

車載カメラを用いた自転車検出システムの研究

著者	鄭 ?旭
発行年	2013
その他のタイトル	A Study on a Bicycle Detection System Employing Car Vision
学位授与年度	平成25年度
学位授与番号	17104甲工第366号
URL	http://hdl.handle.net/10228/5292

博士論文

車載カメラを用いた
自転車検出システムの研究

Ph. D. Thesis

A Study on a Bicycle Detection System
Employing Car Vision

平成 25 年度
九州工業大学大学院 工学府
機械知能工学専攻 知能制御工学コース
博士後期課程

11584204

鄭 燾旭

指導教員 石川 聖二 教授

摘要

近年、物体認識技術の発展によって車載カメラを用いた先進安全自動車システムが開発され、商用化している。しかし現在のシステムは、カメラだけでなく他のセンサーも用いたシステムであるため高価であり、まだ技術的にも解決すべき問題点が多い。またこれらのシステムは、歩行者の検出が中心である。交通事故を減らすためには人以外、例えば、自転車、自動二輪車、車などの検出が必要である。特に、多くの交通事故が自転車と関連するものであるにもかかわらず、自転車の自動検出に関する研究はほとんど見られない。

そこで本研究は、自車の周囲を走行する自転車を、自車に装着されたカメラの映像から検出して運転者に知らせるシステムの実現を目標に置いて、そのようなシステムに必要な自転車検出法の開発を目的とする。本論文は自転車検出のための有効な特徴量を獲得する方法、及び自転車かどうかを判断する有効な識別器、さらに、オクルージョン問題の対策として、自転車車輪と人の上半身を部分的に検出して自転車検出を行う手法を提案する。

目次

第 1 章	序論	1
1. 1	研究背景と目的	1
1. 2	関連研究	2
1. 3	本論文の構成	3
第 2 章	MSC-HOG 特徴量を用いた自転車検出	4
2. 1	処理のながれ	4
2. 2	HOG 特徴量	5
2. 3	MSC-HOG 特徴量	8
2. 4	HSV 変換	11
2. 5	RealAdaBoost	11
2. 6	Cascade Structure	14
2. 7	検出環境の構築	15
2. 8	実験と評価	17
2. 9	考察	30
2. 10	まとめ	30
第 3 章	円を用いた自転車検出	32
3. 1	円と楕円検出	32
3. 2	上半身検出	43
3. 3	自転車輪と上半身の統合処理	48
3. 4	まとめ	52
第 4 章	検出器と追跡器を用いた自転車位置とスケールの推定	53
4. 1	処理のながれ	53
4. 2	Particle Filter を用いた物体追跡	53
4. 3	自転車の位置とスケール推定	56
4. 4	実験と評価	59
4. 5	考察	62
4. 6	まとめ	63
第 5 章	結論	64

参考文献

謝辭

第 1 章 序論

1.1 研究背景と目的

近年、物体検出と認識の研究が盛んに行われ、様々な分野に適用されている。エンターテインメント分野では顔認識やカメラコントローラを用いた技術 [4]、セキュリティ分野では、監視カメラを用いた物体検出と認識技術に用いられている [1-3]。様々な分野で人の活動を支援するロボットは数十年間盛んに研究が行われている。しかし、ロボットの視覚知能システムはそれほど進歩している訳ではない。人の目は物体認識において大きな役割を持つ。ロボットの視覚知能システムが人の視覚系のような機能・能力を持てば、ロボットはもっと広い分野で活用できるであろう。人のような視覚情報処理系の開発は、未来の知能型ロボットにとって核心的に重要な技術の一つである。

近年、物体認識技術が最も注目されている分野は先進安全自動車(ASV: Advanced Safety Vehicle)システムである [5]。このシステムは、車載カメラを用いて歩行者、車を検出することができる。現在、商用化されている代表的な先進安全自動車システムには、SUBARU の Eyesight, Volvo の Pedestrian Detection with Full Auto Break がある。SUBARU のシステムはステレオカメラを用いており、30 km/h 以内で走行している時に、前方の障害物を検出して停止することができる。また他の機能として、前車との距離を認識することができる。Volvo のシステムは広角カメラとレーダーを用いており、35km/h 以内で走行している時に前方の障害物検出と停止が可能である。また他の機能として、前方の歩行者との距離を認識することができる。

実際の交通環境で物体の認識を行う場合、様々な物体の存在、複雑な背景、オクルージョン、画像劣化、照明変化等が認識を困難にする原因となる。このような問題のため、SUBARU や Volvo のシステムは、歩行者の未検出・誤検出を防ぐことができない。実際の交通環境では歩行者の安全のため、より高精度、より高速度の物体検出が要求される。

日本警察庁の調査資料によれば、平成 22 年度に国内で発生した交通事故の総数は 66 万 4,907 件であり、その中で自転車の交通事故が 14 万 4,018 件を占めている。様々な安全システムの開発により交通事故は目に見えて減っているが、最近 5 年間の交通事故では、自転車に関連する事故の割合は 20 % を超えている。その中では自転車対自動車の事故が最も多く、80% 以上を占めている。このような状況であるから、便利な交通手段で多くの人々が利用している自転車の事故の低減は、現代社会にあって緊要の課題である。近年、車載ビジョンの研究

が盛んに行われているが、ほとんどが歩行者検出法に関する研究[8-10]で、自転車検出のための研究はほとんど見られない。

以上のことから本論文は、自転車事故を減らすために、実際の交通環境で走行している自転車に注目し、車載カメラとコンピュータを用いた二つの自転車検出法を提案する。

1.2 関連研究

歩行者と異なり、自転車は見る方向によって形状が大きく変化するため、自動検出が困難になる。自転車検出の従来研究には、部分モデルを用いた自転車検出法 [6] やペダルの動きを用いた自転車検出法がある [7]。しかし、これらの手法は実際の交通環境における実験がないため、実際の交通環境下での確に自転車検出が行われるかどうかは未知である。実際の交通環境下での物体検出は、複雑な背景、オクルージョン、照明変化等を考慮する必要がある、的確な物体検出は困難な問題である。また実交通環境下では、高精度および高速な検出が要求される。高精度検出のためには、有効な特徴の選択に基づく識別器が必要である。また高速検出のためには、識別器の高速化が必要である。さらにオクルージョンの問題を解決するために部分認識法の導入が考えられる。

1.2.1 特徴抽出

物体から有効な特徴を抽出するための研究は盛んに行われている。SIFT(Scale Invariant Feature Transform)[11]およびHOG(Histograms of Oriented Gradients)[12]は物体認識によく使われる特徴抽出法である。SIFTはスケールの変化、照明変化、回転などに頑強で、画像マッチングや顔認識などによく用いられる。HOGは人のような形状の変化がある物体の検出や認識に頑強な特徴量であり、複雑な背景下の人物を検出する場合によく用いられる。しかし、これらの方法は以下の短所もある。SIFTは計算コストが高いという点、HOG特徴量はスケール変化や回転変化に弱いという点が挙げられる。従ってSIFTとHOG特徴量を改良した様々な方法が提案されている [13-15]。

1.2.2 分類器

SVM(Support Vector Machine)とAdaBoostは、2クラスの分類に有効な識別器として知られている [12-13, 16-17]。SVMは分離超平面とサポートベクトル間の距離であるマージンを最大化して、ふたつのクラスを分類する方法 [16]で、AdaBoostは汎化能力の低い多数の弱識別器を組み合わせて、汎化能力の高い強識別器を構築する方法である [18]。SVMは人検出に、またAdaBoostは顔検出や認識に用いられ、成功を収めている [12, 19]。また、さらに高性能を得るた

めに, SVM と AdaBoost は他の物体特徴量と共に用いられることがある [19, 20]. 検出時間の短縮のためには, カスケード構造や coarse-to-fine 処理が用いられることがある [13, 21].

1.2.3 部分認識

画像認識で最も困難な問題はオクルージョンの問題である. オクルージョン問題を解決するために, これまで多くの研究が行われてきた. その中で二つを紹介する.

第一は, 追跡を用いる方法である. 物体追跡は今まで盛んに行われてきた研究である. 物体追跡の方法ではテンプレートマッチングによる物体追跡, 勾配法による物体追跡(Mean Shift 法), 確率分布による物体追跡(Particle Filter), 弱識別器を応用した物体追跡(Online Boosting, Online Real Boosting), 物体の形状変化への対応(Soft Decision Feature による物体追跡), 特徴点ベースの物体追跡(SURF Tracking), さらに Kalman Filter を利用した方法等, 様々な手法がある. この中で Particle Filter を用いた物体追跡は, 物体の一部が隠れても追跡ができる手法である [22].

第二は, 物体を部分的に認識 (部分認識) する方法である. 部分認識の詳細は第 3 章で説明する.

1.3 本論文の構成

本論文は, 自車の周囲を走行する自転車を, 自車に装着されたカメラの映像から, コンピュータを用いて自動検出する手法を提案している. 第 2 章では, HOG 特徴量を改良した MSC-HOG(Multiple-scale Cell HOG) 特徴量に基づく自転車検出法を提案し, 実際の交通環境映像を用いた実験によって提案法の有効性を示す. 第 3 章では, オクルージョン問題の解決を意図して, 自転車車輪と人の上半身の部分検出によって自転車を検出する方法を提案する. 第 2 章と同様に, 実際の交通環境映像を用いて実験を行い, 提案する手法の有効性を示す. 第 4 章では, 検出精度の向上のために, 自転車検出器と追跡器を両方用いて自転車の位置とスケールを推定する方法を提案する. 最後に第 4 章では, 本論文の考察と今後の課題について述べる.

第2章 MSC-HOG を用いた自転車検出

近年、多くの研究者が特徴記述子と識別器を用いた歩行者検出法の研究を行っている [8-10]. 自転車については、幾つかの論文でその検出法と追跡法について扱っているが、実環境での実験データはなく、実環境で効果的に自転車の検出と追跡ができるかどうかは不明である [6-7]. 実交通環境では様々な物体が存在し、検出する物体のオクルージョン現象などが物体検出を困難にしている. 実交通環境下で自転車事故を減らすためには、高精度かつ高速な自転車検出が要求される. そのためには有効な特徴記述子と高速な識別器が必要である. 本章では HOG 特徴量を改良した MSC-HOG (Multiple-scale Cell HOG) 特徴量と, RealAdaBoost 識別器を用いた自転車検出法を提案する.

2.1 処理の流れ

学習によって識別器を構築する場合, **positive class** の画像と **negative class** の画像が必要となる. **positive class** の画像は人が乗っている自転車の画像, **negative class** の画像はそれ以外の様々な画像を用いる.

次のような流れで学習を行う (図 2.1 参照).

- (I) **positive class** の画像の集合と **negative class** の画像の集合を入力する.
- (II) 入力した画像から MSC-HOG 特徴量を抽出する.
- (III) 抽出した MSC-HOG 特徴量のベクトルを用いて RealAdaBoost による学習を行い, 弱識別器を作成する.

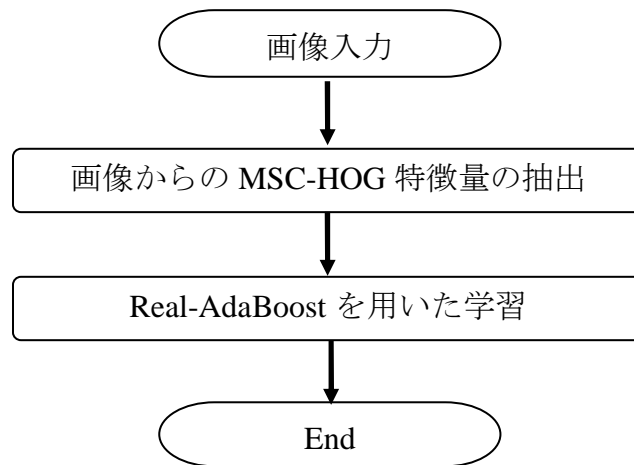


図 2.1 学習アルゴリズム

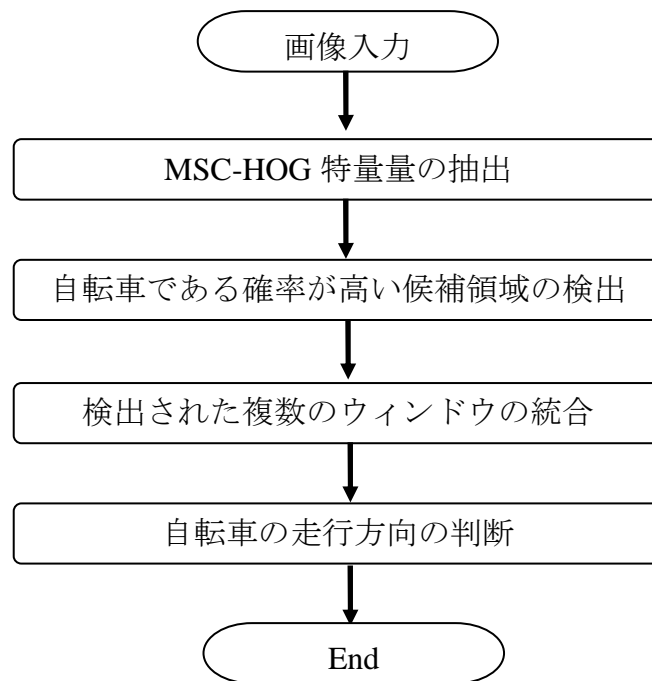


図 2.2 検出アルゴリズム

以下の流れで画像上の自転車検出を行う（図 2.2 参照）。

- (I) 検出を行う画像を入力する。
- (II) 入力画像に対し、検出ウィンドウごとに MSC-HOG 特徴量を抽出する。
- (III) 学習された **positive class** と類似度の高い画像上の領域を、自転車を含む領域として検出する。
- (IV) 複数検出されたウィンドウを統合する。

2.2 HOG(Histograms of oriented gradients)特徴量

Dalal and Triggs [12] が提案した HOG (Histograms of oriented gradients) は、輝度勾配方向ごとに画像の局所領域内の輝度勾配強度を累積してヒストグラム化する手法である。

次のような流れで輝度勾配ヒストグラムを求める。

- (I) 画像の勾配の強度と方向を算出する。
- (II) セルごとに輝度勾配ヒストグラムを生成する。
- (III) ブロックごとにヒストグラムを正規化する。

2.2.1 勾配の強度と方向の算出

画像座標を (u,v) ，輝度値を f とする場合，水平方向の一階微分を表す $f_u(u,v)$ と垂直方向の一階微分を表す $f_v(u,v)$ は式(2.1)を用いて求められる．

$$\begin{aligned} f_u(u,v) &= f(u+1,v) - f(u-1,v) \\ f_v(u,v) &= f(u,v+1) - f(u,v-1) \end{aligned} \quad (2.1)$$

式(2.1)で求められた $f_u(u,v)$ ， $f_v(u,v)$ を用いた輝度勾配画像の例を図 2.3 に示す．同図(a)は入力画像，(b)は輝度勾配画像である．

画像のエッジ特徴である勾配の強度と方向は式(2.2)から算出される．

$$\begin{aligned} m(u,v) &= \sqrt{f_u(u,v)^2 + f_v(u,v)^2} \\ \theta(u,v) &= \tan^{-1} \frac{f_u(u,v)}{f_v(u,v)} \end{aligned} \quad (2.2)$$

ここで m は勾配の強度でエッジの強さ， θ は勾配方向でエッジの角度を表す．

2.2.2 輝度勾配ヒストグラムの生成

図 2.4 に示すように，セルごとに輝度勾配ヒストグラムを生成する．例えば $0^\circ \sim 180^\circ$ の範囲を 20° ずつ分割した 9 個のビンでヒストグラムを生成すれば，特徴量ベクトル $\hat{b} = (b_1, b_2, \dots, b_9)$ が得られる．入力画像が 30×60 ピクセルで 5×5 ピクセルを 1 セルとすると，すべてのセルに対してヒストグラムを生成すれば，総計 $72 \times 9 = 648$ 個の特徴量を持つベクトル $\hat{B} = (\hat{b}_1, \hat{b}_2, \dots, \hat{b}_{72}) = (\hat{b}_1, \hat{b}_2, \dots, \hat{b}_{648})$ が得られる．

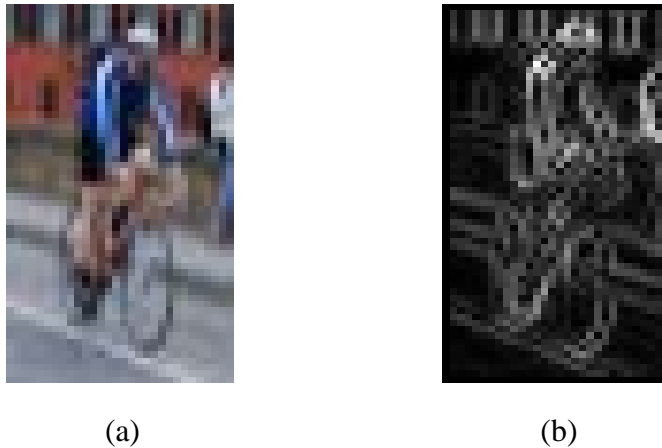


図 2.3 輝度勾配画像の導出：(a) 入力画像，(b) 輝度勾配画像

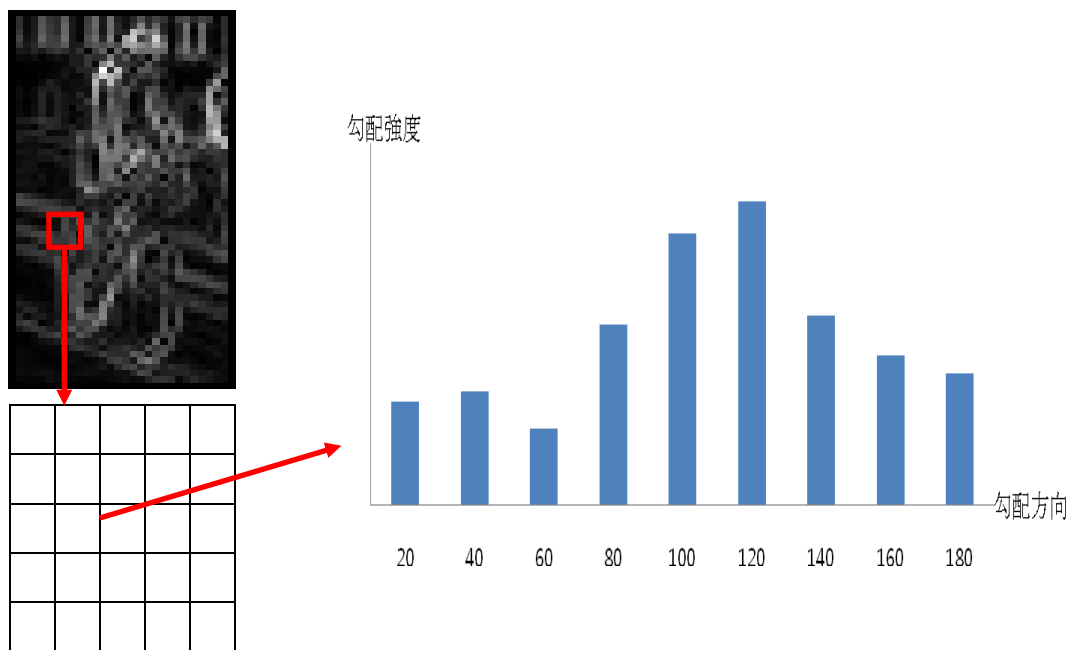


図 2.4 ヒストグラムの生成

2.2.3 ヒストグラムの正規化

2.2.2 節までに得られた特徴量ベクトルは図 2.5 のようにブロックごとに正規化される。正規化は、1 ブロックにおいて 1 セルずつ移動させながら行う。正規化されたヒストグラムは 1 つのベクトルとして表現される。

