検出対象をリコンフィグ可能な Joint-HOG による FPGA ハードウェア検出器 FPGA Hardware with Target-Reconfigurable Object Detector by Joint-HOG

矢澤 芳文 † 吉見 勤 † 都筑 輝泰 † 土肥 智美 † 藤吉 弘亘 ‡

Yoshifumi YAZAWA†, Tsutomu YOSHIMI†, Teruyasu TSUZUKI†, Tomomi DOHI†, Hironobu FUJIYOSHI‡

†三栄ハイテックス株式会社 ‡ 中部大学 工学部情報工学科

†SANEI HYTECHS co., ltd. ‡Department of Computer Science, Chubu University

E-mail: yazawa.yoshifumi@sanei-hy.co.jp, post@sanei-hy.co.jp

Abstract

近年,統計的学習方法と局所領域の low-level な特徴量 を用いた人や車両などの物体検出の研究が取り組まれて いる. これらは車載・監視カメラなどの組み込み用途で の利用が望まれている. なかでも複数の low-level な特徴 である Histograms of Oriented Gradients(HOG) 特徴 を組み合わせることで、「連続的な形状」や「対称的な形 状」を捉えることが可能である Joint-HOG 特徴量が物 体検出に有効と報告されている.本稿では、Joint-HOG 特徴量の「高速化」と,検出対象の「柔軟性」を目的と したハードウェアアーキテクチャを提案する. HOG 特 徴量算出・識別演算部と Joint-HOG 学習結果を分離し たハードウェアアーキテクチャの設計を行った. FPGA ボードに実装し動作確認したところ,実用的な時間で動 作させることができた. さらに事前学習結果を差し替 えることで"人"や"車両"などの、検出対象を動的に 変更することが可能であることを確認できた.

1 はじめに

近年, セキュリティやマーケティングなどの分野で, 画像を用いた物体認識に関する研究が多くなされてい る [1]-[3]. 中でもカメラ映像から画像認識技術により 人や車両の検出を行い, 交通事故防止を目的とした車載 カメラといった用途での使用が望まれている [4],[5]. そ のような用途の場合, 実時間での高精度な認識が求めら れ, ハードウェア化の研究が行われている [5],[6]. ハー ドウェア化の際に演算を最適化する場合, 検出対象を固 定あるいは制限することが多く, 検出器の柔軟性を損な う場合がある.

本研究では、物体検出に有効とされる Joint-HOG[3] に注目する. これは、複数の low-level な特徴である HOG 特徴を 2 段階の Real AdaBoost により組み合わ せた手法である. 異なる 2 つの領域の HOG 特徴を使用 することで、単一の特徴量で表現が困難な

- 1. 連続的な形状
- 2. 左右対称的な形状

を自動的に捉えることが可能である.

本稿では、物体検出に有効とされる Joint-HOG[3] の 「高速化」と、検出対象の「柔軟性」を目的としたハー ドウェアアーキテクチャを提案する.検出対象に適し たウィンドウ形状にて HOG 特徴量を演算し、事前学習 結果を LUT で構成した識別器により検出を行う.LUT 内容を書き換えることで、柔軟に検出対象を変更可能で ある.また、各演算ステージをパイプライン状に処理し、 ウィンドウ毎の並列化と組み合わせて高速化した.

2 Joint-HOG による物体検出

Joint-HOG 特徴の学習の流れを図1に示す.本手 法は、2段階の Boosting により最終識別器を構築す る.Boosting には弱識別器の出力値が連続量で得られ る Real AdaBoost を用いる.まず, HOG 特徴プールを 入力とした1段階目の Real AdaBoost により位置の異 なる2つの low-level な特徴である HOG 特徴を組み合 わせた Joint-HOG 特徴プールを作成する.Joint-HOG 特徴は複数の HOG 特徴を同時に観測するため、単一の HOG 特徴のみでは捉えることのできない、対称性や連 続的なエッジを表現することが可能である.次に、2段 階目の Real AdaBoost により Joint-HOG 特徴プール から自動的に物体検出に最適な Joint-HOG 特徴を選択 し、最終識別器により物体の検出を行う.

