle$ とノルム $|x|$ は、それぞれ論理積とビットカウントからなるビット演算で計算することが可能である。このような、バイナリコード間の論理積は実数間の積に比べて高速に演算することができる。また、ビットカウントは SSE4.2 から CPU に直接実装されている POPCNT 関数を用いることで高速な演算が可能である。

3.2 基底数 N_b と近似計算結果

我々は、線形 SVM の近似計算過程における基底数 N_b と近似計算結果に着目する。図 2 は、近似計算の基底数 N_b と誤差の関係をグラフで表したものである。誤差は重みベクトル w を実数ベクトル β 及び基底バイナリコード b により近似した結果 $\sum_{i=1}^{N_b} \beta_i b_i$ と重みベクトル w との 2 乗誤差である。重みベクトル w の近似精度は、基底数 N_b が増加するにつれて高くなっていることがわかる。線形 SVM の識別精度を維持するためには、基底数 N_b を多く用いて識別器を近似計算する必要がある。その反面、高速な識別処理を行うためには基底数 N_b をできるだけ削減して、識別器を近似計算することが求められる。

図 3 は、データセットのサンプルを例に基底数 N_b による線形 SVM の近似計算結果をグラフ化したものである。図 3 より、線形 SVM の近似計算では基底数 N_b が少ない段階では、おおまかに識別器の出力値を算出している。基底数 N_b が多くなるにつれて、線形 SVM との誤差を埋めるように識別器の出力値が変化している。また、図 3 の二つのサンプルは基底数が少ない段階で、線形 SVM の関数マージンから離れた出力値となっている。精度の良い物体検出を行うためには、この関数マージン内で二つのクラス y を最も多く

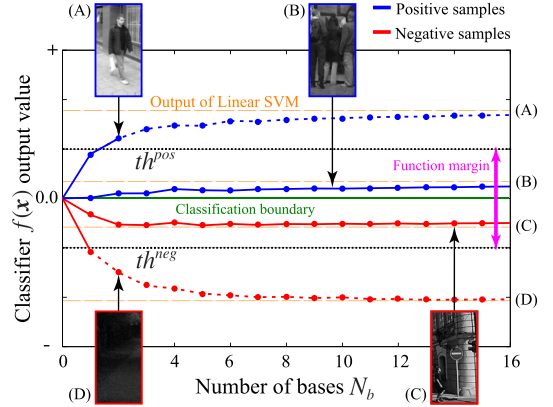


図 3 線形 SVM の近似計算結果

Fig. 3 Results of linear SVM approximation calculations.

分けるような識別器のしきい値 th を設定する。そのため、線形 SVM の近似計算時に、少ない基底数 N_b で関数マージンから離れた出力値をもつサンプルは、基底数 N_b を増加してもしきい値 th による識別結果が変わることはないといえる。

3.3 近似計算結果の早期判定による識別処理の高速化

基底数 N_b による識別精度と速度はトレードオフの関係にあり、膨大な検出ウィンドウを高速に処理するためには基底数 N_b をできるだけ削減したい。そこで、提案手法では関数マージンを利用して近似計算過程における計算結果から早期判定を導入することで、識別器の性能を維持したまま識別処理の高速化を行う。Algorithm 2 に近似計算結果の早期判定による識別処理の高速化アルゴリズムを示す。ここで、 th^{pos} 及び th^{neg} はそれぞれ +1 及び -1 とした。また、 $th^{pos} < f(x) < th^{neg}$ は関数マージン内を示しており、サポートベクタの出力値よりも識別境界に近い値を表している。図 3 に示すように、正と負の方向に大きな値を出力した場合に線形 SVM の近似計算を打ち切り、識別結果を早期判定することが可能となる。関数マージン内の識別が困難な入力サンプルは、多くの基底数 N_b を用いて線形 SVM との誤差の少ない近似計算を行い判定をする。これにより、線形 SVM の識別精度を維持しつつ、検出ウィンドウによっては早期判定することで高速な識別処理が可能となる。