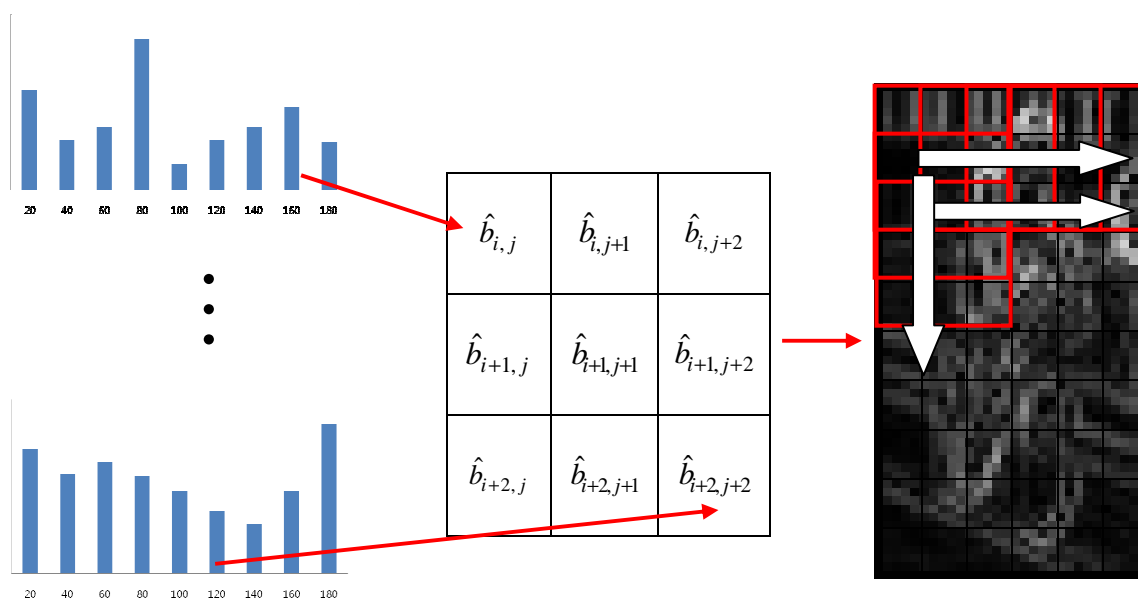


図 2.5 ヒストグラムの正規化

ブロックにより得られたベクトルを \hat{V} ，正規化後のベクトルを \hat{V}_a とすれば，正規化には次の2つの方法がある．

$$\text{L1-norm : } \hat{V}_a = \frac{\hat{V}}{\|\hat{V}\| + \varepsilon} \quad (2.3)$$

$$\text{L2-norm : } \hat{V}_a = \frac{\hat{V}}{\|\hat{V}\|^2 + \varepsilon^2} \quad (2.4)$$

ここで， $\hat{V} = (\hat{b}_{i,j}, \hat{b}_{i,j+1}, \hat{b}_{i,j+2}, \hat{b}_{i+1,j}, \hat{b}_{i+1,j+1}, \hat{b}_{i+1,j+2}, \hat{b}_{i+2,j}, \hat{b}_{i+2,j+1}, \hat{b}_{i+2,j+2})$ で表すことができる． ε は勾配の強度の値が0である場合，分母が0にならないための正の実数である． ε は非常に小さい値で，認識結果には影響を及ぼさない．

入力画像が 30×60 ，1セルが 5×5 ピクセル，ビン数が9，1ブロックが 3×3 セルの場合，1ブロックあたり81個の特徴量を持ち，総計40ブロックになるため，総計3,240個の特徴量が得られる．

HOG特徴量を用いる場合，セルとブロックのサイズ，ビン数，正規化する方法等の選択により性能に差が出るため，実験を行って最適な結果を得る必要がある．

2.3 MSC-HOG (Multiple-scale Cell HOG) 特徴量

MSC-HOG (Multiple-scale Cell HOG) 特徴量はDalalらのHOG特徴量[12]を基にしている．DalalらのHOG特徴量と本論文で提案するMSC-HOG特徴量との違いは，MSC-HOGはセルのサイズを可変にしていることである．Dalalらが提案したHOG特徴量はセルのサイズを固定して一定なエッジの長さ情報を得る．よってセルのサイズの違いによって結果が変わるため，最適な結果を得るセルのサイズを実験によって探さなければならない．セルのサイズを可変にすれば，検出する対象に対してより良いエッジ情報を得ることができる．図2.6はHOG特徴量とMSC-HOG特徴量とのセルサイズと分布の違いを表す．同図(a)は従来形[1]，(b)は提案法である．

MSC-HOGは次のように求められる．

- (I) Integral Histogram を計算する．
- (II) セルのサイズと位置を計算する．
- (III) 特徴量を算出する．
- (IV) 正規化を行う．

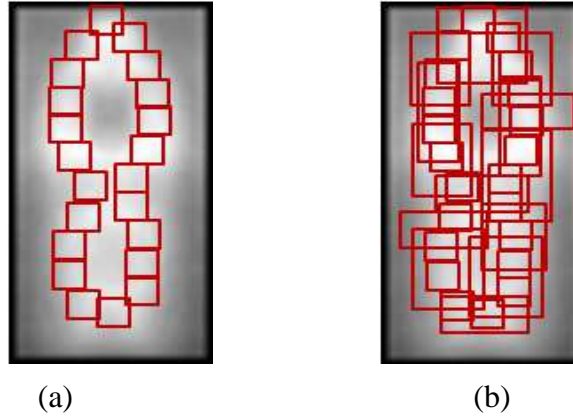


図 2.6 セルのサイズと分布の違い : (a) オリジナル HOG 特徴量,
(b) MSC-HOG 特徴量

2.3.1 Integral Histogram の計算

MSC-HOG 特徴量を用いてセルのサイズを可変にする場合, 特徴量の次元は膨大になる. 特徴量が高次元になれば, 学習と検出を行う時多くの時間が必要となる. 自転車検出を高速に行うために Integral Histogram を用いて特徴量の算出を高速化する. まず輝度勾配の強度と方向を 2.2 節の式(2.1)と(2.2)を用いて求める. そして式(2.5)を用いて各勾配方向のビン b ($b=1,2,\dots,B$)ごとに Gradient Integral Image を求める.

$$ii_b(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} i_b(x', y') \quad (2.5)$$

ここで, $ii_b(x, y)$ はビンごとの Integral Image, $i_b(x', y')$ はビン b の輝度勾配強度である.

最後に式(2.6)と式(2.7)を用いて, ビンごとに Integral Image の輝度勾配強度の累積値を求める.

$$l_b(x, y) = l_b(x, y-1) + i_b(x, y) \quad (2.6)$$

$$ii_b(x, y) = ii_b(x-1, y) + l_b(x, y) \quad (2.7)$$

ここで, $l_b(x, y)$ は同一行の輝度勾配の総和である.

Integral Image を用いれば, ある矩形領域内の画素値の総和の演算が高速にできる. 図 2.7 に示す矩形領域 D の画素値の総和 L は式(2.8)で算出され, この計算は矩形の大きさによらない.

$$L = ii(x_1, y_1) + ii(x_4, y_4) - ii(x_2, y_2) - ii(x_3, y_3) \quad (2.8)$$

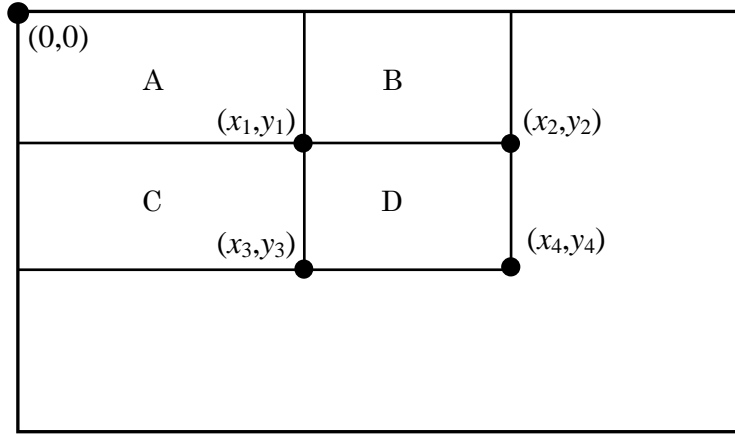


図 2.7 Integral Image の計算

2.3.2 セルのサイズと位置の計算

セルのサイズは式(2.9)を用いて計算する.

$$S_{i,x} = MIN_x + STEP \times i \quad S_{i,y} = MIN_y + STEP \times i \quad (i = 0, 1, 2, \dots, I) \quad (2.9)$$

ここで, $s_{i,x}$ は i 番目のセルの横サイズ, $s_{i,y}$ は i 番目のセルの縦サイズ, $STEP$ はセルサイズの変化幅, MIN_x, MIN_y はセルの x, y 方向の最小サイズである. 可変的なセルは画像の中にランダムに配置し, 互いに **overlap** を許容する.

2.3.3 特徴量の算出

画像の左上の座標を (x, y) とすれば, k ($k=1, 2, \dots, K$) 番目のセルは横のサイズが $S_{k,x}$, 縦のサイズが $S_{k,y}$ になる. そしてそのセルの b 番目のビンに関するヒストグラムの頻度は式(2.10a)で算出される.

$$v'_{kb} = ii_b(x, y) + ii_b(x + S_{k,x}, y + S_{k,y}) - ii_b(x + S_{k,x}, y) - ii_b(x, y + S_{k,y}) \quad (2.10a)$$

画像内のセル数は K であるから, その画像を表す特徴ベクトルは式(2.10b)で与えられる.

$$\mathbf{v}' = (v'_{1,1}, v'_{1,2}, \dots, v'_{1,B}, \dots, v'_{k,1}, v'_{k,2}, \dots, v'_{k,B}, \dots, v'_{K,1}, v'_{K,2}, \dots, v'_{K,B}) \quad (2.10b)$$

ここで, 全ての K 個のセルでビン数 B は等しいとする.

2.3.4 正規化

正規化は式(2.11a)を用いて行われる.

$$\mathbf{v} = \frac{\mathbf{v}'}{\|\mathbf{v}'\| + \varepsilon} \quad (2.11a)$$

ここで ε は分母が 0 にならないための正の実数である。 ε は非常に小さい値で、認識結果には影響を及ぼさない。

最後に、画像の表す特徴ベクトルは式(2.11b)のように定義される。

$$\mathbf{v} = (v_{1,1}, v_{1,2}, \dots, v_{1,B}, \dots, v_{k,1}, v_{k,2}, \dots, v_{k,B}, \dots, v_{K,1}, v_{K,2}, \dots, v_{K,B}) \quad (2.11b)$$

2.4. HSV 変換

識別の精度を上げるために MSC-HOG 特徴量だけではなく、テクスチャ特徴として HSV 表色系の Hue を用いる。自転車検出を行う時、自転車だけでなく自転車に乗っている人の領域も検出する。Hue は明度(Value)より照明の変化に頑強で人の肌色の特徴を持っている。RGB 表色系の R(Red), G(Green), B(Blue)から HSV 表色系 Hue(色相), Saturation(彩度), Value(明度)への変換は次式で与えられる。

$$\begin{aligned} MAX &= \max(R, G, B) \\ MIN &= \min(R, G, B) \\ Value &= MAX \end{aligned} \quad (2.12)$$

ここで $V=0$ の場合は $S=0, H=0$ である。それ以外の場合は以下の式で算出する。

$$Saturation = \frac{MAX - MIN}{MAX} \quad (2.13)$$

$$Hue = \begin{cases} 60 \times \frac{G - B}{MAX - MIN} & \text{if } R = MAX \\ 60 \times \frac{B - R}{MAX - MIN} + 120 & \text{if } G = MAX \\ 60 \times \frac{R - G}{MAX - MIN} + 240 & \text{if } B = MAX \end{cases} \quad (2.14)$$

2.5 RealAdaBoost

RealAdaBoost は R. E. Schapire[23] が提案した AdaBoost を拡張したアルゴリズムで、弱識別器の出力を positive クラスと negative クラスの確率密度分布 W^+, W^-

の差によって、連続値として実数値化したアルゴリズムである。

RealAdaBoost のアルゴリズムはサンプルの重みの更新を効率的にすることが可能で、AdaBoost より少ない弱識別器で精度の高い検出ができるため、学習時間を短縮することができる。

2.5.1 前処理とサンプルの重みの初期化

N 個のサンプル (\mathbf{x}_n, y_n) ($n=1,2,\dots,N$) に対して \mathbf{x}_n は特徴量ベクトル、 y_n は $Y = \{-1,+1\}$ の 2 値のいずれかを表す。自転車画像を学習する場合、 y_n は自転車画像 (positive 画像) に対して +1、自転車でない画像 (negative 画像) に対して -1 の値を取る。

初めにサンプル重みで初期化を行う。

$$D_0(n) = \frac{1}{N} \quad (n=1,\dots,N) \quad (2.15)$$

2.5.2 確率密度関数 W_t の作成

式(2.16)、式(2.17)を用いて、positive クラスの確率密度分布 W^+ と negative クラスの確率密度分布 W^- を作成する。

$$W^+(i) = \sum_{BIN(x)=i \wedge y_n=+1} D_t(n) \quad (2.16)$$

$$W^-(i) = \sum_{BIN(x)=i \wedge y_n=-1} D_t(n) \quad (2.17)$$

ここで、 BIN は bin への変換関数、 x は特徴量ベクトル \mathbf{x} から選択された特徴量、 i は bin の ID、 $t(t=0,1,\dots,T)$ は学習回数である。

1次元のヒストグラムのビン数は、学習サンプル数や特徴量数によって、適切に設定する必要がある。作成した確率密度分布 W^+ と W^- は、クラスごとに確率密度分布の総和が 1 になるように正規化する。

2.5.3 弱識別器の評価値 Z_m の算出

評価値 Z_m は確率密度分布の類似度を表す。正規化後の確率密度分布 $(W^+(i), W^-(i))$ を用いて、式(2.18) のように二つの分布の Bhattacharyya 距離を用いて類似度を算出する。

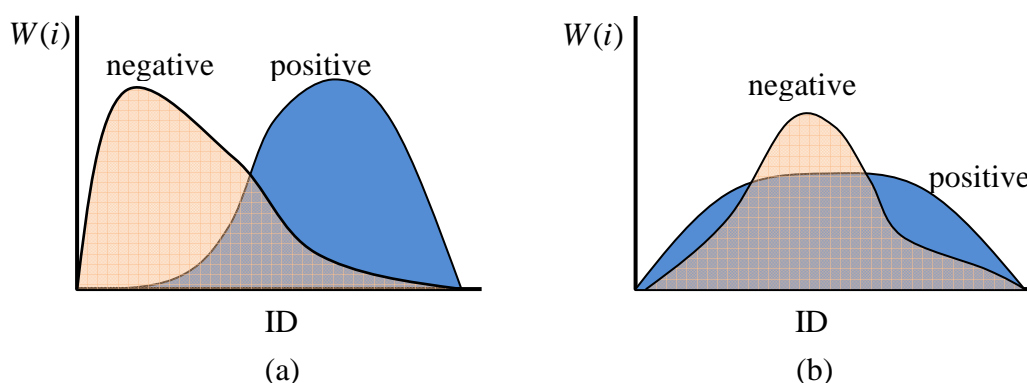


図 2.8 確率密度関数と評価値の関係:(a) Z が小さい場合, (b) Z が大きい場合.

$$Z = \sum_i \sqrt{W_+^i W_-^i} \quad (2.18)$$

ここで, Bhattacharyya 距離は統計学の分野で, 2 つの離散確率分布の間の距離を求める尺度として用いられている距離である. 画像間の類似度を評価する代表的な指標として用いられる. 図 2.8 に示すように Z が小さいほど分類は容易である.

2.5.4 弱識別器の選択

各学習ラウンド t に対して, 式(2.19)を用いて評価値 $Z_{t,m}$ が最小になる弱識別器を選択する.

$$h_t = \arg \min Z_{t,m} \quad (2.19)$$

ここで, m は弱識別器の番号である.

弱識別器の出力 $h_t(x)$ は式(2.20)から計算できる.

$$h_t(x) = \frac{1}{2} \ln \frac{W^+(BIN(x)) + \varepsilon}{W^-(BIN(x)) + \varepsilon} \quad (2.20)$$

ここで, $W^+ = W^-$ の時は $h_t(x) = 0$, $W^+ > W^-$ の時は $h_t(x) > 0$, $W^+ < W^-$ の時は $h_t(x) < 0$ である. ε は W^+ と W^- が 0 にならないための, 正の実数である.