2.1 low-level 特徵: HOG

本研究では、low-level な特徴として Dalal らが提案 した Histograms of Oriented Gradients(HOG)[1]を用 いる. HOG 特徴は CELL(図 2(b)(5×5 ピクセル))にお ける勾配方向をヒストグラム化した特徴であり、物体の 形状を表すことが可能である. HOG 特徴は照明の変化 を受けにくく、局所的な幾何学変化に頑健な特徴とな る. 下記に HOG 特徴算出の手順を示す.

まず, 各ピクセルの輝度 *L*から *x* 方向, *y* 方向の輝度 差分を求めた後, 勾配強度 *m* と勾配方向 *θ* を次式より



図1 Joint-HOG による学習の流れ



図2 HOG 特徴量の算出 (検出対象が人の場合)

算出する.

$$m(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2}$$
(1)

$$\theta(x,y) = \tan^{-1} \frac{f_y(x,y)}{f_x(x,y)}$$
(2)

$$\begin{cases} f_x(x,y) = L(x+1,y) - L(x-1,y) \\ f_y(x,y) = L(x,y+1) - L(x,y-1) \end{cases}$$
(3)

次に,算出した勾配強度 m と勾配方向 θ より CELL (5×5 ピクセル)ごとに輝度の勾配方向ヒストグラムを 作成する.勾配方向を 20°ずつに分割することにより, 9 方向の勾配方向ヒストグラムを得る.

最後に,次式によりブロック領域毎 (図 2(c)) に特徴 量を正規化する.

$$v(j) = \frac{f(j)}{\sqrt{\left(\sum_{i=0}^{k} f(j)\right)^2 + \epsilon}} \qquad (\epsilon = 1) \qquad (4)$$

ここで、vは HOG 特徴量、k はブロック内の HOG 特 徴量の数、 ϵ は分母が 0 の場合に計算不能になること を防ぐ係数である.

2.2 HOG 特徴量の共起

Joint-HOG 特徴を生成するために, 複数の HOG 特 徴量により共起を表現する, まず, 次式から対象物体も しくは非対象物体を表す2値化符号 *s*を算出する.

$$s(\mathbf{V}) = \begin{cases} 1 & p \cdot v_o > p \cdot \theta \\ 0 & \text{otherwise} \end{cases}$$
(5)

ここで、 θ はしきい値、pは不等号の向きを決定する符 号である. $\mathbf{V} = [v_1, v_2, \cdots v_9]$ は1つのセルから算出さ れる特徴量、oは勾配の方向を表す、これにより、得ら れた2値化符号を2つ組み合わせることで共起を表現 した特徴jを得る。例えば、図3のような入力画像に おいて、HOG 特徴を2値化した符号 $s_1 = 1$, $s_2 = 1$ を 観測したとき、共起を表現した特徴jは $j = (11)_2 = 3$ となる。共起を表現した特徴jは2進表現された特徴 の組み合わせのインデックス番号であり、今回は2つの 特徴の組み合わせとなるため、4 通りの値をとる。



図3 HOG 特徴量の共起 (検出対象が人の場合)

2.3 mid-level 特徵: Joint-HOG 特徵

ここまでに算出した HOG 特徴量の共起を表現した特 徴量を組合わせて mid-level な特徴量として表現する. 2.2 にて算出した HOG 特徴の共起を用いて, 2 つのセ ルの low-level 特徴から算出した共起を表現した特徴量 と1段階目 Real AdaBoost により Joint-HOG 特徴を 生成する.これにより,物体形状の対称性やエッジの連 続性だけでなく,セルとセルの関係を捉えることが可能 となる.

まず、位置の異なる2つのセル*cm、cn*において、共 起を表現した特徴の中から、識別に有効な特徴を選択 する.入力画像*x*から HOG 特徴の共起を観測する関 数を $J_t(x)$ で表す.入力画像*x*からの特徴量 $J_t(x) = j$ を観測したとき、1 段階目の Real AdaBoost の弱識別 器 $h_t(x)$ を次式により表す.

$$h_t(x) = \frac{1}{2} \ln \frac{P_t(y=+1|j) + \epsilon}{P_t(y=-1|j) + \epsilon}$$
(6)