3.4 バイナリコードにおける共起表現

本研究では、3.1 で述べたようにビット演算を利用して近似計算を高速化するために、HOG 特徴量

Algorithm 2 Fast discrimination by early judgment.**Require:** input image I

```

1. Raster scan the detection windows for image  $I$ 
for  $k = 1$  to  $K$  do //  $K$ : Total number of detection windows
2. Extract binary coded feature vectors  $\mathbf{x}_k$  from detection window  $I(k)$ .
3. Initialize  $f(\mathbf{x}_k) \leftarrow 0$ 
4. Obtain the classifier output value with the approximation calculation.
for  $i = 1$  to  $N_b$  do //  $N_b$ : Maximum number of bases
4.1 Linear SVM approximation calculation:  $\beta_i(2 < \mathbf{b}_i^+, \mathbf{x}_k > -|\mathbf{x}_k|)$ .
 $f(\mathbf{x}_k) += \beta_i(2 \text{ POPCNT}(\mathbf{b}_i^+ \& \mathbf{x}_k) - \text{POPCNT}(\mathbf{x}_k))$ 
4.2 Judge the calculation result.
if  $f(\mathbf{x}_k) > th^{pos}$  or  $f(\mathbf{x}_k) < th^{neg}$  then
break // End the approximation calculation.
end if
end for // Approximation calculation up to  $N_b$ 
5. Judge label  $y$  for the target from threshold  $th$ .
 $y_k = \begin{cases} 1 & \text{if } f(\mathbf{x}_k) > th \\ -1 & \text{otherwise} \end{cases}$ 
end for // End raster scanning.
return  $y_1, y_2, \dots, y_K$ 

```

をバイナリコード化した B-HOG 特徴量を利用する。B-HOG 特徴量は HOG 特徴量と比較して、2 値化の際の量子化誤差の影響により物体検出の識別精度が低下する問題がある。B-HOG 特徴量の高精度化としては、複数領域の勾配の関係を捉える R-HOG 特徴量が提案され、勾配の共起表現により高精度な物体検出を実現できることが報告されている [11]。そこで、提案手法では B-HOG 特徴量間のビット演算により共起性を表現することで、バイナリコードの特徴ベクトルによる識別精度の高精度化を行う。

3.4.1 HOG 特徴量の共起表現

実数ベクトルで表現される HOG 特徴量の共起表現による特徴表現の効果を確認するために予備実験を行う。HOG 特徴量の共起表現としては、Boosting アルゴリズム [13] を利用した手法 [14], [15] などが提案されている。本研究では、共起表現による特徴表現の有効性を確認するために、特徴量間で下記の式 (5)~式 (7) に示す演算により共起を表現した。このように、積、和、差の絶対値により HOG 特徴量を共起表現したものを HOG 特徴量の特徴量ベクトルに結合する。

$$\mathbf{V}_{c_i, c_j}^{mul} = \mathbf{V}_{c_i} \cdot \mathbf{V}_{c_j} \quad (5)$$

$$\mathbf{V}_{c_i, c_j}^{sum} = \mathbf{V}_{c_i} + \mathbf{V}_{c_j} \quad (6)$$

$$\mathbf{V}_{c_i, c_j}^{sub} = |\mathbf{V}_{c_i} - \mathbf{V}_{c_j}| \quad (7)$$

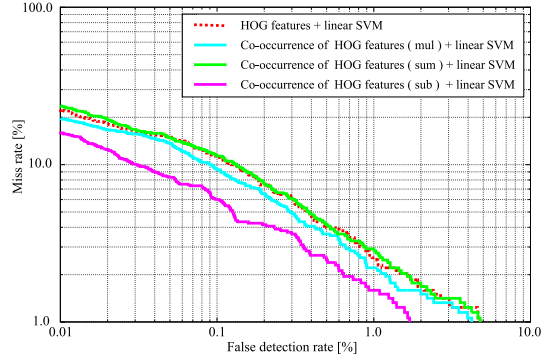


図 4 DET カーブ (HOG 特徴量の共起表現)

Fig. 4 DET curve (Co-occurrence of HOG features).