2.5.5 学習サンプルの重み $D_t(n)$ の更新

学習サンプルの重み $D_t(n)$ の更新には式(2.21)を用いる.

$$D_{t+1}(n) = D_t(n)e^{-y_n h_t(x)} \quad (2.21)$$

学習サンプルの重み更新後は、式(2.22)を用いて学習サンプルの重みの正規化を行う。

$$D_{t+1}(n) = \frac{D_{t+1}(n)}{\sum_{n=0}^N D_{t+1}(n)} \quad (2.22)$$

学習サンプルの重みを更新する場合、 $W^+ = W^-$ ($h_t(x) = 0$)の場合は、 W^+ の positive サンプルの重みと W^- の negative サンプルの重みを変更しない。 $W^+ > W^-$ ($h_t(x) > 0$)の場合は、 W^+ の positive サンプルの重みを減らして W^- の negative サンプルの重みを増やす。 $W^+ < W^-$ ($h_t(x) < 0$)の場合は、 W^+ の positive サンプルの重みを増やし W^- の negative サンプルの重みを減らす。

2.5.6 強識別器の構築

最終の強識別器は、式(2.23)のように、学習を T 回繰り返す間に求められた弱識別器 h_t の線形和として求められる。

$$H(x) = \text{sign}\left(\sum_{t=1}^T h_t(x)\right) \quad (2.23)$$

2.6 カスケード構造

識別を行う場合、弱識別器を多く用いると分類精度は上がるが処理時間が増大する。逆に弱識別器を減らせば分類精度は低下する。そこで、分類を行う場合、分類が簡単な画像は弱識別器数を少なく、難しい画像は弱識別器数を増やすカスケード構造を用いて分類のための処理時間を短縮する。

図 2.9 に示すように弱識別器出力の累計が閾値より低い場合、以降の処理を省略して Negative クラスとして処理する。同図の WC_i は i 番目の弱識別器、 h_i は第 i Round までの弱識別器の累計、 th_i は i Round での閾値を表す。 th_i は式(2.24)で算出される。

$$th_i = \min\{h_{i,1}, h_{i,2}, \dots, h_{i,ID}\} \quad (ID \in N) \quad (2.24)$$

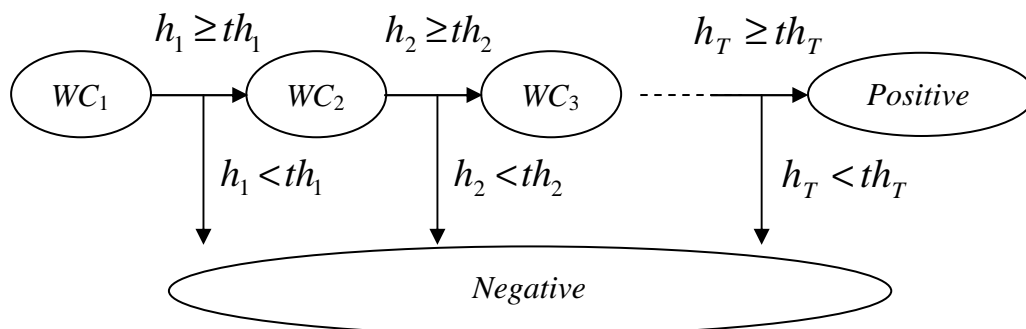


図 2.9 Cascade structure を用いた識別

ここで、 i は Round, ID はサンプル画像の ID を表す.

2.7 検出環境の構築

画像内の自転車を検出するには、画像の中のどこに自転車があるかを探索する処理と、自転車と判定された部分を統合する処理の二つの処理が必要である。画像内のどこに自転車があるかを探索する処理は、検出ウィンドウをラスタ走査しながら HOG 特徴量を用いて行う。また自転車の検出時には、一般に複数のウィンドウが検出される。それを統合する方法についても述べる。

2.7.1 ラスタ走査による探索

自転車がどこにあるかを検出するため、ラスタ走査による探索を行う。ラスタ走査をする時は、検出ウィンドウを用いて、学習した positive クラスと類似の特徴を持つ画像上の領域を自転車と判断して検出を行う。ラスタ走査は、図 2.10 に示すように、検出を行う画像の左上から右下まで横方向に行う。

検出を行う画像内にある自転車の大きさは画像によって異なるため、初めは小さい検出ウィンドウで探索を行い、ウィンドウを漸次拡大しながらラスタ走査の探索を繰り返す。これで、大きさの異なる複数の自転車や、画像によって大きさの異なる自転車を検出することができる。

2.7.2 Mean Shift によるクラスタリング

ラスタ走査探索で自転車検出を行う時は、自転車として複数のウィンドウが検出される。検出を行う画像の中に複数の自転車がある場合は、複数の自転車の周囲に検出ウィンドウが集中する。画像の中にある自転車の数だけ自転車として検出するために、自転車のまわりに自転車として検出された複数のウィンドウから中心点を求め、単一のウィンドウで統合する処理が必要である。



図 2.10 ラスタ走査探索

複数のウィンドウを統合するため、Mean Shift クラスタリングを用いる。次の方法で Mean Shift クラスタリングを行なう。

- (I) カーネル関数を用いて密度を推定する。
- (II) 密度が高い位置に mean を移動させる。
- (III) 密度が最大の位置に mean が移動するまで繰り返す。

密度の推定および密度が高い位置に移動を行う処理には、式(2.25)の Mean Shift Vector を用いる。

$$m(x) = \frac{\sum_{i=1}^n x_i K\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n K\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x \quad (2.25)$$

式(2.25)の右辺第一項は密度の推定を行っている。また右辺第二項は密度が高い位置に中心点を移動するという働きがある。カーネル関数 K は式(2.26)で定義される。

$$K(x) = \begin{cases} c(1-\|x\|) & \|x\| < 1 \\ 0 & \|x\| \geq 1 \end{cases} \quad (2.26)$$

式(2.25)の Mean Shift Vector が 0 になるまで上記の処理を繰り返せば、密度が最大の位置に中心点を移動させることができる。

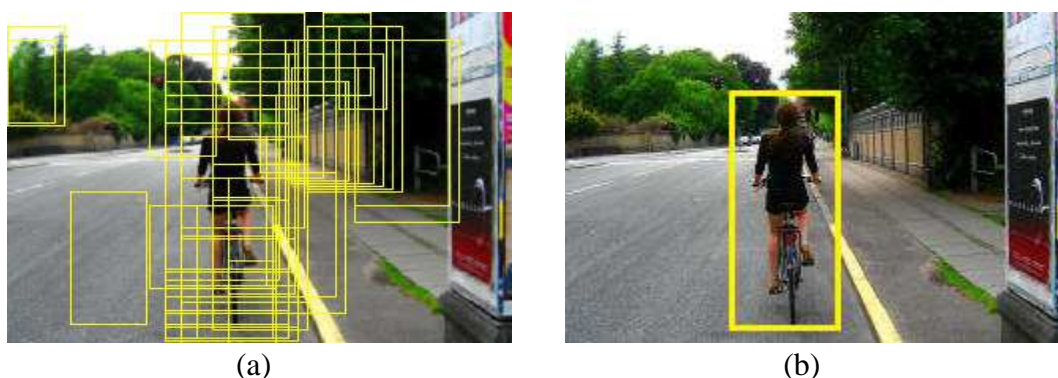


図 2.11 検出ウィンドウの統合 : (a) 統合前, (b) 統合後

2.7.3 最近傍決定則

2.7.2 節でクラスタリングされた検出ウィンドウの中心座標を一つの点に統合する時は, 式(2.27)のユークリッド距離を用いる.

$$d = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (2.27)$$

各サンプル点のユークリッド距離を求め, ユークリッド距離が閾値より小さい場合は統合処理を行う. 統合処理を行うと図 2.11 のような結果が得られる.

2.7 実験と評価

提案手法の有効性を検証するために, 実交通環境における自転車検出実験と自転車の走行方向認識実験について述べる.

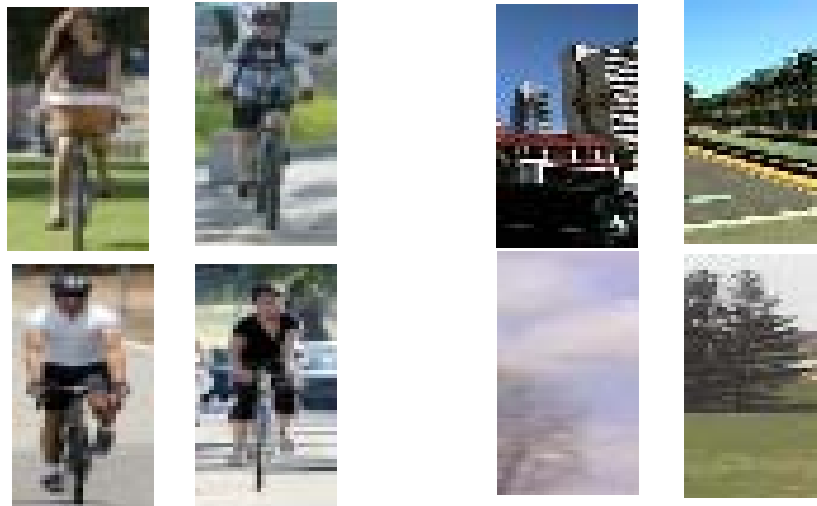
2.7.1 実験環境

学習に使う画像の例を図 2.12 に示す. インターネット上の自転車画像を収集して正規化した画像をデータベースとする. 自転車およびその走行方向を検出するために, 正面向き(1,200 枚)と右向き(1,200 枚), 左向き(1,200 枚) の **positive** 画像を収集し, さらに **negative** 画像(6000 枚) については **INRIA** のデータを用いた. 検出実験に使う画像は, 図 2.13 に示すように, 車に設置したカメラで車を走行させながら実交通環境での映像を撮った. 表 6.1 は学習と検出を行う時に使用した PC の性能, およびカメラの情報である.



(a)

(b)



(c)

(d)

図 2.12 学習データ : (a) 左向き画像, (b) 右向き画像, (c) 正面向き画像, (d) negative 画像



図 2.13 自動車に搭載されたカメラ

表 2.1 実験に用いた PC とカメラの仕様

OS	Microsoft Window 7 Home Premium 64bit
CPU	Intel®Core™ i5 2.53GHz
Memory	4.0GB
Camera	SONY HDR-HC7
Software Tool	Microsoft Visual 2008 C++

2.7.2 実験結果

1) 自転車の分類実験(Exp_1s)

図 2.12 の総数 3,600 枚の Positive 学習データの中で 2,700 枚(正面 900 枚, 右 900 枚, 左 900 枚)を学習に用い, 他の 1,200 枚(正面 300 枚, 右 300 枚, 左 300 枚)は分類実験に用いた. Negative 学習データは INRIA のデータを 6,000 枚用いた. またインターネットから取った 1,000 枚の Negative 画像を分類実験に用いた.

実験に用いたパラメータを表 2 に示す. 提案法を Dalal&Triggs[12], Watanabe ら[24]の CoHOG を用いた手法と比較した. それぞれの特徴次元数を表 2.3 に示す. 表 2.4 は, 総数 2,200 枚の分類データを用いた実験結果の認識率, また処理時間を表す.

表 2.2 実験(Exp_1s)に用いたパラメータ

Minimum cell size of width and height	5
Maximum cell size of width and height	23
Size change of cell	3
The number of orientation's bin	9
Total dimensions of MSC-HOG	160290

表 2.3 実験(Exp_1s)の特徴次元数の比較

Original HOG [1]	CoHOG [25]	Proposed method(MSC-HOG)
3240	34560	160290

表 2.4 実験(Exp_1s)の認識率と処理時間

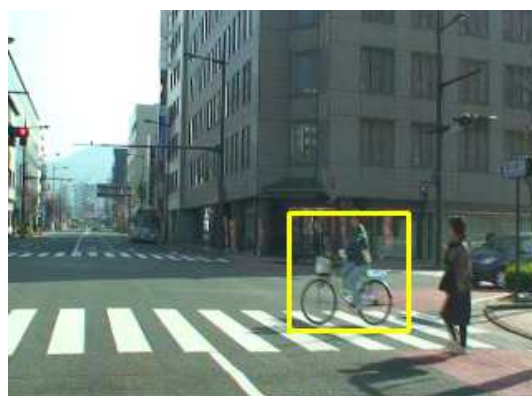
	Rate[%]	Time[ms]
Original HOG [1] + RealAdaBoost	96.5	412
CoHOG [25] + RealAdaBoost	96.8	238
Proposed method	97.1	136

2) 実交通環境における自転車検出実験 (Exp_1)

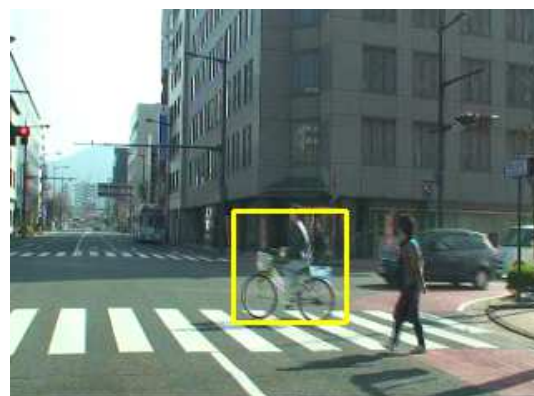
実交通環境での自転車の検出実験では、カメラを装着した車からその周囲を走行している自転車を撮影した映像の中で、自転車がどこにあるかを検出し、その位置を色付きのボックスで示す。学習には総 3,600 枚(正面 1,200 枚, 右 1,200 枚, 左 1,200 枚)の positive 画像を使い, negative 画像(6000 枚)は INRIA のデータを用いた。検出を行った結果を図 2.14, 図 2.15 に示す。図 2.14, 図 2.15 は各々 100frame で右から左へ横断歩道を自転車が走行する映像である。図 2.14 はオクルージョンがない場合(Exp_1v1), 図 2.15 はオクルージョンがある場合(Exp_1v2)である。



Frame 15



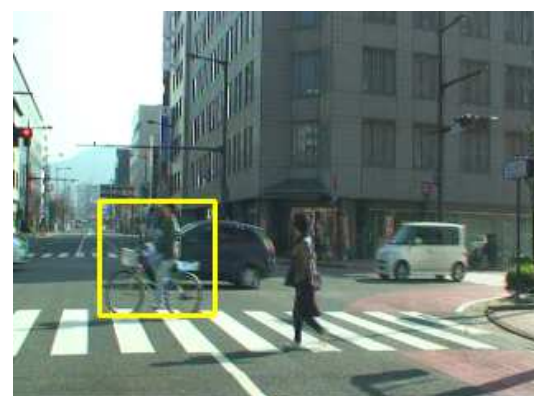
Frame 30



Frame 45



Frame 60



Frame 75



Frame 90

図 2.14 自転車検出実験(Exp_1v1) – オクルージョンがない場合



Frame 10



Frame 24



Frame 38



Frame 42



Frame 56



Frame 80

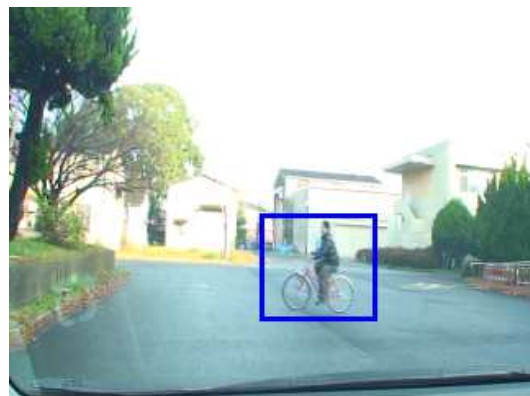
図 2.15 自転車検出実験(Exp_1v2) - オクルージョンがある場合

3) 自転車の走行方向認識実験 (Exp_2)

自転車の走行方向の検出実験は，正面，右方向，左方向に走行している自転車を検出し，その方向によって異なる色のボックスで表示する．学習には総 3,600 枚(正面 1,200 枚，右 1,200 枚，左 1,200 枚)の positive 画像を使い，negative 画像 (6000 枚)は INRIA のデータを用いた．3 方向を検出するために 3 つの方向別検出器を用いた．図 2.16 (Exp_2L) は左，図 2.17 (Exp_2R) は右，図 2.18 (Exp_2F) は正面方向に走行している自転車を検出した結果を示す．



Frame 21



Frame 40



Frame 60



Frame 80



Frame 100



Frame 120

図 2.16 自転車走行方向検出実験 (Exp_2L) – 左方向



Frame 20



Frame 39



Frame 60



Frame 80

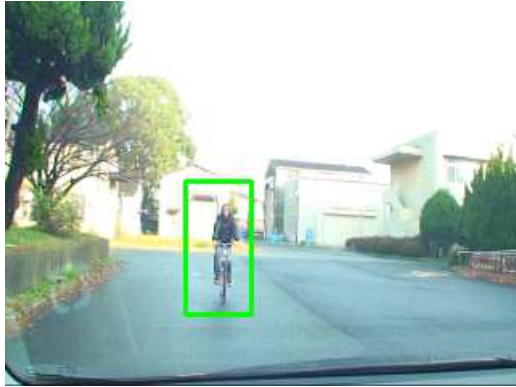


Frame 100



Frame 120

図 2.17 自転車走行方向検出実験 (Exp_2R) – 右方向



Frame 24



Frame 40



Frame 60



Frame 80



Frame 100



Frame 120

図 2.18 自転車走行方向検出実験 (Exp_2F) – 正面

2.7.3 実験結果の評価

実環境で行った自転車検出の実験と、左、右、正面方向に走行している自転車検出の実験の検出精度の評価について述べる。

検出成功は式(2.28)の式を用いて判断する。

$$\delta = \frac{R_{gt} \cap R_d}{R_{gt} \cup R_d} \quad (2.28)$$

ここで、 R_{gt} は図 2.19 のように Ground True Data で実際に自転車を含んでいる矩形領域、 R_d は本研究の検出手法を用いて自転車と判断して検出された矩形領域を示す。 δ が 0.5 以上の場合、自転車は検出されたと判断する。