ここで, t は学習回数, ϵ は計算不能になることを防ぐた めの係数である. $P_t(y = +1 \mid j)$ 及び $P_t(y = -1 \mid j)$ は, それぞれ HOG 特徴の共起を表現した特徴 j を観測 したときの条件付き確率である. 条件付確率は, 学習サ ンプル i の重み $D_t(i)$ に基づき次式により算出される.

$$P_t(y = +1|j) = \sum_{p: J_t(x_p) = j \land y_i = +1} D_t(i) \quad (7)$$

$$P_t(y = -1|j) = \sum_{p:J_t(x_p) = j \land y_i = -1} D_t(i)$$
 (8)

$$D_{t+1}(i) = D_t(i) \exp[-y_i h_t(x_i)]$$
 (9)

1 段階目の強識別器である Joint-HOG 特徴 *H_{cm, cn}(x)* を次式より構築する.

$$H_{cm,cn}(x) = \sum_{t=1}^{T} h_t^{cm, cn}(x)$$
(10)

ここで,*T*は1段階目の学習回数を示す。上記の処理 を全てのセルの組み合わせにおいて行う.

作成した Joint-HOG 特徴プールから、2 段階目の Real AdaBoost により最終識別器を構築する.まず、この Joint-HOG 特徴プールを入力とする弱識別器出力 $g_t(c)$ を次式より算出する.

$$g_t(c) = \frac{1}{2} \ln \frac{W_+^k + \epsilon}{W_-^k + \epsilon} \tag{11}$$

c はセルの組み合わせを表す通し番号, ϵ は分母が0の 場合に計算不能になることを防ぐための係数である.こ こで, W_+, W_- はポジティブクラスとネガティブクラス の確率密度関数であり, 次式より算出される.

$$W_{+}^{k} = \sum_{i:k \in K \land y_{i}=+1} D_{t}(i)$$
(12)

$$W_{-}^{k} = \sum_{i:k \in K \land y_{i}=-1} D_{t}(i) \tag{13}$$

 $D_t(i)$ は学習サンプルの重みであり、1段階目と同様に式 (9)を用いて作成される. 最後に、次式より最終的な 強識別器 G(c)を得る.

$$G(c) = \begin{cases} 1 & \sum_{t=1}^{T} g_t(c) > \lambda \\ 0 & \text{otherwise} \end{cases}$$
(14)

 λ は検出器のしきい値である.2段階目の Real Ada-Boost により、Joint-HOG 特徴量プールから識別に有 効な特徴量のみを選択された強識別器が得られる.

3 ハードウェアアーキテクチャ

図4にハードウェアアーキテクチャを示す.以下の 4段階の演算ステージによって,入力画像(図中左画像) からHOG特徴量を算出し識別演算を行う.浮動小数点 型の算術演算ライブラリを使用することで,ソフトウェ アアルゴリズムから精度を落とすことなく演算するこ とができる.

設計に際して,以下の点に注目してハードウェア化を 行った.

- 高速化
 - パイプライン処理 (各演算サイクルの最適化)複数ウィンドウの並列演算
- 柔軟性
 - ウィンドウサイズ可変
 - 事前学習結果の LUT 化

演算ブロック内容を表1に示す.各ブロックでは専用の 小規模メモリを複数活用することで,演算サイクルの最 適化を図った.これにより,演算待ち時間を極力減らす ことができた.

表1 演算ブロック

ブロック名	機能	出力
MG	輝度勾配演算	画素毎の勾配強度/方向
CPHIST	ヒストグラム演算	CELL ヒストグラム
HOGNRM	正規化演算	正規化ヒストグラム
CLSF	識別演算	最終識別結果

3.1 HOG 特徵量演算

輝度画像データを入力として MG/ CPHIST/ HOG-NRM の各モジュールを経ることで,式 (1)-(4) の演算 を行う.また,検出対象を柔軟に変更可能とするために, 図5に示す4種類の検出ウィンドウに対応して HOG 特 徴を計算する.また,検出対象例とともに,検出ウィン ドウ (CELL 数) に対する CELL 組み合わせ数を示す. Joint-HOG アルゴリズムでは, すべての CELL の組み



図4 ハードウェアアーキテクチャ

合わせから検出に最適な特徴を選択して識別器を構築 する.