図 4 は、HOG 特徴量の共起の有効性を評価するために人検出による実験結果を DET カーブにより比較したものである。DET カーブより、実数ベクトルの HOG 特徴量では特徴量間の差の絶対値による共起表現は、高精度な物体検出が可能であることを確認した。

3.4.2 ビット演算を利用した B-HOG 特徴量の共起表現

3.4.1 で有効とされた共起表現をバイナリコードで実現する。提案手法によるバイナリコードの共起表現方法は、ブロックに属する二つのセルの B-HOG 特徴量 $\mathbf{P}_{c_i}$ と $\mathbf{P}_{c_j}$ から式 (8)~式 (10) に示すビット演算により勾配方向ヒストグラムにおけるバイナリコード間の関係性を表現する。ここで、論理演算子には論理積 (AND)、論理和 (OR)、排他的論理和 (XOR) のいずれかを用いる。

$$\mathbf{P}_{c_i, c_j}^{and} = \mathbf{P}_{c_i} \& \mathbf{P}_{c_j} \quad (8)$$

$$\mathbf{P}_{c_i, c_j}^{or} = \mathbf{P}_{c_i} | \mathbf{P}_{c_j} \quad (9)$$

$$\mathbf{P}_{c_i, c_j}^{xor} = \mathbf{P}_{c_i} \wedge \mathbf{P}_{c_j} \quad (10)$$

1 ブロックが 4 セルから構成されているとき、6 パターンの共起バイナリコード $\mathbf{P}^{operator} = \{\mathbf{P}_{c_1, c_2}, \mathbf{P}_{c_1, c_3}, \mathbf{P}_{c_1, c_4}, \mathbf{P}_{c_2, c_3}, \mathbf{P}_{c_2, c_4}, \mathbf{P}_{c_3, c_4}\}$ が生成される。このとき、論理演算子が AND の場合、二つのセルに共通して勾配が存在するときを“1”のビットで表現する。OR の場合、二つのセルでどちらかに勾配があるときを“1”のビットで表現し、二つのセルに共通して勾配が存在しないときを“0”のビットで表現する。XOR の場合、二つのセルに共通しない勾配を“1”のビットで表現し、二つのセルに共通して勾配が存在する及び存在しないときを“0”のビットで表現する。このように、ビット演算を利用して共起表現することでセル間

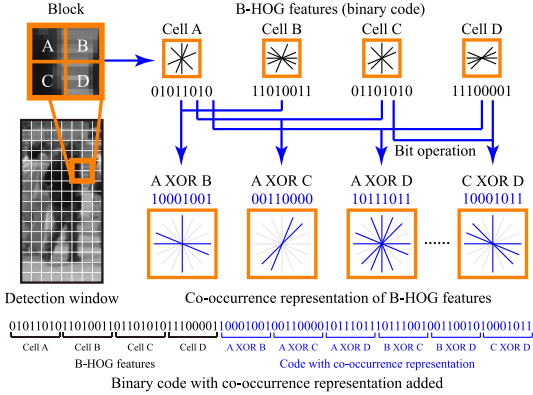


図 5 B-HOG 特徴量の共起表現

Fig. 5 Co-occurrence representation of B-HOG features.

の勾配の関係性を捉えることが可能となり、バイナリコードによる物体検出の高精度化が期待できる。バイナリコード間の共起には、ビット演算を用いるため低コストで共起表現をすることができる。

提案手法は図 5 に示すように、ブロック領域内のセルを組み合わせで共起表現したバイナリコードを B-HOG 特徴量のバイナリコードに追加する。共起表現を追加した特徴量は、 $M = 8$, $R = 2$, $N = 8$ のとき 3,360 ビットの B-HOG 特徴量に 5,040 ビットのバイナリコードを追加するため、8,400 ビットのバイナリコードとなる。提案手法では、この B-HOG 特徴量を共起表現した 8,400 ビットのバイナリコードを入力特徴ベクトル x として用いる。