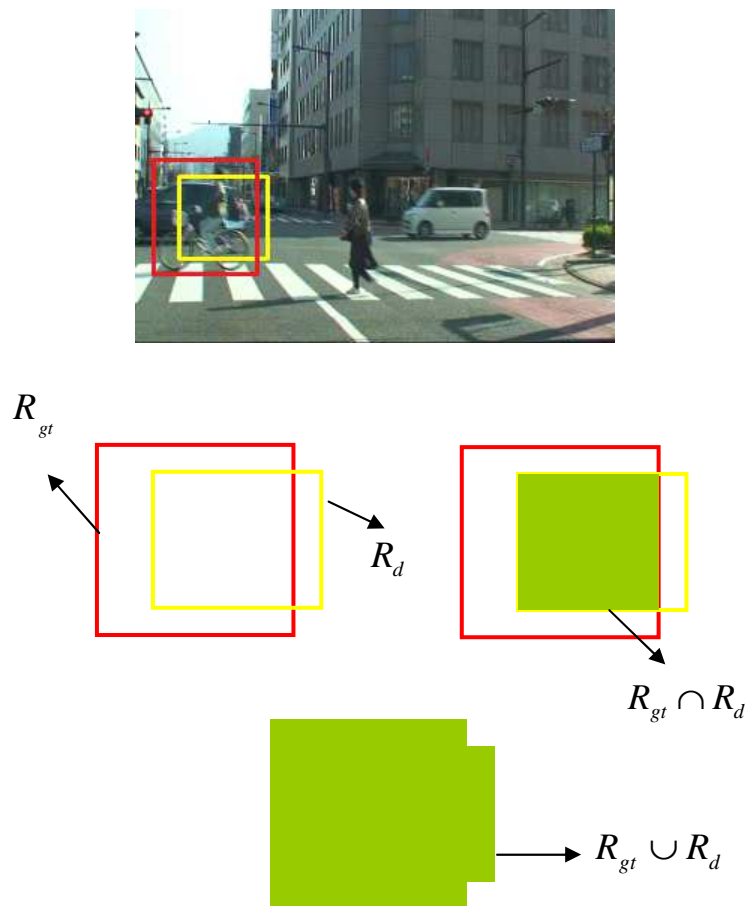


図 2.19 検出の判断

精度を評価するには、適合率(precision)と再現率(recall)を用いる。適合率と再現率はそれぞれ式(2.29)および式(2.30)を用いて求めることができる。

$$precision = \frac{TP}{TP + FP} \times 100 \quad [\%] \quad (2.29)$$

$$recall = \frac{TP}{TP + TN} \times 100 \quad [\%] \quad (2.30)$$

ここで、 TP は True Positive で、画像内の自転車が検出された検出ウィンドウの数の総和、 FP は False Positive で、画像内の自転車ではない領域を検出したウィンドウの数の総和を表す。また TN は True Negative で、画像内の自転車で検出ウィンドウが検出しなかった自転車の総数を示す。

式(2.29)の precision(適合率)は検出を行なったウィンドウの中でどのくらい自転車と検出されたか、式(2.30)の recall(再現率)は全画像の中にある自転車のうちどのくらいの自転車が検出されたかを表す。

表 2.5 は実際の交通環境での検出実験、表 2.6 は自転車の走行方向の検出実験に対する適合率と再現率を表している。表中の FPR は False Positive Rate で誤検出率を表す。 FPR は式(2.31)を用いて算出される。

$$100 - precision = \frac{FP}{TP + FP} \times 100 \quad [\%] \quad (2.31)$$

表 2.5 実交通環境における自転車検出結果の比較

	precision[%]		recall [%]		FPR [%]	
	Exp_1v1	Exp_1v2	Exp_1v1	Exp_1v2	Exp_1v1	Exp_1v2
Original HOG	96.0	84.3	91.0	79.0	4.0	15.7
CoHOG	96.0	86.5	91.0	83.0	4.0	13.5
MSC-HOG	96.0	89.0	91.0	85.0	4.0	11.0

表 2.6 自転車の走行方向の検出実験の評価

	precision[%]	recall [%]	FPR [%]
Left	92.0	83.0	8.0
Right	95.0	85.0	5.0
Front	95.0	81.0	5.0

2.8 考察

提案する手法の有効性をテストするために三つの実験を行った。第一は、positive画像とnegative画像が正しく分類できるかどうかをテストした。この実験で、DalalらのHOG特徴量およびCoHOG特徴量と比較した結果は、表2.4に示したように精度と速度面で最も良い結果が得られた。精度面ではMSC-HOG特徴量とHueの色情報を用いることが認識に最も有効な特徴量と考えられる。また速度については、カスケード構造を用いた結果、negative画像を高速に棄却することができ、それが速度の向上に寄与したと考えられる。

第二は、実交通環境における自転車検出実験である。実交通環境では様々な物体が存在し、背景が複雑で、オクルージョンが発生する場合は特に自転車検出が困難になる。実験した二つの映像の中で、Exp_1v1 (図2.14) は自転車にオクルージョンが発生しない状況、またExp_1v2 (図2.15) は自転車の一部にオクルージョンが発生する状況である。そのため、Exp_1v2 の検出率がExp_1v1 より低い。しかし提案法は、Exp_1v2 においてもオリジナルHOG特徴量やCoHOG特徴量より良い検出率が得られた。これは、オリジナルHOG特徴量やCoHOG特徴量より、提案法の方が自転車検出法としての性能がよいためと考えられる。

第三は、自転車の走行方向を検出する実験である。正面、右、左の三つの方向を検出するため、三つの検出器を用いた。正面方向の検出実験(図2.18)では、自転車が遠方にあるとスケールが小さくなるため、検出が難しくなる。この理由により、表2.6で正面方向の検出率が一番低くなったと考えられる。右と左の方向では稀に左右逆に検出されることがあった。これは自転車の形がほぼ左右対称であるためと考えられる。

提案した手法は自転車だけではなく、学習データを変えれば様々な物体検出にも適用できる手法である。また提案法はオリジナルHOG特徴量とCoHOG特徴量より精度と速度性能が良い。多くの論文では、実験的な結果としてpositive, negative画像をpositiveかnegativeかの分類だけで終らせているが、実交通環境の画像に適用するとうまく検出ができない場合が多い。それは実交通環境の画像ではオクルージョンが発生したり、検出する物体と類似の特徴を持つ物体が散在していたりと、現実的に様々な困難な問題があるためである。提案法は自転車検出、またその走行方向の判別に有効な手法であるが、提案法が実交通環境においても有効な手法であることを実験によって示した点にも意義があると考えられる。

2.9 まとめ

本章では、MSC-HOG特徴量とRealAdaBoost識別器を用いた自転車検出法、および自転車の走行方向識別法を提案した。提案法はオリジナルHOG特徴量やCoHOG特徴量より高速度でより良い精度を持つ。

今後の課題として、先進安全自動車システムへの適用を目指すために、精度や速度をさらに向上させること、また自転車以外の様々な物体の検出にも応用すること等が挙げられる。

第3章 円を用いた自転車検出

第一章で紹介したように、物体認識技術は様々な分野で適用されている。しかし、物体認識技術にはまだ問題が存在する。画像を用いて物体認識を行う場合、一番問題になるのはオクルージョンの発生である。オクルージョン問題に対して多くの解決方法が提案されている。

第一は、追跡を用いた方法である。追跡を用いた研究では、Kalman Filter, Particle Filter, Mean Shift 等を用いてオクルージョン問題に対応している[25-28]。

第二は、物体の部分を認識してそれが何の一部であるかを推測する、部分検出を使った方法である [29-31]。本章では、この部分検出法を用いた自転車検出法を提案する。即ち、自転車に乗った人の上半身と自転車車輪を部分的に検出することにより、自転車の検出を行う手法を提案する。

円／楕円型の物体は我々の周辺で多く見られる形である。それで、円／楕円検出は、昔から多くの研究者が関心を持って研究する分野であった[32-36]。円／楕円の検出は顔検出、目検出[37]、車輪検出[38]、円状の部品検出など様々な分野で適用され、今まで盛んに利用されている。

近年盛んに行われている人検出は、人の全体を検出している。一般環境下では、人は立位の状態と座位の状態が一般的である。そのため一般環境下では、上半身より下半身の形態変化が多い。上半身は形態変化が少ないが、その検出は人の部分検出として重要な課題であると考えられる。

本章の 3.1 節では円検出を用いた自転車車輪の検出法、3.2 節では上半身検出法、また 3.3 節では検出した自転車車輪と上半身部分の統合処理について述べる。

3.1 円と楕円検出

円と楕円の検出にはハフ変換がよく用いられる[39, 40]。ハフ変換を用いて円を検出する場合は直交座標上の一つの点を三次元空間(円の中心座標と半径) に写像し、最も頻度が高い位置を探してそれを逆変換することにより円を検出する。楕円の場合は五次元空間に変換されるため、円検出より計算が複雑になる。ハフ変換はノイズ、形状の歪曲(shape distortion) に頑強な長所を持つが、計算コストが高いという短所がある。また、線幅が狭いとデジタル誤差のため円が検出され難くなるという点、及び、一つの円が検出された場合、その近傍に円があればそれを検出することが難しいという短所を持つ。

本章で提案する円／楕円検出法は、ハフ変換に比べて計算コストが低く、隣

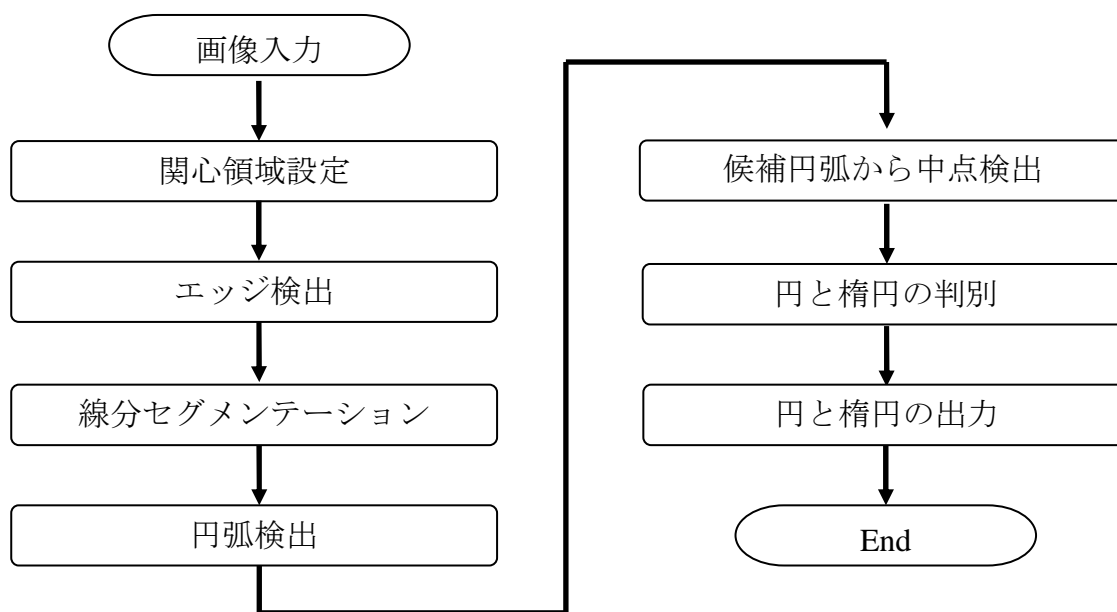


図 3.1 円／楕円検出アルゴリズム

接した円／楕円すべてが検出可能という長所を持つ。処理の流れを図 3.1 に示す。

3.1.1 関心領域の設定

画像処理を行う場合、画像全体を処理すれば多くの時間がかかる。図 3.2 (a) に示すように、車載カメラで撮影された画像を見れば、空などのように、画像の上部には明らかに自転車が存在する可能性のない領域が存在する。検出に不要な領域を除去し、関心領域だけを処理すれば処理時間の短縮が図れる。また図 3.2 (a)のように関心領域外の誤検出を減らすこともできる。

3.1.2 エッジの検出

エッジは画像内に輝度が急変する部分で、円検出のような形状認識には基礎となる大事な情報である。エッジの検出には様々な手法があるが、伝統的に 1 次微分または 2 次微分を用いた手法が一般的である。1 次微分オペレータはソーベル(Sobel)、プレウITT(Prewitt)、ロバーツ(Roberts)等、2 次微分オペレータはラプラシアン(Laplacian)、ガウシアン(Gaussian)オペレータがある。様々なエッジ検出法の中でも、1984 年に Canny が提案した edge detector は、現在でも定評のあるエッジ検出法である。これは次の手順でエッジ検出が行われる。

- (I) ガウシアンフィルタにより平滑化処理をする。
- (II) 勾配強度と勾配の方向を算出する。



図 3.2 関心領域設定の例：(a) 全体の画像，(b) 処理する関心領域

(Ⅲ) 非極大値の抑制処理を行う。

(Ⅳ) ヒステリシス閾値処理を行う。

次に、本研究で用いる Canny エッジ検出法について詳述する。

(Ⅰ) ガウシアンフィルタによる平滑化

画像に含まれるノイズを除去するために、ガウシアンフィルタを用いて平滑化を行う。2次元空間のガウシアン関数は式(3.1)で定義される。

$$G_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}} \quad (3.1)$$

ここで σ は標準偏差、ガウシアン関数の平均値は0である。 σ の値が大きいほど平滑化の効果は大きくなる。

(Ⅱ) 勾配強度と方向の算出

2章の式(2.1)及び式(2.2)を用いて勾配の強度と方向を算出する。勾配方向は4方向に量子化する(0度(0~22.5度及び157.5~180度), 45度(22.5~67.5度), 90度(67.5~112.5度), 135度(112.5~157.5度))。

(Ⅲ) 非極大値の抑制

エッジの方向に沿ってエッジを追跡し、エッジではない(最大ではない)点を削除する。画像座標を (u, v) 、その勾配強度を $m(u, v)$ とすれば、

0度の場合

$$\begin{aligned} m(u, v) > m(u-1, v) \ \&\& m(u, v) > m(u+1, v) \quad \rightarrow \quad m(u, v) = m(u, v) \\ \text{それ以外} \quad &\rightarrow \quad m(u, v) = 0, \end{aligned}$$

45度の場合

$$\begin{aligned} m(u, v) > m(u-1, v+1) \ \&\& m(u, v) > m(u+1, v-1) \quad \rightarrow \quad m(u, v) = m(u, v) \\ \text{それ以外} \quad &\rightarrow \quad m(u, v) = 0, \end{aligned}$$

90度の場合



図 3.3 エッジ検出 : (a) 関心領域, (b) 関心領域のエッジ検出

$$\begin{aligned}
 m(u,v) > m(u,v-1) \ \&\& m(u,v) > m(u,v+1) \quad \rightarrow \quad m(u,v) = m(u,v) \\
 \text{それ以外} \quad \rightarrow \quad m(u,v) = 0,
 \end{aligned}$$

135 度の場合

$$\begin{aligned}
 m(u,v) > m(u-1,v-1) \ \&\& m(u,v) > m(u+1,v+1) \quad \rightarrow \quad m(u,v) = m(u,v) \\
 \text{それ以外} \quad \rightarrow \quad m(u,v) = 0
 \end{aligned}$$

である.

(IV) ヒステリシス閾値処理

$m(u,v) < T_L$ の場合は点 (u,v) はエッジ上でない点, $m(u,v) > T_H$ の場合は点 (u,v) は強エッジ上の点, $T_L < m(u,v) < T_H$ の場合は点 (u,v) は強エッジに連結される点と見なし, エッジに追加する.

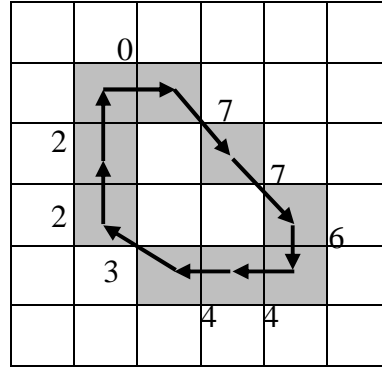
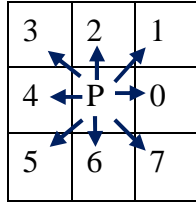
Canny エッジ検出法を用いてエッジ検出を行った結果を図 3.3 に示す.

3.1.3 線分のセグメンテーション

求められた全てのエッジが必要な情報という訳ではない. 必要なエッジ情報を得るために, 求められたエッジを追跡しながらエッジをセグメンテーションする. 追跡には図 3.4 に示すチェーンコードを用いる. 次の流れでセグメンテーションを行う.

- i) 孤立点と内部点は無視する.
- ii) 開始(start)点は新しいセグメントを作る.
- iii) 連結点を探し, 現在のセグメントに追加する.
- iv) 端点の場合, 現在のセグメントを終了する.
- v) 分岐点と交差点の場合は現在のセグメントに追加し, 現在のセグメントを終了して新しいセグメントを作る.

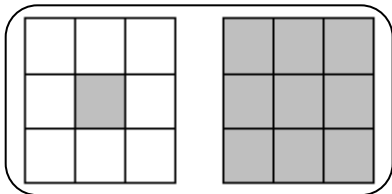
画素の連結の例を図 3.5 に示す.