図5 検出ウィンドウ

MG モジュールでは、画像ウィンドウの輝度情報から 一次微分を行い、輝度勾配ベクトル (勾配強度 m、勾配 方向 θ)を算出する.1ピクセルあたりの輝度勾配ベク トルを算出するために上下左右の4ピクセルを使用す るが、1ピクセル毎にデータを扱うことをせずに、上下 の3ライン分をFIFOにて管理し、シフトレジスタと組 み合わせデータ処理する.これにより、ほぼ1ウィンド ウ分のデータ読み出しサイクル数で輝度勾配ベクトル の算出が可能となる.

次に CPHIST モジュールにて, 算出した勾配強度 mと勾配方向 θ から CELL (5×5 ピクセル) ごとに輝度の 勾配方向ヒストグラムを作成する.これは, 勾配方向 θ ごとの配列メモリに勾配強度 mを累積加算することで 算出する.これもほぼ 1 ウィンドウ分のデータ読み出 しサイクル数にて, CELL 数 × 方向数の次元量のデー タとして出力する.

最後に HOGNRM モジュールでブロック領域毎に正 規化を行う.式(4)に示す二乗演算,ブロック領域ごと の累積加算,平方根演算,除算の各処理を経て正規化さ れた HOG 特徴量を算出する.





表 2 事前学習結果

cm, cn	有効な CELL の組み合わせ	
p	不等号決定用符号	
heta	HOG の共起しきい値	
$P_t(y = +1 j), P_t(y = -1 j)$	1段階目確率密度関数	
$W_+, \ W$	2段階目確率密度関数	

図 6 に CLSF モジュールでの演算概略図を示す. HOGNRM モジュールで算出した HOG 特徴量から検 出に最適な CELL の組み合わせを使用し,2 段階 Real AdaBoost 処理から TRUE/FALSE を判定する.表2 で示した事前学習結果を LUT 化し, Joint-HOG 特徴に よる識別を行う.これにより,事前学習にて選択した検 出対象ごとに最適な特徴量の組み合わせでの識別演算 が可能となる.また,LUT 化することで,学習結果の差 し替えが容易となり,検出対象を人・車両と変更可能と なる.



図7に識別結果出力までの流れを示す. MG - CLSF のそれぞれのモジュールはほぼ同じ演算時間で処理す るように設計することができたため,パイプライン状に 構成することで,後段モジュールの出力をまたずに演算 可能である.

4 評価

提案ハードウェアを Verilog-HDL で記述し論理設計 を行った.これを論理シミュレーションし,その出力が ソフトウェアアルゴリズム出力と一致することを確認 した.その上で,FPGA ボードに実装し,カメラ入力か ら検出結果出力までの動作確認を行った.

4.1 ソフトウェア処理時間

比較対象として、ハードウェアと同じ Joint-HOG ソ フトウェアによる検出処理時間を測定した.

計測環境としては, Core2 Duo @2.33GHz, メモリ 1.95GB, OS は Windows XP SP3 である. 検出パラ メータとともに, ソフトウェア処理時間を表 (3) に示す.

入力画像	640×480
検出ウィンドウ (ピクセル)	40×80
検出ウィンドウ (CELL)	6×12
スキャン間隔	縦, 横方向 20 ピクセル
スケールステップ	0.1
マルチスケール回数	5
ウィンドウ数	2940
2940 ウィンドウ処理時間	77.30ms(約 13fps)

表3 ソフトウェア処理時間

4.2 画像処理 FPGA ボード

図8に使用した画像処理 FPGA ボードを示す. Altera 社製 Cyclone III FPGA を使用し,カメラリンク入力さ れた画像データに対して検出演算を行う. このとき,1 画面データについて検出ウィンドウごとのラスタスキャ ン処理にて, HOG 特徴量演算・識別を行い,検出結果を 画像データとともに USB 経由で PC に転送する. PC で受信した検出座標は, Mean Shift クラスタリングに よるウィンドウの統合処理を行い [7],最終的なウィン ドウ座標として表示される. また,図6に示した LUT データは USB 経由にて PC から書き換え可能である.

4.3 実装結果

Cyclone III FPGA をターゲットとして合成した結果 (検出回路1個分)を1画面 (2940 ウィンドウ)の処理時 間とともに表4に示す.1ウィンドウ分の検出器を比較 的小規模な回路としてインプリメントできた.