4. 評価実験

提案手法の有効性を評価するために評価実験により識別精度及び識別過程の処理時間の比較を行う。

4.1 データセット

評価実験には、人検出のベンチマークとして広く利用されている INRIA Person Dataset [2] を使用する。学習サンプルには、ポジティブサンプルを 2,416 枚、ネガティブサンプルを 12,180 枚を使用する。識別精度を評価するためのテストサンプルには、ポジティブサンプルを 1,132 枚、ネガティブサンプルは背景画像 453 枚から網羅的にラスタスキャンさせて切り出した画像 1,306,029 枚を使用する。学習サンプル、テストサンプルの画像サイズはそれぞれ 64×128 ピクセルの大きさに正規化して使用する。また、識別器の処理時間を評価するための入力画像には、検出対象と非検

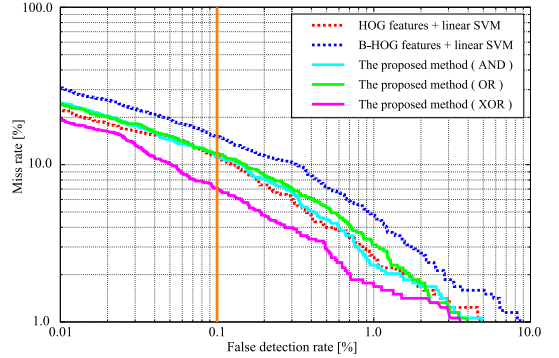


図 6 DET カーブ (提案手法の有効性の評価)

Fig. 6 DET curve (Effectiveness of proposed method).

出対象を含む INRIA Person Dataset の評価用画像を 288 枚使用する。

4.2 実験概要

提案手法の有効性を評価ために、以下の特徴量と識別器の組み合わせについて識別精度を比較する。

- HOG 特徴量+線形 SVM
- B-HOG 特徴量+線形 SVM
- 提案手法 (“論理演算子”)

B-HOG 特徴量の共起表現と線形 SVM の近似計算による早期判定

提案手法は、B-HOG 特徴量を共起表現するための論理演算子に、AND, OR, XOR のそれぞれを評価して比較する。評価には Detection Error Trade-off (DET) カーブを用いる。DET カーブは横軸に誤検出率、縦軸に未検出率を表しており、原点に近いほど高精度であることを表す。処理時間の比較では、OS: Windows Server 2008 Enterprise x64, CPU: Intel Xeon CPU X7542 @ 2.67GHz, RAM: 256GB の PC を用いる。全ての実験において線形 SVM は SVM Light [16] を用いて学習を行う。各特徴量のパラメータは $M = 8$, $R = 2$, $N = 8$ とする。また、評価実験では予備実験の結果より、線形 SVM の近似計算における基底数 $N_b = 16$ とする。

4.3 識別精度の評価

実験結果の DET カーブを図 6 に示す。DET カーブより、線形 SVM を用いた B-HOG 特徴量は HOG 特徴量と比較して識別精度が低下している。一方、提案手法 (AND), 提案手法 (OR) は B-HOG 特徴量と同様のバイナリコードの特徴量であるが、HOG と同等の識別率を得た。更に、提案手法 (XOR) は誤検出率 0.1% において HOG 特徴量+線形 SVM と比較し

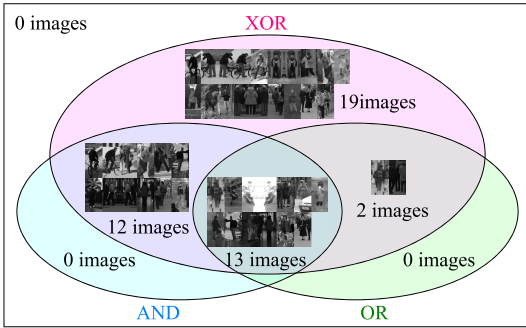


図 7 検出可能になったポジティブサンプル
Fig. 7 Positive sample that was detected.

表 1 各識別器の検出率 (DR) [%] と処理時間 (PT) [ms]
Table 1 Detection rate (DR) [%] and processing time (PT) [ms] for each classifier.