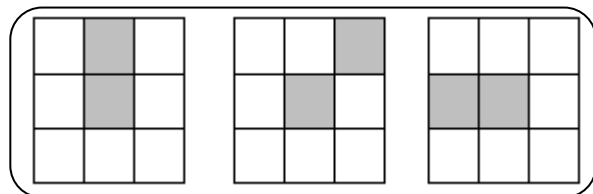
(a)

(b)

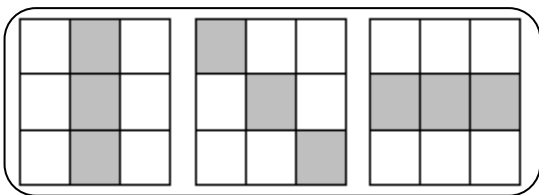
図 3.4 チェインコード : (a) 優先する追跡進行方向,
(b)チェインコードの例(077644322)



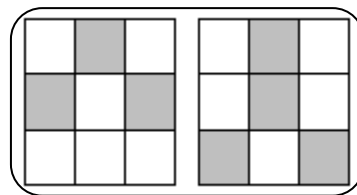
(a) 孤立点, 内部点



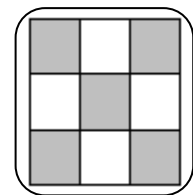
(b) 端点



(c) 連結点



(d) 分岐点



(e) 交差点

図 3.5 画素の連結の例

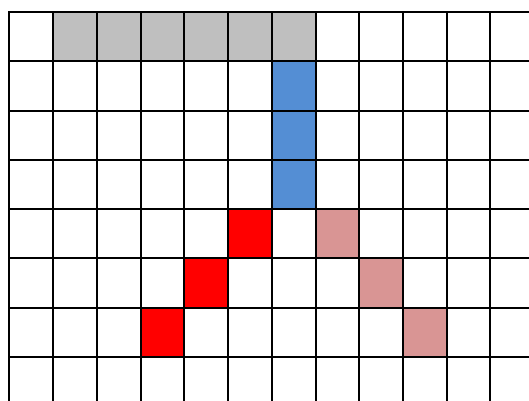


図 3.6 エッジのセグメンテーションの例

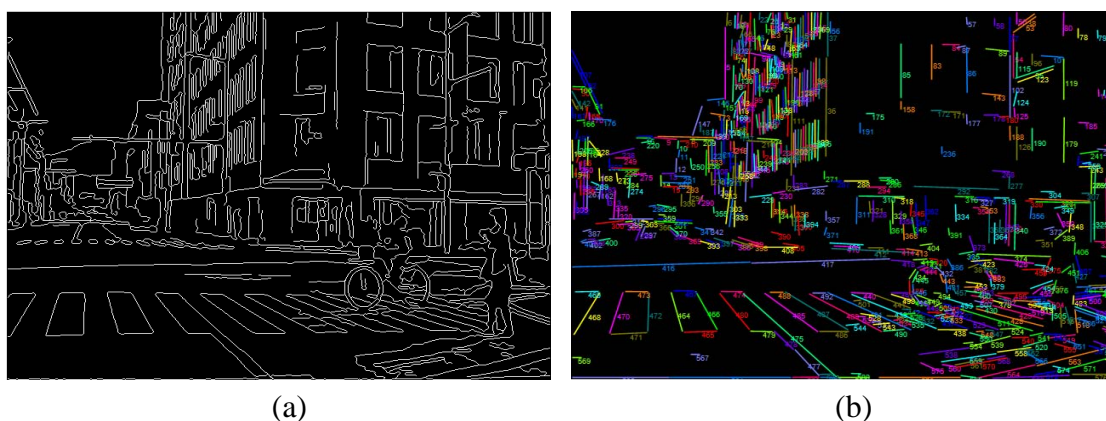


図 3.7 線分セグメンテーションの結果 : (a) エッジの画像,
(b) 線分セグメンテーションの画像

図 3.6 はエッジセグメンテーションの例である．エッジセグメンテーション後に，最小二乗法を用いてエッジに直線を当てはめる処理を行う．エッジを求めて線分セグメンテーションを行った結果を図 3.7 に示す．

3.1.4 円弧の検出

図 3.8 に示すように，連続する線分セグメントを探し，線分セグメント間の関係を式(3.2)及び式(3.3)を用いて求める．

$$\theta_i = \cos^{-1} \left(\frac{\mathbf{v}_i \cdot \mathbf{v}_{i+1}}{\|\mathbf{v}_i\| \|\mathbf{v}_{i+1}\|} \right) \quad (3.2)$$

$$\text{sign}_i = \text{sign}(x_i y_{i+1} - x_{i+1} y_i) \quad (3.3)$$

表 3.1 に示すように、式(3.2)、式(3.3)を求めれば方向が変わった点を探すが可能である。

Line	Length	Angle	Turn Direction
1	25	65	+
2	32	32	-
3	24	28	-
4	27	31	-
5	29	39	-
6	34	64	+
7	39	未知	未知

表 3.1 隣接する線分セグメントとの関係情報

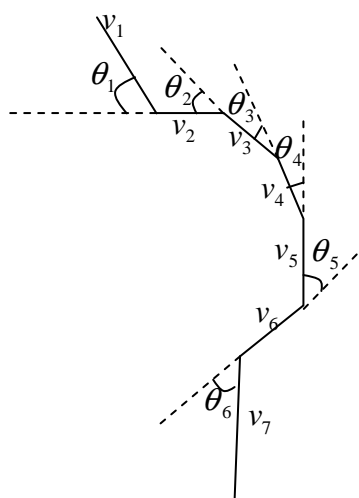


図 3.8 線分セグメントの連結例

3.1.5 円・楕円の候補円弧から中点の検出

画像内では円弧と、円弧になっている線分セグメントの情報だけが残っている。図 3.9 に示すように同じ方向の線分セグメントを探して円・楕円の中点を求める。線分セグメントの方向は、線分セグメント内の二つの点の位置情報を用いれば簡単に求めることができる。

中点の候補点として幾つもの点を選べられた場合は平均値を用いる。

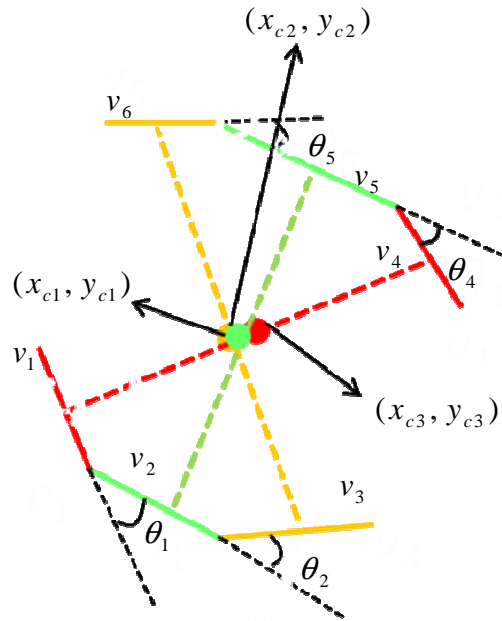


図 3.9 円と楕円の中心

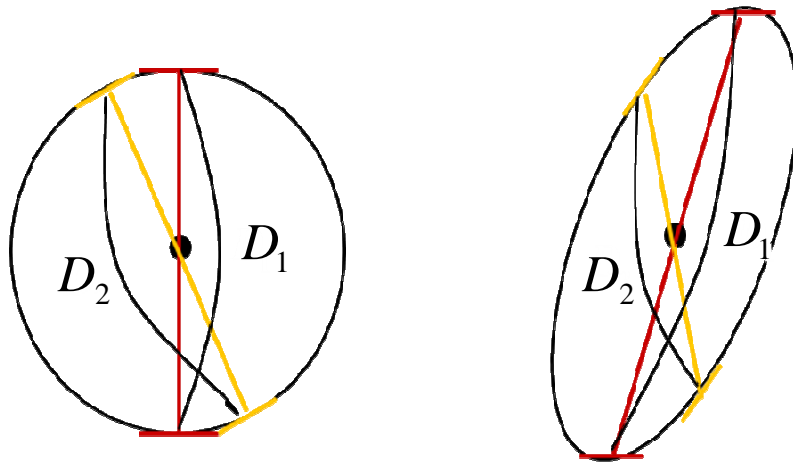


図 3.10 円と楕円の判断

3.1.6 円・楕円の判別

円・楕円の判断には次式を用いる.

$$D_m = \frac{D_1 + D_2 + \dots + D_n}{N} \quad (3.4)$$

$$\sigma^2 = \sum_1^n (D_i - D_m)^2 \quad (3.5)$$

円の場合は $\sigma \geq th$, 楕円の場合は $\sigma < th$ である. ここで, σ は標準偏差, th は閾値である.

3.1.7 円・楕円の出力

円の出力には円の方程式, また楕円の出力には楕円の方程式を用いる. 円の方程式を式(3.6)に示す.

$$x^2 + y^2 + Ax + By + C = 0 \quad (3.6)$$

円では, $A^2 + B^2 - 4C > 0$ が成り立つ. また楕円の方程式を式(3.7)に示す.

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (3.7)$$

楕円では, $b^2 - 4ac < 0$ が成り立つ.

3.1.8 円・楕円の検出結果と評価

円/楕円検出結果の例を図 3.11 と図 3.12 に示す. 図 3.11 は自転車の画像を用いた円の検出結果, 図 3.12 は実交通環境下での円検出結果である. 図 3.11 は総数 500 枚の自転車画像を用いた. 画像のサイズは, 横方向の自転車画像では 160×160 ピクセル, 斜め方向の自転車画像では 90×180 ピクセルである. 図 3.12 では, 総フレーム数 100 で画像のサイズは 320×150 ピクセルである.

円・楕円検出結果の評価を表 3.2, 表 3.3 に示す. 円の場合は式(3.8)のように, 円の中点の座標と半径を用いて評価を行う.

$$E_s = \eta \cdot (|x_{true} - x_D| + |y_{true} - y_D|) + \mu \cdot |r_{true} - r_D| \quad (3.8)$$

ここで, $(x_{true}, y_{true}, r_{true})$ は ground-truth 円の中心座標と半径, また (x_D, y_D, r_D) は検出された円の中心座標と半径である. 精度を判断するための重みは $\eta = 0.05$, $\mu = 0.1$ である.

楕円の場合は, 式(3.9) に示すように, 楕円の中点と長半径, 短半径を用いて評価を行う.

$$E_s = \eta \cdot (|x_{true} - x_D| + |y_{true} - y_D|) + \mu \cdot (|lr_{true} - lr_D| + |sr_{true} - sr_D|) \quad (3.9)$$

ここで, $(x_{true}, y_{true}, lr_{true}, sr_{true})$ は ground-truth 楕円の中心座標と長半径, 短半径, (x_D, y_D, lr_D, sr_D) は検出された円の中心座標と長半径, 短半径である. 精度判断を行うための重みは $\eta = 0.05$, $\mu = 0.05$ である.

式(3.8)と式(3.9)を用いて精度評価を行った結果を表 3.2, 表 3.3 に示す.

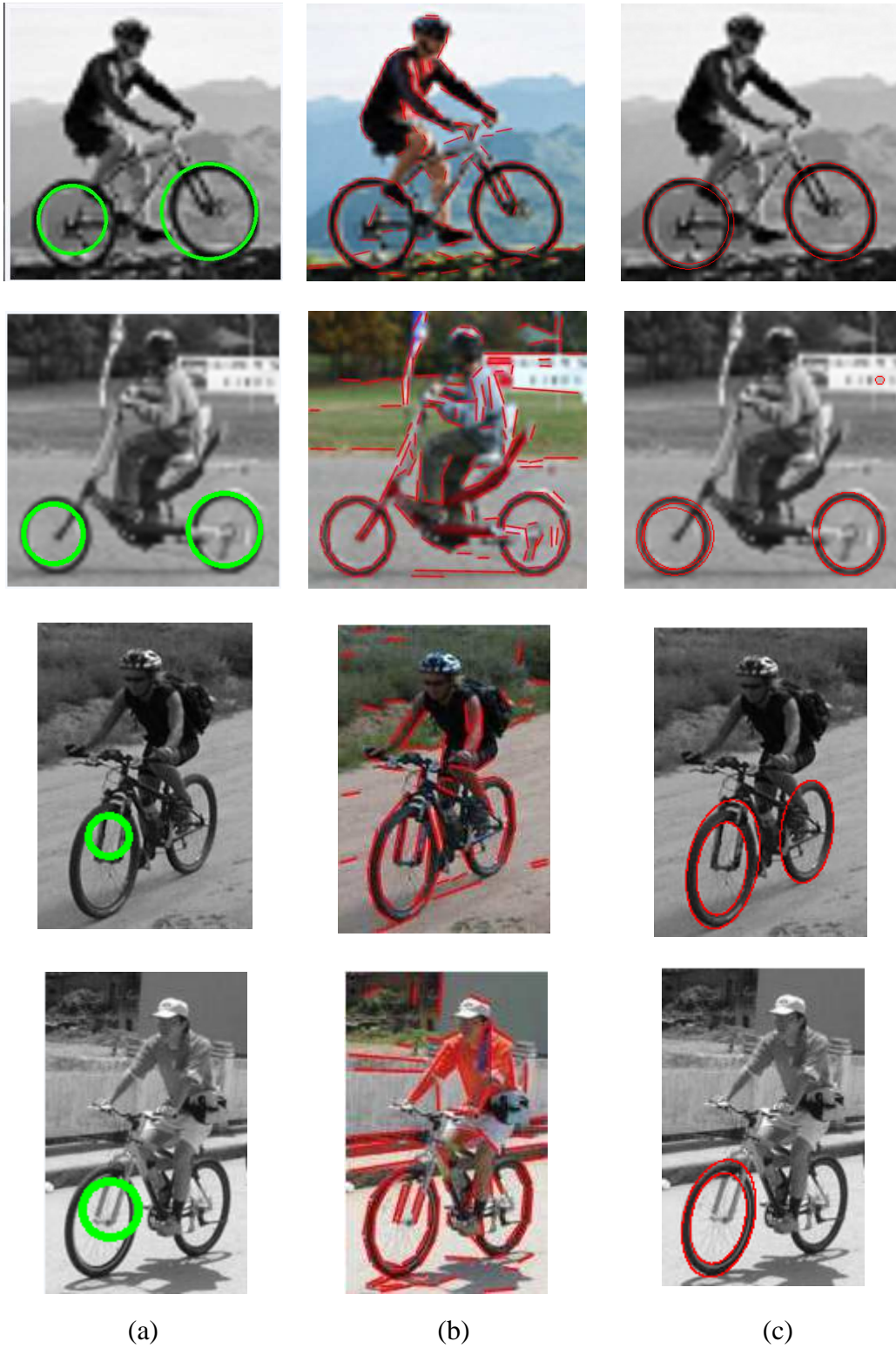


図 3.11 円・楕円の検出結果 : (a) OpenCV を用いた円検出, (b) 線分セグメンテーション, (c) 提案法を用いた円検出結果



Frame 8



Frame16



Frame 32



Frame48



Frame 64



Frame80

図 3.12 実交通環境下における円・楕円の検出結果

表 3.2 円・楕円の検出精度と処理時間の比較

	検出率 [%]	処理時間[ms/sheet]
OpenCV(CHT)	71.7	57
Proposed Method	82.9	24

表 3.3 実交通環境下における円・楕円の検出精度と処理時間

	検出率 [%]	処理時間[ms/sheet]
Proposed Method	79	40

3.1.9 円・楕円の検出結果に関する考察

表 3.2 は自転車だけが写っている画像を用いて円・楕円検出の精度評価を行った結果である。提案手法と、OpenCVにある組合せハフ変換(CHT)を用いた円・楕円検出法との比較を行った。その結果提案法は、検出率・処理時間の双方においてハフ変換より良い結果を得た。斜めの自転車画像の場合は自転車車輪が楕円型であるが、その場合、ハフ変換を用いても有効に検出できなかった。

表 3.3 は実際の交通環境下での円・楕円検出結果で、ハフ変換を使った場合は検出できなかったため、比較は行わなかった。提案法は図 3.12 のように複雑な背景下でも円・楕円の検出が可能である。本手法を用いても円・楕円検出が適切に行えなかった場合があるが、その原因は適切なエッジ情報が得られなかったことによる。よって本手法は、エッジ情報の獲得精度が上がる程、円・楕円の検出率も向上することが期待できる。

3.2 上半身の検出

自転車に乗っている人の上半身の検出に用いる特徴量は、2.3 節で説明した MSC-HOG 特徴量と 2.4 節で述べた HSV 表色系の Hue を用いる。識別には、2.5 節で述べた RealAdaBoost を用いる。

学習の方法と検出の手法は 2 章の自転車検出法で詳しい説明があるため、ここでは省略する。

3.2.1 上半身の検出結果と評価

上半身は図 3.13 のような学習データを用いる。学習データは INRIA Person Data を 30×30 ピクセルに正規化した上半身の画像 1,200 枚と、2 章で使った自転車の positive 画像の上半身部分を 30×30 ピクセルで正規化した画像 1,200 枚、

合計2,400枚の positive 画像を用いる。また negative 画像は, 2章で使った negative 画像を 30×30 ピクセルで正規化した総数 6,000 枚の画像を用いる。

まず, 上半身だけの画像(positive images) 500 枚と上半身ではない画像(negative images) 500 枚を用いて識別実験を行う。実験結果を表 3.4 に示す。さらに実際の交通環境下における実験を行う。この実験には二つの映像を用いる。各映像は 100frame で, 2章の実験でも使った映像である。その実験結果を図 3.14, 図 3.15 及び表 3.5 に示す。検出の成功・失敗の判断には第2章の式(2.28)を用い, また評価指標には式(2.29), 式(2.30), 式(2.31)を用いる。

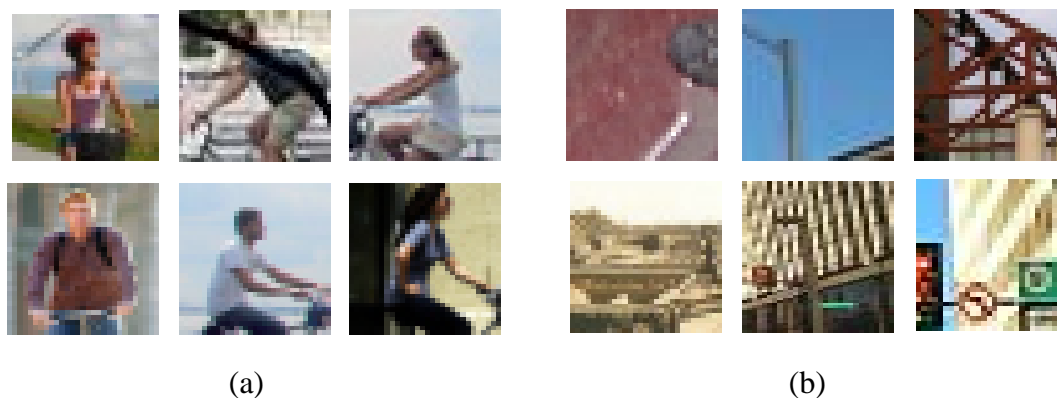
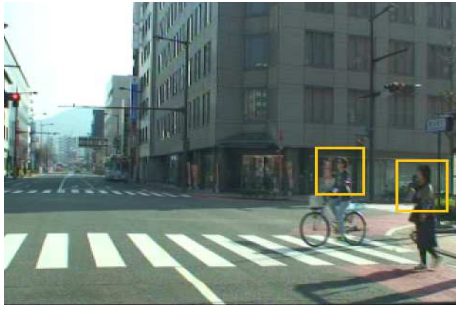
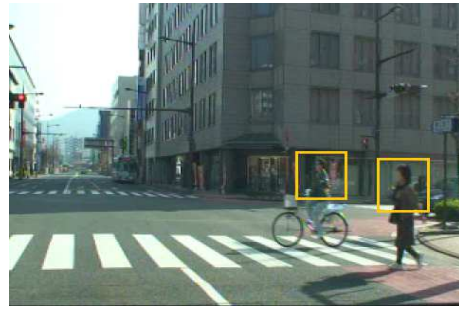


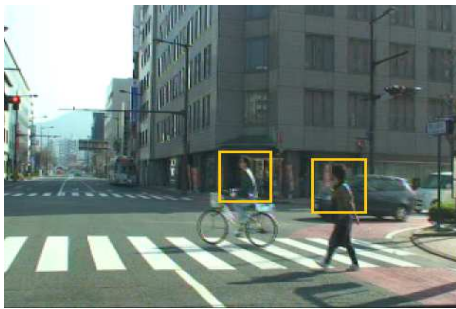
図 3.13 上半身検出に用いる学習データの例 : (a) Positive 学習データ, (b) Negative 学習データ