また,図7に示すパイプライン処理により,ソフト ウェア処理時間(3)に及ばないものの,1画面(2940ウ



図 8 画像処理 FPGA ボード

表 4 実装結果 (Cyclone III EP3C120)

総 LE 数	17,419(15%)
総レジスタ数	11,306(9%)
内部メモリ bit 数	1,046,647 bit(26%)
動作周波数	70MHz
2940 ウィンドウ処理時間	93.95ms(約 10fps)
2940 ウィンドウ処理時間 (2 並列動作)	46.98ms(約 20fps)

インドウ)の検出演算をほぼ実用的な時間内で実行できた (約 10fps).

ただし、ハードウェアリソースの許す限り提案回路の 複数実装が可能であり、これによる並列動作での高速 化が可能となる.例として2並列とした場合、ソフト ウェア処理時間を上回る約20fpsでの検出演算を確認で きた.

4.4 動作例

=

図9に学習に用いたデータベース例を示す.表5に示 す画像データベースを用いて,人/車両それぞれの学習 を行った.ここで人を対象とした場合には6×12CELL, 車両を対象とした場合には9×9CELLの検出ウィンド ウをそれぞれ使用した(図5). 得られた事前学習結

長5 学習データベース

衣 5 子白ノ · ク · 、 · ヘ				
検出対象	ポジティブサンプル	ネガティブサンプル		
人	2,054 枚	6,258 枚		
車両	710 枚	8,860 枚		

果 LUT データ (約 360Kbit) を FPGA 内部メモリへ書 き込み,画像データからの検出動作実験を行った.図 10 に動作例を示す.事前学習結果を差し替えることで, "人"/"車両"それぞれを検出することができた.

学習結果を容易に差し替え可能であることは、検出器





(a) 人検出例





(b) 車両検出例







(b) 単画字音テーダベース 図 9 学習に用いたデータベース

の最適化を行う場合にも有用である.異なるシーンでの 人検出実験を行った動作例を図 11 に示す.図 11(a) は 表5の学習結果による検出例だが,このシーンでは背景 の樹木の枝を誤検出している.そこで,樹木をネガティ ブサンプルとして追加し再学習したところ,誤検出を大 幅に削減することができた(図 11(b)).

5 おわりに

本稿では、Joint-HOG 特徴を利用したハードウェア 検出器を提案した.事前学習した結果を FPGA 内部の LUT に持ち、これを差し替えることで、検出対象に応じ て最適な特徴量の組み合わせを使用した検出器が構築 することができる.提案ハードウェアを FPGA ボード に実装し、"人"/"車両"と検出対象を変更可能であることを確認した.本検出器は1 画面処理を実用的な時間内で演算可能であるが、今後はさらなる高速化・小サイズ化を進める予定である.

参考文献

- N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", IEEE Computer Vision and Pattern Recognition, vol.1, pp. 886-893, 2005.
- [2] 山内 悠嗣,藤吉 弘亘, Hwang Bon-Woo,金出 武雄,"アピアランスと時空間特徴の共起に基づく人検出",画像の認識・理解シンポジウム (MIRU2007), pp. 1492-1497, 2007.
- [3] 三井 相和、山内 悠嗣、藤吉 弘亘、"Joint HOG 特徴 を用いた2 段階 AdaBoost による人検出"、第14回 画像センシングシンポジウム SSII08, IN1-06, 2008.
- [4] 高木 雅成,山内 悠嗣,三井 相和,藤吉 弘亘,"Geometric Context を用いた特徴量間の共起による物 体検出の高精度化",画像の認識・理解シンポジウ ム (MIRU2009), pp. 643-650, 2009.
- [5] 門田 亮二, 中村 行弘, 宮本 龍介, "実時間歩行者認 識に向けた HOG 特徴量のハードウェア実装", 電 子情報通信学会技術研究報告, pp.43-48, 2009.
- [6] 山中悠歩,山崎俊彦,相澤清晴, "FPGA による histogram of oriented gradients アルゴリズムの高速



(b)シーンに最適化した学習データ使用

図 11 再学習による学習データアップデート

化",情報処理学会創立50周年記念全国大会,2010.

[7] D. Comaniciu and P. Meer, "Mean Shift Analysis and Applications", IEEE International Conference on Computer Vision, pp. 1197-1203, 1999.