Classifier	DR	PT
SVM without appr. comp.	94.16	0.034
SVM with appr. comp. ($N_b=2$)	91.07	0.002
SVM with appr. comp. ($N_b=16$)	94.08	0.013
SVM with appr. comp. and early judg.	94.08	0.002

て約 6.1% 向上させることができた。これは、ビット演算後のバイナリが出力される割合に起因するためである。AND と OR により演算して求めたバイナリコードの“0”と“1”の割合は偏りが発生するため、ポジティブとネガティブの両方で同じバイナリが観測されることが多い。しかし、XOR のバイナリコードの“0”と“1”の割合は等しく無駄のないバイナリコードが生成できるためだと考えられる。

XOR による B-HOG 特徴量の共起表現の有効性を実験的に確認する。図 7 に B-HOG 特徴量で未検出のテストサンプルに対して、共起表現により検出可能となった全てのポジティブサンプルを示す。XOR で共起表現した場合、AND と OR で検出可能となった 27 枚と、AND と OR では検出することが不可能な 19 枚のサンプルを検出できたことがわかる。

4.4 処理速度の評価

表 1 に、XOR を用いて共起表現した特徴ベクトルを用いたときの検出率と一つの検出ウィンドウの識別に要する処理時間を示す。近似計算法は基底数 $N_b = 16$ のとき、線形 SVM と同等の識別精度である。処理時間を比較すると、近似計算法は線形 SVM よりも約 3 倍高速な演算が可能である。提案手法は早期判定を行うため近似計算法よりも更に速く、線形 SVM と比較して約 17 倍高速な識別器の演算を実現した。このと

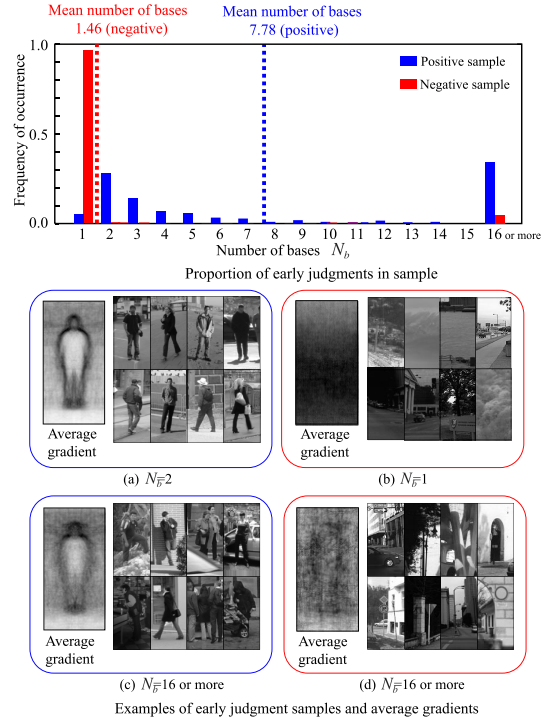


図 8 識別結果が早期判定されたサンプル
Fig. 8 Samples for which early judgment was made in the classification results.

き、提案手法は平均でポジティブサンプルは 7.78 個、ネガティブサンプルは 1.46 個の基底数による近似計算結果から早期判定している。基底数 $N_b = 2$ のときの近似計算法は、提案手法と同じ処理時間となるが検出率は低下する。

VGA サイズ (640×480 ピクセル) の入力画像に対して検出ウィンドウのサイズ $(l_w, l_h) = (64, 128)$ をマルチスケール $S = 2 (l_s = 1, 2)$ でずらし幅 $l_m = 4$ としてラスタスキャンした場合、全ての検出ウィンドウの識別処理に要する時間は、線形 SVM では $674.2[\text{ms}](1.48[\text{fps}])$ なのに対して、提案手法は $39.66[\text{ms}](25.21[\text{fps}])$ と高速にラスタスキャンをすることが可能となる。

図 8 に提案手法により識別結果が早期判定されたサンプルの割合とそのサンプル例を示す。基底数が少ない段階では、早期判定されたサンプルの平均勾配画像 (a), (b) から確認できるように、人と背景に分離しやすいサンプルであることがわかる。一方、困難なサンプル (c), (d) は最大基底数まで ($N_b = 16$) 継続して近似計算が行われる。また、ネガティブサンプルの 9

表 2 各特徴量のメモリと検出率, 処理時間
Table 2 Feature memory size, detection rate, and processing time.