Frame 10



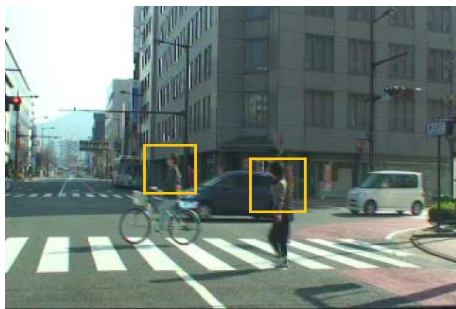
Frame 20



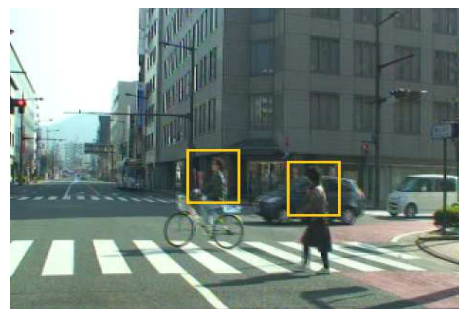
Frame 30



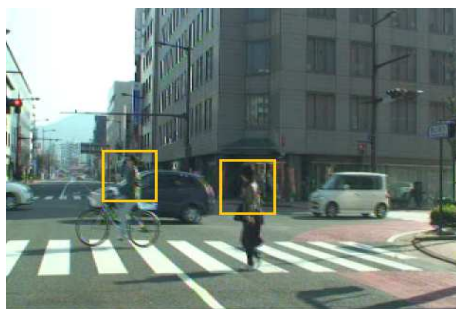
Frame 40



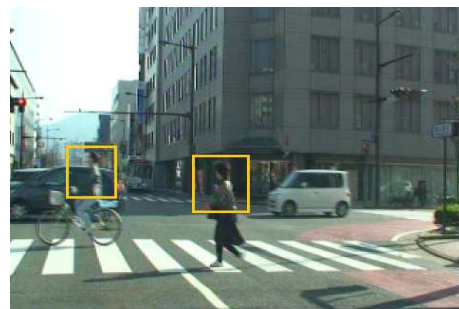
Frame 50



Frame 60



Frame 70

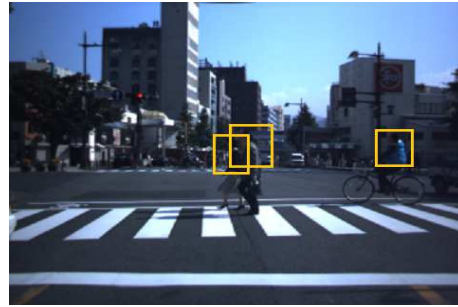


Frame 80

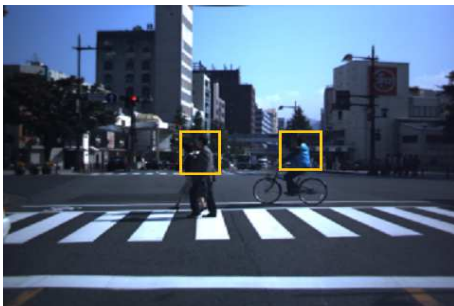
図 3.14 実交通環境下における上半身の検出 (シーン 1).



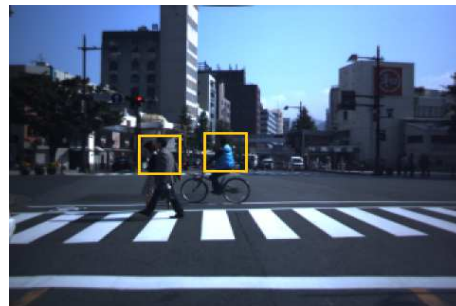
Frame 1



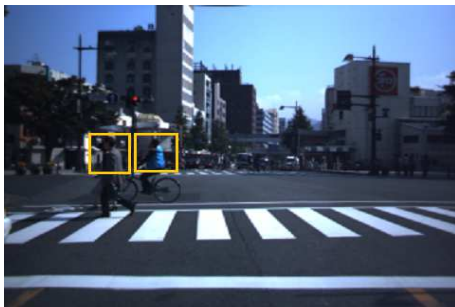
Frame 11



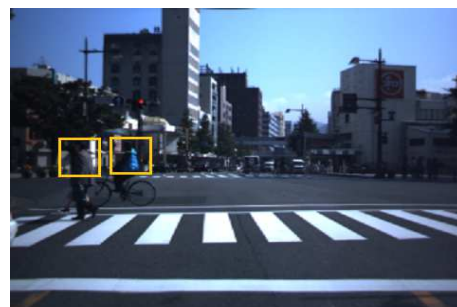
Frame 21



Frame 31



Frame 41



Frame 51



Frame 61



Frame 71

図 3.15 実交通環境下における上半身の検出 (シーン 2).

表 3.4 自転車画像を用いた上半身検出実験の結果

検出率[%]	未検出率 [%]	誤検出率 [%]
91.2	3.7	5.1

表 3.5 実交通環境下における上半身検出実験の結果

	precision[%]	recall [%]	FPR [%]
シーン 1	94.0	89.0	6.0
シーン 2	76.0	72.0	24.0

3.2.2 上半身の検出結果の考察

表 3.4 は、上半身の識別器が **positive** 画像と **negative** 画像を正しく識別するかどうかを調べる実験である。総数 1,000 枚の画像を用いた実験の検出率は 91.2 % であった。評価する画像のサイズは 30×30 ピクセルであるが、立位人物の検出の場合には一般に 30×60 ピクセルの画像を用いるので、本実験の場合は立位人物の検出より情報量が半分少ない。そのため検出率も低くなると考えられる。特に **negative** 画像が **positive** 画像と誤検出される場合が未検出より多かった。

実交通環境下における上半身検出の実験では、シーン 1 では自転車に乗っている人の上半身も歩いている人の上半身もともに良好に検出できた。未検出の原因は、照明の影響や画像の劣化によると考えられる。誤検出では、複雑な背景で人の上半身と類似する特徴を持つ領域が誤検出された。シーン 2 では、シーン 1 より検出率が 17%低い。この原因は以下の点が考えられる。シーン 2 では自転車に乗っている人の上半身はほぼ検出されたが、横断歩道を歩いている二人が平行になって、車のカメラからは一人が見えないというオクルージョンが発生する場合があります。そのため検出率が低くなったと考えられる。しかしながら、車載カメラで歩行者検出を行う場合、人が平行になって歩いている時は、一人だけ検出しても安全かどうかの判断にはあまり影響がないと考えられる。

3.3 自転車車輪と上半身の統合処理

3.1 節で述べた自転車車輪の検出器と，3.2 節で述べた上半身の検出法を用いて，人が乗って走行している自転車の全体領域を検出する手法について述べる。

二つの車輪と上半身が検出された場合，図 3.16 のように a_1, a_2, a_3, a_4 の長さを設定すれば良い。即ち， $a_1 = b_1$ 及び $a_2 = a_3 = a_4 = 1.5 \times r$ とする。一つの車輪と上半身が検出された場合， $a_1 = b_1$ ， $a_2 = 1.5 \times r$ ， $a_3 = 5.5 \times r$ とする。

上半身だけが検出された場合，立っていたり歩いている人の上半身か，自転車に乗っている人の上半身かを判断する必要がある。そのために，上半身の下側の領域を下半身検出器を用いて検出する。下半身でなければ自転車に乗っている人の上半身と判断する。車輪だけ検出された場合は，車輪領域と判断する。

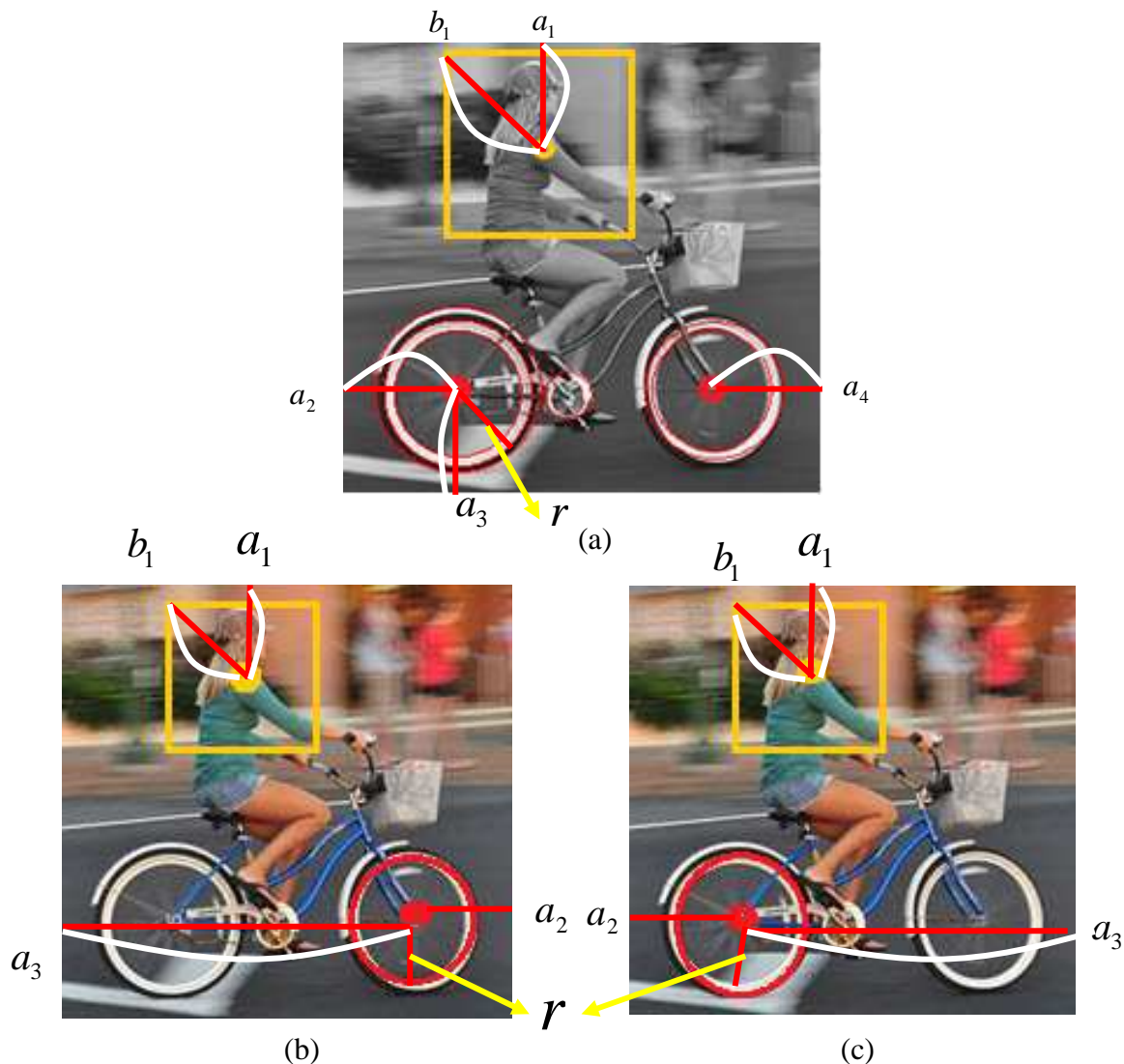
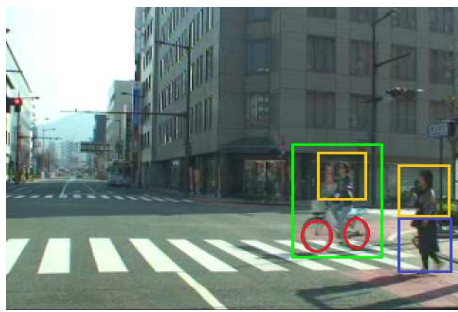
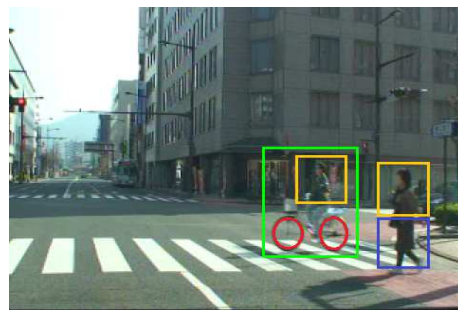


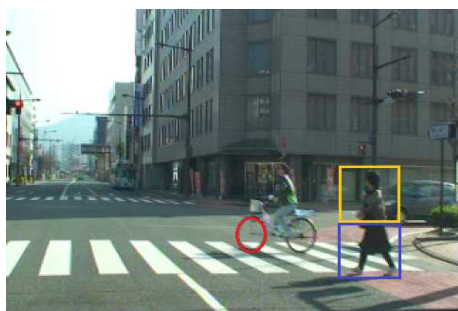
図 3.16 自転車運転者の上半身と車輪の統合：
(a) 上半身と車輪二つ，(b)と(c) 上半身と車輪一つ。



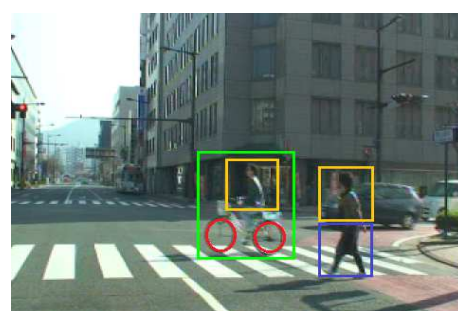
Frame 10



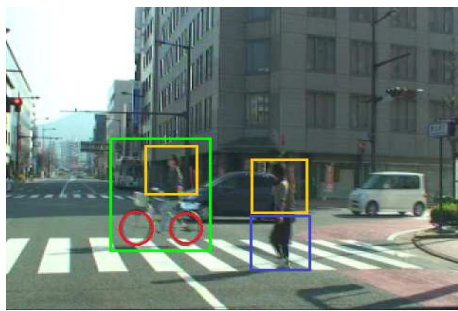
Frame 20



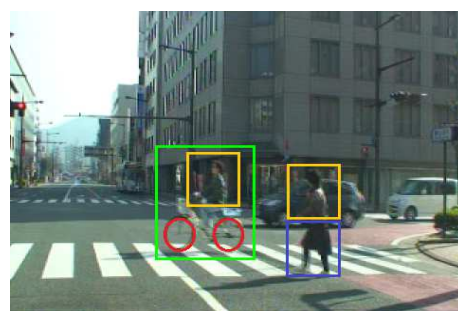
Frame 30



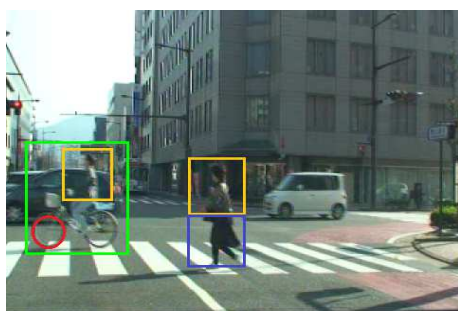
Frame 40



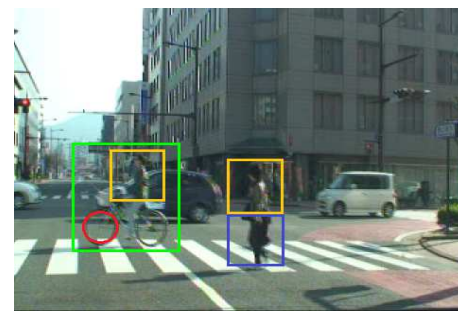
Frame 50



Frame 60

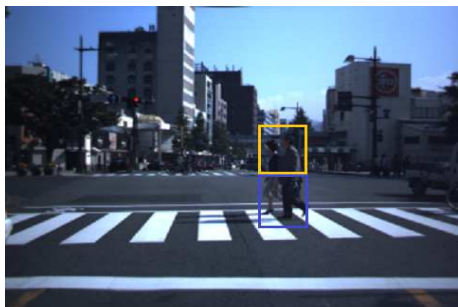


Frame 70

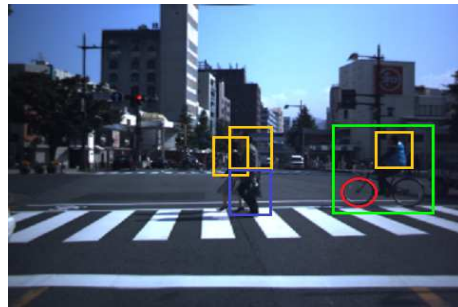


Frame 80

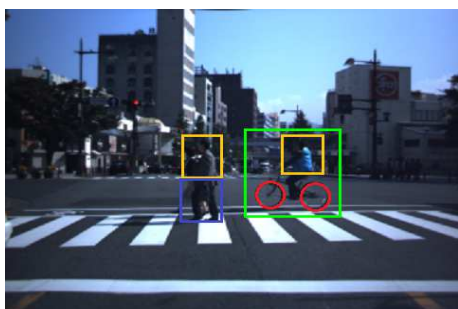
図 3.17 実交通環境下における自転車検出 (シーン 1).



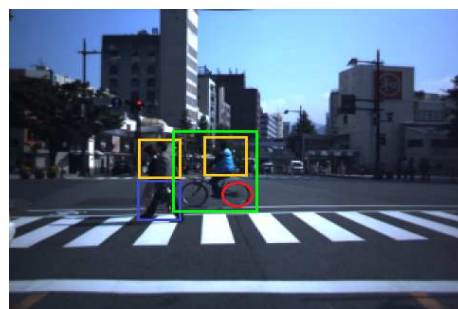
Frame 1



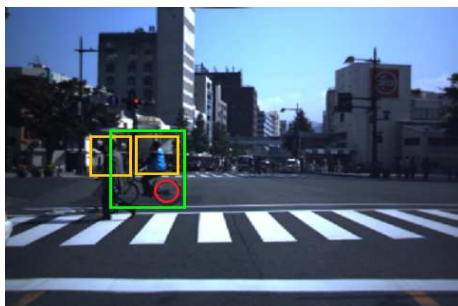
Frame 11



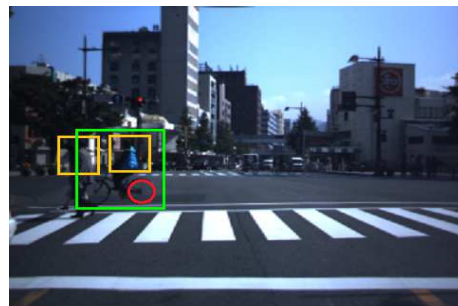
Frame 21



Frame 31



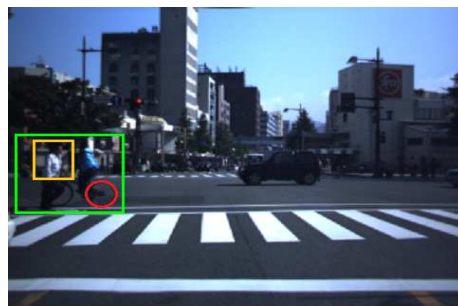
Frame 41



Frame 51



Frame 61



Frame 71

図 3.18 実交通環境下における自転車検出 (シーン 2).

3.3.1 自転車の検出と結果の評価

検出された自転車車輪領域と人の上半身領域の統合処理によって自転車検出を行う。3.1 節で円検出を行った自転車画像を用いて実験を行う。この実験は、3.1 節のように横向きと斜め向きの自転車の画像 500 枚を用いる。実験結果を表 3.6 に示す。

また実際の交通環境下における実験を行う。この実験には二つの映像を用いる。各映像は 100 frame で、3.2 節の実験で使った映像である。実験結果を図 3.17、図 3.18 及び表 3.7 に示す。検出の成功・失敗の判断には第 2 章の式(2.28)を用い、評価指標には式(2.29)、式(2.30)、式(2.31)を用いる。

3.3.2 統合処理による自転車検出結果の考察

表 3.6 は、円検出で得られた円検出率より、円検出と上半身検出を行ってそれらを統合する方法の方が精度が良いことを示している。円の一部分だけが検出されても上半身との統合処理が行われるので、精度が向上したものと考えられる。

実交通環境下における自転車検出実験では、シーン 1 (図 3.17 参照) では、円検出も上半身検出も正しく行えたので統合処理も適切に行えた。しかしシーン 2 (図 3.18 参照) では、二人の歩行者が並んで歩いているため、オクルージョンによって上半身検出が正しく行えない場合があった。このような場合、統合処理を導入すれば、上半身の下側の領域を人の下半身領域かどうか判断するため、自転車に乗っている人の上半身の判断ができる。その上半身の領域と円検出で行った車輪検出領域を統合して、表 3.7 の結果を得た。この結果は 2 章で提案した方法の実験結果より良い。特にシーン 2 の場合が 2 章で提案した方法より良い。シーン 2 は自転車の一つの車輪にオクルージョンが発生する場合が時々あるが、3 章の手法では、一つの自転車車輪しか検出できなくても自転車を検出することができるため、より良い結果が得られた。

表 3.6 自転車画像を用いた自転車検出実験の結果

検出率[%]	未検出率 [%]	誤検出率 [%]
90.0	1.2	8.8

表 3.7 実交通環境下における自転車検出の実験結果

	precision[%]	recall [%]	FPR [%]
シーン 1	94.0	91.0	6.0
シーン 2	90.0	87.0	10.0

3.4 まとめ

本章では、自転車車輪を検出するための円・楕円検出と、人の上半身検出を用いて自転車領域の部分的な検出を行った後に、統合処理により自転車検出を行う手法を提案した。

また、提案した円・楕円検出法は、実験によってその有効性が示された。本円・楕円検出法は、ハフ変換を用いた方法より高速、高精度で円・楕円の検出を行うことができる。

提案法は、オクルージョンによって自転車の車輪が一つしか見えない場合でも、自転車を運転する人の上半身検出を行って車輪と上半身を統合する処理により、効果的に自転車の検出が行えることを示した。

第4章 検出器と追跡器を用いた自転車の位置と

スケールの推定

コンピュータビジョンにおいて、物体追跡は応用性が高く、これまでに盛んに研究されている分野の一つである。物体追跡は知能型監視カメラ、知能ロボット、工場自動化システム、無人自動車、無人航空機(UAV ; Unmanned Aerial Vehicle)など幅広く応用されている。物体追跡にはテンプレートマッチング、勾配法、オンライン学習、確率分布を用いた方法等、様々な手法がある。その中でも Particle Filter を用いた確率分布による物体追跡は、複雑な背景下での不規則な動きにも対応できる効率的な物体追跡手法として知られている。本章では、HOG 特徴量を用いた物体検出器と Particle Filter を用いた物体追跡器を利用して、逐次位置情報の更新による自転車検出と追跡法を提案する。HOG 特徴量を用いた物体検出は2章で説明したので、本章では Particle Filter を中心に提案法を説明する。HOG 特徴量を用いた検出器には特徴量の算出を高速化するため Integral Histogram を用いる。また識別には RealAdaBoost を用いる。

4.1 処理のながれ

提案する手法の処理の流れを図4.1に示す。提案法では、図4.1に示すように、HOG 特徴量を用いた検出器による情報 x', y', H'_x, H'_y と Particle Filter を用いた追跡器による情報 x, y, H_x, H_y の両方を用いて、物体の位置 (x, y) とスケール (H_x, H_y) を求める。検出器による位置と追跡器による位置を両方用いるため、検出器が検出できなかった場合は追跡器による位置推定、追跡器が追跡できなかった場合は検出器による位置推定ができる。そのため、未検出と誤検出を減らすことができると考えられる。よって提案法は、本論文の目的である実交通環境への適用についても有効と考えられる。提案手法の有効性は実交通環境下における実験により確かめる。

Particle Filter を用いた追跡は、図4.1の点線の中の処理である。詳しい説明は4.2節で述べる。

4.2 Particle Filter を用いた物体追跡

Particle Filter は、近年盛んに利用されている物体追跡アルゴリズムである。複

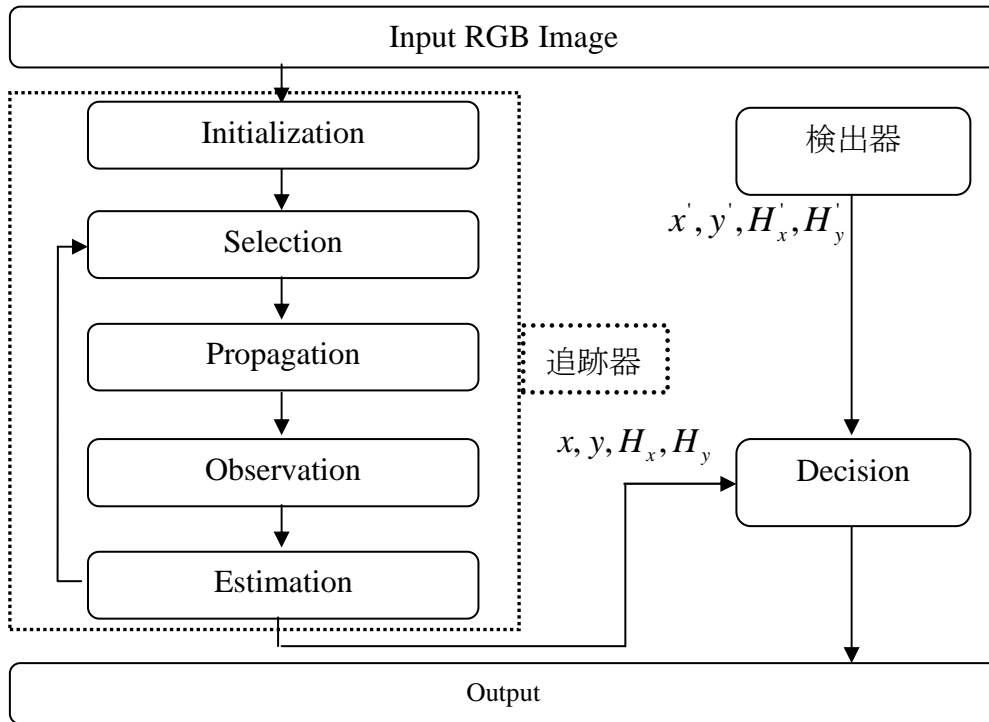


図 4.1 提案法の処理の流れ

雑な背景の一般環境ではノイズが含まれやすいため、その状況で物体を効果的に追跡するのは難しい問題である。Particle Filter は複雑な背景の中でも移動物体を効果的に追跡する方法として立証されている[41, 42].

Particle Filter のアルゴリズムは、Particle というサンプルの事前確率分布の近似により目標物の位置を推定する方法である。本章で使う Particle Filter のアルゴリズムを以下に示す。

Step1: サンプルセット $S = \{s_0^{(n)}, w_0^{(n)}\}_{n=1}^N$ を初期化する。

ここで、 $s = \{x, y, \dot{x}, \dot{y}, H_x, H_y\}$ 、 w はサンプルの重みで、サンプルセットは $1/N$ で初期化される。 x, y は最初に検出された自転車のウィンドウの中心に初期化される。 H_x, H_y は検出された自転車のウィンドウサイズである。

Step2: $S_t^{(n)}$ と $w_{t-1}^{(n)}$ のサンプルセットから N 個のサンプルを選択する。

(a) 累積分布関数 $c_{t-1}^{(n)}$ を式(4.1)を用いて計算する。

$$c_{t-1}^{(0)} = 0$$

$$c_{t-1}^{(n)} = c_{t-1}^{(n)} + w_{t-1}^{(n)} \quad (4.1)$$

$$c_{t-1}'^{(n)} = c_{t-1}^{(n)} / c_{t-1}^{(N)}$$

(b) N 個の乱数 r を生成する ($r \in [0,1]$) .

(c) バイナリサーチにより最小値 j を求める ($c_{t-1}'^{(n)} \geq r$).

(d) フィルタ粒子を $S_{t-1}'^{(n)} = S_{t-1}^{(n)}$ から求める.

$$S_t^{(n)} = AS_t'^{(n)} + G_{t-1}^{(n)} \quad (4.2)$$

$$\begin{pmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{pmatrix} + \begin{pmatrix} g_{t-1}^x \\ g_{t-1}^y \\ g_{t-1}^{\dot{x}} \\ g_{t-1}^{\dot{y}} \end{pmatrix}$$

Step3: 式(4.2)に示すように、リサンプリングのセット S_{t-1}' から各サンプルを遷移させる. ここで、 G_{t-1} は 多変量ガウスランダム変数の集合である.

Step4: カラー分布を観測する.

(a) 各サンプルセット S_t におけるカラー分布密度関数 $p_{S_t}^{(u)}$ を式(4.3)を用いて計算する.

$$p_{S_t}^{(u)} = f_1 \sum_{i=1}^I k \left(\frac{\|S_t^{(n)} - X_i\|}{a} \right) \delta[h(X_i) - u] \quad (4.3)$$

ここで、 f_1 は式(4.4)で正規化された定数、 I はサーチ領域の総ピクセル数、 k は式(4.5)で定義されるカーネル関数、 a はスケール要素、 X_i はサーチ領域のピクセル位置、 h は M -bin のカラーヒストグラム、 u は第 m 次のビンである.

正規化は次式で行われる.

$$f = 1 / \sum_{i=1}^I k \left(\left\| \frac{S_t^{(n)} - X_i^*}{a} \right\|^2 \right) \quad (4.4)$$

ここで X_i^* は 0 から画像の中心座標で正規化されたピクセル位置である。ただし、

$$k(r) = \begin{cases} 1 - \|r\|^2 & \|r\| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

(b) 式(4.6)を用いて S_t の各サンプルから Bhattacharyya 係数 ρ を計算する。

$$\rho[p_{S_t^{(n)}}, q] = \sum_{u=1}^m \sqrt{p_{S_t^{(n)}}^{(u)} q^{(u)}} \quad (4.6)$$

q は式(4.7)によるカラー分布密度関数である。

$$q^{(u)} = f_2 \sum_{i=1}^I k(\|X_i^*\|^2) \delta[b(X_i^*) - u] \quad (4.7)$$

ここで、

$$f_2 = 1 / \sum_{i=1}^I k(\|X_i^*\|)$$

である。

(c) 式(4.8)を用いて各サンプルの重み S_t を求める。

$$w_t^{(n)} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(1-\rho[p_{S_t^{(n)}}, q])}{2\sigma^2}} \quad (4.8)$$

Step5: 式(4.9)によりサンプルの位置を推定する。

$$E[S_t] = \sum_{n=1}^N w_t^{(n)} S_t^{(n)} \quad (4.9)$$

4.3 自転車の位置とスケール推定

従来研究では、検出器だけを用いて物体を検出する方法[8, 9, 24]と検出器で検出した後、追跡器を用いて追跡を行う方法[7, 43]がある。提案法は、フレームごとに、検出器から検出される情報と追跡器による追跡情報を用いて現在の自転車位置とスケールの推定を行う。

自転車のスケール情報は式(4.10)及び式(4.11)を用いて求める。

$$BI^s = (PF_t^s + Mean_DT_t^s) / 2$$

$$if \quad |PF_t^s - (Mean_DT_t^s)| < th_1 \quad (4.10)$$

$$BI^s = Mean_DT_t^s$$

$$if \quad |PF_t^s - (Mean_DT_t^s)| > th_1 \quad (4.11)$$

ここで、 $Mean_DT_t^s = (DT_{t-2}^s + DT_{t-1}^s + DT_t^s) / 3$ で、 t は現在のフレームである。 PF_t^s は追跡器を用いたスケールの情報、 DT_t^s は検出器を用いたスケールの情報である。

自転車位置のパラメータである BI^p は式(4.12)及び式(4.13)を用いて求める。

(i) $|PF_t^p - DT_t^p| < th_2$ の場合

$$BI^p = (PF_t^p + DT_t^p) / 2 \quad (4.12)$$

$$if \quad PF_t^p = 0, \text{ then } BI^p = DT_t^p$$

$$if \quad DT_t^p = 0, \text{ then } BI^p = PF_t^p$$

ここで、 PF_t^p は追跡器の位置情報、 DT_t^p は検出器の位置情報である。

(ii) $|PF_t^p - DT_t^p| > th_2$ の場合

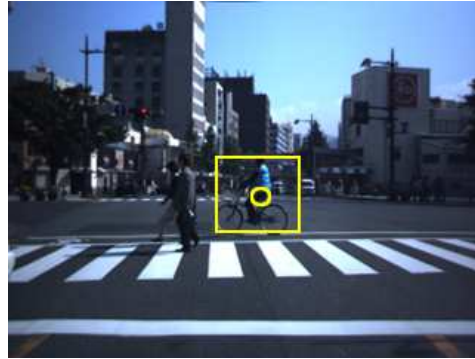
$$PF_t^p = PF_{t-1}^p + ((PF_{t-3}^p - PF_{t-2}^p) + (PF_{t-2}^p - PF_{t-1}^p)) / 2$$

$$DT_t^p = DT_{t-1}^p + ((DT_{t-3}^p - DT_{t-2}^p) + (DT_{t-2}^p - DT_{t-1}^p)) / 2 \quad (4.13)$$

提案法の全体の処理例を図 4.2 に示す。図 4.2(a)は追跡器による追跡された自転車の Particle 分布を表す。図 4.2 (b)は追跡器を用いた追跡結果、(c)は検出器を用いた検出結果を表す。図 4.2 (d)に示すように、式(4.10)~式(4.13)を用いて、追跡による自転車の位置とスケールの情報、また検出器による自転車位置とスケールの情報を算出し、現在の自転車の位置とスケールを推定する。



(a)



(b)



(c)



(d)

図 4.2 提案法の流れ: (a) Particle 分布, (b) 追跡器による追跡結果, (c) 検出器による検出結果, (d) 位置, スケールの推定結果

4.4 実験と評価

本手法の有効性を確かめるために，実交通環境下の三種類の車載カメラ映像を用いて実験を行う．実験結果を図 4.3，図 4.4，図 4.5 に示す．

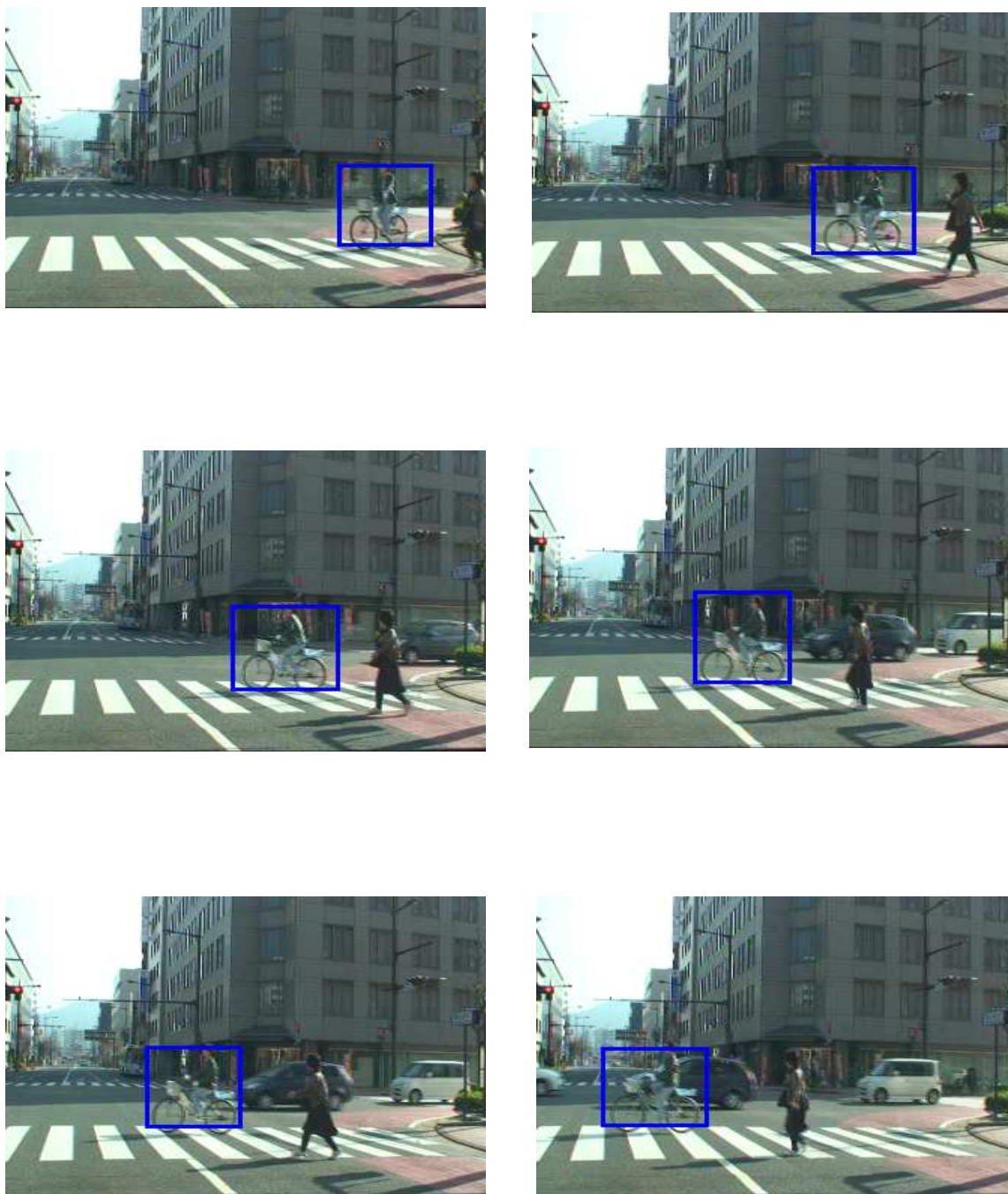


図 4.3 実交通環境下における検出器と追跡器を用いた自転車検出（シーン 1）



図 4.4 実交通環境下における検出器と追跡器を用いた自転車検出 (シーン 2)