Features	Dimensionality	Memory [bytes]	Detection rate[%]	Processing time[ms]
HOG features	3360	3360	88.69	1.554
B-HOG features	3360	420	84.89	1.561
Proposed method (XOR)	8400	1050	94.08	1.576
Random Projection	3360	420	86.57	50.989
	8400	1050	87.63	128.059
	23520	2940	88.69	360.219

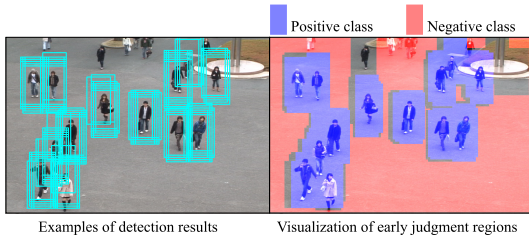


図 9 検出結果例と早期判定された領域の可視化
Fig. 9 Examples of detection results and visualization of early judgment regions.

割以上は基底数が 1 の時点で早期判定することで、高速な識別処理が可能となる。一般的な物体検出シーン(入力画像)では、多くの領域が背景(非検出対象)であるため、図 9 に示すように提案手法は多くの検出ウィンドウにて少ない基底数 N_b で線形 SVM を近似計算していることがわかる。以上より、提案手法は検出ウィンドウに応じて適応的に識別結果を早期判定することで、線形 SVM の性能を維持しつつ高速な識別処理を実現した。

4.5 特徴量のメモリサイズと処理時間の評価

提案手法のバイナリコードによる特徴表現に必要なメモリ量を調査する。ここでは、特徴ベクトルの距離関係を保持したままバイナリコード化することのできる Random Projection [17] を用いて、提案手法の共起表現した特徴量のメモリサイズを評価する。図 10 に、Random Projection により HOG 特徴量を任意のビット数のバイナリコードで空間写像した際の識別精度を示す。Random Projection では、実数ベクトルの HOG 特徴量と同等の識別精度を再現するには 23,520 ビット必要である。一方、表 2 に示すように、提案手法 (XOR) は検出率を約 6.1% 向上させつつそのビット数は 8,400 ビットであり、効率の良いバイナリ表現ができているといえる。更に、提案手法 (XOR) の共起抽出に要する時間は 1.576[ms]、HOG 特徴量は 1.554[ms] とわずかな時間が増えるだけで特徴抽出を行うことができた。

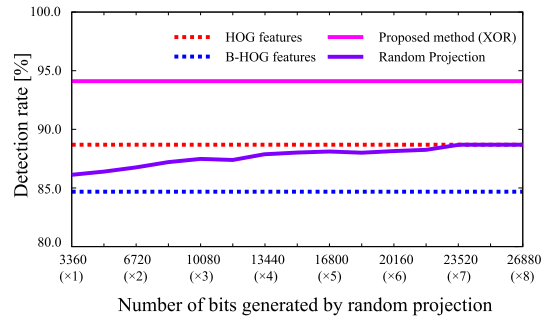


図 10 Random Projection の識別精度
Fig. 10 Random projection classification accuracy.

5. む す び

本論文では、物体検出における識別過程の高速化について取り組み、本研究の貢献は以下の 2 点である。1. 線形 SVM に早期判定を導入することによる識別処理の高速化。2. B-HOG 特徴量を共起表現したバイナリコードの特徴量による物体検出の高精度化。1. では、線形 SVM を近似計算し、検出ウィンドウに応じて適応的に識別結果を早期判定することで、識別過程において識別精度を維持しつつ約 17 倍高速な識別処理を実現した。2. では、B-HOG 特徴量のバイナリコードに対してビット演算を用いた共起表現により高速かつ効率の良いバイナリコード化を行い、メモリサイズを約 1/3 にしつつ識別精度の高精度化を実現した。今後は、HOG 特徴量の高速化手法を提案手法に導入して特徴抽出過程を高速化することで、検出手法全体において高速な物体検出を実現する予定である。

文 献

- [1] C. Cortes and V. Vladimir, "Support-vector networks," Machine Learning, vol.20, no.3, pp.273–297, 1995.
- [2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Computer Vision and Pattern Recognition, vol.1, pp.886–893, 2005.
- [3] X. Wang, T.X. Han, and S. yan, "An HOG-LBP human detector with partial occlusion handling," Inter-