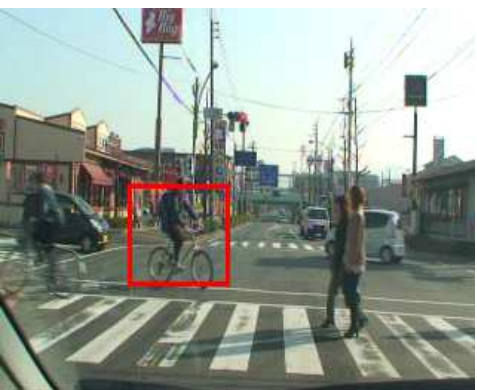
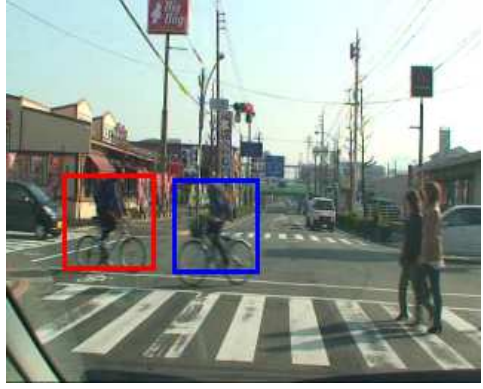
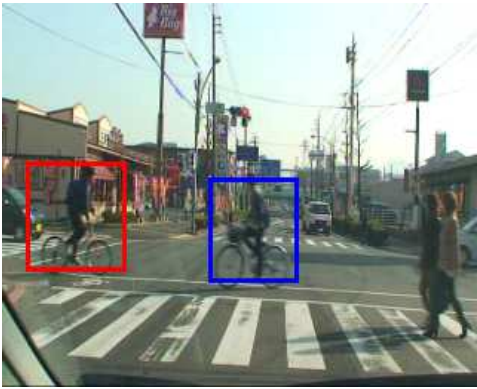
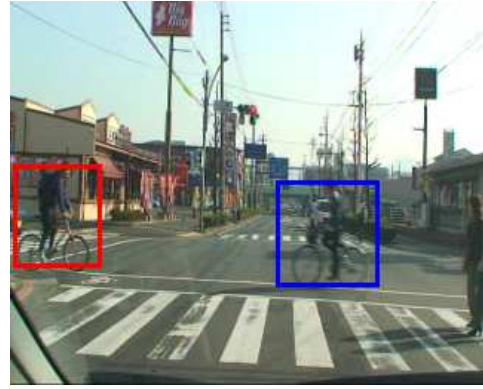
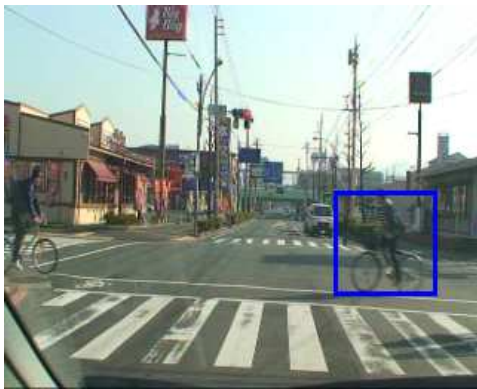


図 4.5 実交通環境下における検出器と追跡器を用いた自転車検出 (シーン 3)

表 4.1 実交通環境下における検出器と追跡器を用いた自転車検出の結果

	<i>precision</i> [%]	<i>recall</i> [%]	<i>FPR</i> [%]
シーン 1	96.0	86.2	4.0
シーン 2	87.6	81.3	12.4
シーン 3	89.0	70.6	11.0

表 4.2 実交通環境下における検出器のみを用いた自転車検出の結果

	<i>precision</i> [%]	<i>recall</i> [%]	<i>FPR</i> [%]
シーン 1	96.0	91.0	4.0
シーン 2	81.7	76.8	18.3
シーン 3	88.1	53.3	11.9

シーン 1 は 90 フレーム，シーン 2 は 100 フレーム，シーン 3 は 350 フレームで画像サイズは 320×240 ピクセルである。

表 4.1 は実交通環境下における検出器と追跡器を用いた自転車検出の結果を表す。また，検出器だけを用いて実交通環境下における自転車検出を行った結果を表 4.2 に示す。検出の成功・失敗の判断には第 2 章の式(2.28)を用い，また評価指標には式(2.29)，式(2.30)，式(2.31)を用いる。

4.5 考察

表 4.1 及び表 4.2 の結果より，検出器と追跡器を両方を用いる方が検出率が向上することが分かる。特にシーン 3 の場合は検出率の精度が明確に向上した。シーン 3 の場合はオクルージョンが多数発生するシーンである。検出器だけを用いると，オクルージョンが発生すれば検出できない場合があるが，追跡器はオクルージョンが発生してもある程度適応できる。よって，検出器と追跡器を両方用いることにより精度が向上したと考えられる。シーン 1 では検出器だけを用いた場合の方が検出率が良い。シーン 1 の場合は自転車にオクルージョンが発生しないシーンである。シーン 1 の場合，追跡器の追跡誤差の影響で検出率が低下したと考えられる。

4.6 まとめ

本章では、HOG 特徴量を用いた自転車検出器と Particle Filter を用いた自転車追跡器を用いて、自転車の位置とスケールを推定する方法を提案した。また、実交通環境下の車載カメラ映像を用いた実験によって提案法の有効性を示した。

物体追跡の時、追跡法だけでは最初に追跡する物体がどこにあるか分からない。そのため近年は、最初に物体検出器が用いられることが多い。物体検出器で物体を検出して、それを追跡するという手法である。本章で提案した方法は、物体検出器を最初に用いるだけでなく、物体検出器と物体追跡器を両方継続して用いることにより、物体検出率の向上を図るという手法である。

第5章 結論

本論文では、実交通環境下における自転車検出手法を提案した。2章では、検出精度を向上するために HOG 特徴量を改良した MSC-HOG 特徴量を提案した。従来の HOG 特徴量や CoHOG 特徴量を用いた手法より MSC-HOG 特徴量は特徴を網羅的に抽出することができる。その結果、HOG 特徴量や CoHOG 特徴量を用いた自転車検出より、提案法の検出精度を上げることができた。また提案法は、識別の速度を上げるために Cascade 構造を用いた識別器を導入した。提案法を実交通環境下で撮影された映像に適用して自転車検出を行った結果、従来手法より有効な手法であることが示された。

3章では、オクルージョン問題の対策として、自転車車輪と人の上半身を各々検出して、その結果を統合することにより自転車を検出する手法を提案した。まず、円・楕円の検出を用いた自転車車輪の検出では、提案する円・楕円検出法が、円・楕円検出に一般的に用いられているハフ変換より高速で精度が良いことを示した。自転車は見る方向によって見かけの形が変化するため、すべての方向の自転車を検出することは難しい。しかし、提案法を用いれば、画像上で斜めから横方向までの自転車の検出が可能になる。また、正面方向から自転車車輪が楕円に見えない斜め方向までの自転車検出には、運転者の上半身を認識すれば良い。そのためには、上半身の下の画像領域が人の下半身かどうかを認識すれば良い。なお、提案した円・楕円検出は、車輪検出以外でも顔検出、瞳孔検出、円形の部品検出などの様々な分野に応用できると考えられる。

3章の実験によって、自転車の一部にオクルージョンが発生しても、3章で提案した手法が2章で提案した手法より高い自転車検出精度を持つことを確認した。

4章では、実交通環境下における自転車の位置とスケールの推定精度を向上するために自転車の検出器と追跡器を両方用いる手法を提案した。従来研究では最初検出器を用いて対象物体を検出し、検出された対象物体を追跡器を用いて追跡する方法が多いが、本手法では、検出器と追跡器を両方継続して用いることにより、検出器から検出される情報と追跡器から追跡される情報を補って物体の位置とスケールを推定することが可能である。実験によって、オクルージョンがある複雑な背景で、検出器だけを用いる手法より提案する手法の精度が良いことを示した。

今後の課題としては、自転車以外の様々な障害物を検出する方法の開発、カメラ(画像処理)のみの技術で先進安全自動車システムが商用化できるように、様々な状況でも良い精度を持つ物体認識手法の開発等が挙げられる。

参考文献

- [1] M. Wang and X. Wang, "Automatic adaption of a generic pedestrian detector to a specific traffic scene," Proc. of Conf. on Computer Vision and Pattern Recognition, pp. 3401-3408, 2011.
- [2] V. Atienze-Vanacloig, J. Rosell-Ortega, G. Andreu-Garcia, J. M. Valiente-Gonzalez, "People and luggage recognition in airport surveillance under real-time constraints," Proc. of Int. conf. on Pattern Recognition, pp.1-4, 2008.
- [3] C. Chen, J. Odobez, "We are not Contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video," Proc. of Conf. on Computer Vision and Pattern Recognition, pp. 1544-1551, 2012.
- [4] S. Scher, R. Crabb, J. Davis, "Making real games virtual: Tracking board game pieces," Proc. of Int. Conf. on Pattern Recognition, pp.1-4, 2008.
- [5] A. Geiger, P. Lenz, R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," Proc. of Conf. on Computer Vision and Pattern Recognition, pp. 3355-3361, 2012.
- [6] K. Takahashi, Y. Kuriya, T. Morie, "Bicycle detection using pedaling movement by spatiotemporal gabor filtering," International Journal of Innovative Computing Information and Control, vol. 8, no. 6, pp. 4059-4070, June 2012.
- [7] H. Cho, P. E. Rybski, W. Zhang, "Vision-based bicyclist detection and tracking for intelligent vehicles," Intelligent Vehicles Symposium, pp. 454-461, 2010.
- [8] Y. Nakashima, J. K. Tan, S. Ishikawa, T. Morie, "On detecting a human and its body direction from a video," Proc. of 15th Int. Symposium on Artificial Life and Robotics, pp. 294-297, 2010.
- [9] C. R. Wang, J. Wu, and J. J. Lien, "Pedestrian detection system using cascaded boosting with invariance of oriented gradients," International Journal of Pattern Recognition and Artificial Intelligence, vol. 23, no. 4, pp. 801-823, 2009.
- [10] F. Arnell, Vision-Based Pedestrian Detection System for use in Smart Cars, Master's Thesis, Stockholm, Sweden, 2005.
- [11] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," Proc. of Conf. on International Conference on Computer Vision 2, pp. 1150-1157, 1999.

- [12] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detections," Proc. of Conf. on Computer Vision and Pattern Recognition, pp.886-893, 2005.
- [13] Q. Zhu, S. Avidan, M. C. Yeh, K. T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," Proc. of Conf. on Computer Vision and Pattern Recognition, pp. 1491-1498, 2006.
- [14] H. Bay, T. Tuytelaars, L. V. Gool, "SURF: Speed Up Robust Features," Proc. of European Conf. on Computer Vision, pp. 404-417, 2006.
- [15] X. Wang, T. X. Han, S. Yan, "An HOG-LBP human detector with partial occlusion handling," Proc. of Int. Conf. on Computer Vision, pp. 32-39, 2009.
- [16] B. E. Boser, I. M. Guyon, V. N. Vapnik, "A training algorithm for optimal margin classifiers," Proc. of the First Annual Workshop on Computational Learning Theory 5, pp. 144-152, 1992.
- [17] C. R. Wang, J. Wu, and J. J. Lien, "Pedestrian detection system using cascaded boosting with invariance of oriented gradients," International Journal of Pattern Recognition and Artificial Intelligence, vol. 23, no. 4, pp. 801-823, 2009.
- [18] A. Bertoni, P. Campadelli, M. Parodi, "A boosting algorithm for regression," Proc. of the Seventh International Conference on Artificial Neural Networks, pp. 343-348, 1997.
- [19] P. Viola, M. J. Jones, "Robust real-time face detection," International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, May 2004.
- [20] S. Hussain, B. Triggs, "Feature sets and dimensionality reduction for visual object detection," British Machine Vision Conference, pp. 1-10, 2010.
- [21] M. Pedersoli, A. Vedaldi, J. Gonzalez, "A coarse-to-fine approach for fast deformable object detection," Proc. of Conf. on Computer Vision and Pattern Recognition, pp. 1353-1360, 2011.
- [22] H. Jung, J. K. Tan, S. Ishikawa, T. Morie, "Applying HOG feature to the detection and tracking of a human on a bicycle," Proc. of 11th Int. Conference on Control, Automation and Systems, pp. 1740-1743, 2011.
- [23] R. E. Schapire, Y. Singer: "Improved boosting algorithms using confidence-rated prediction", Machine Learning, vol. 37, no. 3, pp. 297-336, 1999.

- [24] T. Watanabe, S. Ito, K. Yokoi, "Co-occurrence of histograms of oriented gradients for human detection," *IPSJ Transactions on Computer Vision and Applications* 2, pp. 39-47, 2010.
- [25] J. Zhao, W. Qiao, G. Z. Men, "An approach based on mean shift and Kalman filter for target tracking under occlusion," *Int. Conf. Machine Learning and Cybernetics*, pp. 2058-2062, 2009.
- [26] Z. Li, Q. Tang, N. Sang, "Improved mean shift algorithm for occlusion pedestrian tracking," *Electronic Letters*, vol. 44(10), pp. 622-623, 2008.
- [27] X. Lu, L. Song, Y. Xu, S. Yu, "A novel occlusion-adaptive object tracking method," *Int. Conf. on Computer Science and Information Processing*, pp. 1117-1120, 2012.
- [28] N. Amezcua, R. Alquezar, F. Serratos, "Dealing with occlusion in a probabilistic object tracking method," *IEEE Computer Society Conference on CVPRW*, pp. 1-8, 2008.
- [29] S. Cao, G. Duan, H. AI, "Fast human detection using node-combined part detector," *Proc. of Int. Conf. on Image Processing*, pp. 3589-3592, 2011.
- [30] B. Wu, R. Nevatia, "Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detector," *Proc. of Int. Conf. on Computer Vision*, vol. 1, pp. 90-97, 2005.
- [31] B. Wu, R. Nevatia, "Detection and segmentation of multiple, partially occluded object by grouping, merging, assigning part detector responses," *Int. Journal on Computer Vision*, vol. 82, no.2, pp. 185-204, 2009.
- [32] J. Yao, N. Khanna and P. Grogono, "Fast robust GA-based ellipse detection," *Proc. Int. Conf. on Pattern Recognition*, pp. 859-862, 2004.
- [33] V. Ayala-Ramirez, C. H. Garcia-Capulin, A. Perez-Garcia, R. E. Sanchez-Yanez, "Circle detection on images using genetic algorithms," *Pattern Recognition Letters*, pp. 652-657, 2006.
- [34] B. Lamiro, O. Gaucher, L. Fritz, "Robust circle detection," *Int. Conf. on Document Analysis and Recognition*, pp. 526-530, 2007.
- [35] L. G. de la Fraga, G.M.L Dominguez, "Robust detection of several circles or ellipses with heuristics," *Congress on Evolutionary Computation*, pp. 484-490, 2011.

- [36] R. Zhao, "A fast and accurate detection algorithm for circle target," Int. Conf. on Automatic Control and Artificial Intelligence, pp. 54-56, 2012.
- [37] Y. Ito, W. Ohyama, T. Wakabayashi and F. Kimura, "Detection of eyes by circular Hough transform and histogram of gradient," Int. Conf. Pattern Recognition, pp. 1795-1798, 2012.
- [38] V. Vinoharan, A. Ramanan and S. R. Kodituwakku, "A wheel-based side-view car detection using snake algorithm," Int. Conf. Information and Automation of Sustainability, pp. 185-189, 2012.
- [39] N. Bhatia, M. Chhabra, "Improved Hough transform for fast iris detection," Int. Conf. on Signal Processing Systems, pp. 172-176, 2010.
- [40] S. Huang, C. Gao, S. Meng and Q. Li, "Circular road sign detection and recognition based on Hough transform," Int. Congress on Image and Signal Processing, pp. 1214-1218, 2012.
- [41] Y. Jin, F. Mokhtarian, "Variational particle filter for multi-object tracking," Int. Conf. on Computer Vision, pp. 1-8, 2007.
- [42] K. Nummiaro, E. Koller-meier, L. V. Gool, "A color-based particle filter," Proc. of European Conf. on Computer Vision, pp. 53-60, 2002.
- [43] W. Abd-Almageed, M. Husseinm, M. Abdelkader, and L. Davis, "Real-time human detection and tracking from mobile vehicle", IEEE Intelligent Transportation Systems Conference, pp. 149-154, 2007.

謝辞

本研究を進めるにあたり、終始御指導をいただきました石川聖二教授に心から感謝いたします。ならびに、研究のご支援とご指導を賜りましたタンジュークイ準教授に深く感謝いたします。また、本研究に対して多くのご助言をいただきました森江隆教授、金亨燮教授、黒木秀一教授にこの場を借りてお礼申し上げます。

また、石川研究室の皆様にはお忙しい中、研究と実験に対して多くのご協力とご助言をいただきまして深く感謝致します。

最後に博士前期・後期課程の4年半の間、先輩方と後輩、留学生の皆様との出会いで大変有意義な時間を過ごさせていただいた石川研究室に厚くお礼を申し上げます。

本当にありがとうございました。