- national Conference on Computer Vision, pp.32–39, 2009.
- [4] C. Wojek, S. Walk, S. Roth, and B. Schiele, “Monocular 3D scene understanding with explicit occlusion reasoning,” *Computer Vision and Pattern Recognit.*, pp.1993–2000, 2011.
- [5] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.32, no.9, pp.1627–1654, 2010.
- [6] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng, “Fast human detection using a cascade of histograms of oriented gradients,” *Computer Vision and Pattern Recognition*, pp.1491–1498, 2006.
- [7] V. Prisacariu and I. Reid, “Fast human detection with cascaded ensembles on the GPU,” *Intelligent Vehicles Symposium*, pp.325–332, 2010.
- [8] V. Prisacariu and I. Reid, “fastHOG - a real-time GPU implementation of HOG,” *Department of Engineering Science*, no.2310/9, 2009.
- [9] Y. Freund and R.E. Schapire, “Experiments with a new boosting algorithm,” *International Conference on Machine Learning*, pp.148–156, 1996.
- [10] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Computer Vision and Pattern Recognition*, vol.1, pp.511–518, 2001.
- [11] 松島千佳, 山内悠嗣, 山下隆義, 藤吉弘亘, “物体検出のための Relational HOG 特徴量とワイルドカードを用いたバイナリーのマスキング,” *信学論 (D)*, vol.J94-D, no.8, pp.1172–1182, Aug. 2011.
- [12] S. Hare, A. Saffari, and P.H.S. Torr, “Efficient online structured output learning for keypoint-based object tracking,” *Computer Vision and Pattern Recognition*, pp.1894–1901, 2012.
- [13] R.E. Schapire and Y. Singer, “Improved boosting algorithms using confidence-rated predictions,” *Machine Learning*, vol.37, no.3, pp.297–336, 1999.
- [14] 三井相和, 山内悠嗣, 藤吉弘亘, “Joint 特徴量を用いた 2 段階 Boosting による物体検出,” *信学論 (D)*, vol.J92-D, no.9, pp.1591–1601, Sept. 2009.
- [15] 山内悠嗣, 山下隆義, 藤吉弘亘, “Boosting に基づく特徴量の共起表現による人検出,” *信学論 (D)*, vol.J92-D, no.8, pp.1125–1134, Aug. 2009.
- [16] T. Joachims, SVM light, <http://svmlight.joachims.org>
- [17] M.X. Goemans, “Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming,” *J. ACM*, vol.42, pp.1115–1145, 1995.

(平成 25 年 4 月 16 日受付, 8 月 28 日再受付)



後藤 雄飛 (正員)

2012 年画像センシングシンポジウムオーディエンス賞。2013 年中部大学大学院博士前期課程修了。2013 年より株式会社ホンダエリシス入社。同年、画像センシングシンポジウム優秀学術賞。



土屋 成光 (正員)

2008 年中部大学大学院博士前期課程修了。2011 年同博士課程満期退学。2011 年より同大学研究員、パターン認識・理解、機械学習の研究に従事。



山内 悠嗣 (正員)

2012 年中部大学大学院博士後期課程修了。2012 年より同大学院博士研究員。2010 年独立行政法人日本学術振興会特別研究員 DC。コンピュータビジョン、パターン認識の研究に従事。



藤吉 弘亘 (正員)

1997 年中部大学大学院博士後期課程修了。1997–2000 年米カーネギーメロン大学ロボット工学研究所 Postdoctoral Fellow。2000 年中部大学講師, 2004 年同大准教授を経て 2010 年より同大教授。2005–2006 年米カーネギーメロン大学ロボット工学研究所客員研究員計算機視覚, 動画処理, パターン認識・理解の研究に従事。2005 年ロボカップ研究賞。2009 年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優秀論文賞, 2009 年山下記念研究賞。2010 年, 2013 年画像センシングシンポジウム優秀学術賞。2013 年電子情報通信学会情報・システムソサイエティ論文賞。博士 (工学)。情報処理学会, 電子情報通信学会, 電気学会, IEEE 各会員。