ドライブレコーダー映像を用いた頭部検出に基づく人流推定法の提案

原 佑輔[1] 小島 颯平[1] 内山 彰[1] 梅津 高朗[2] 山口 弘純[1] 東野 輝夫[1]

概要：本研究では，近年普及が進んでいるドライブレコーダーの映像を用いた歩道レベルので人流推定法を提案する．歩道上での歩行者の移動方向は前方と後方の 2 種類に大別されるため，提案手法では，遮蔽されにくい歩行者の頭部前方，後方 2 種類の検出器を機械学習により構築する．しかし，車載カメラの設置位置は一般的に歩行者の顔程度の高さであるため，映像内で歩行者同士の重なりが起こりやすい．そこで時間的に連続する複数フレームにおいて断続的に検出される歩行者を追跡するためのアルゴリズムを考案し，遮蔽に対してロバストな頭部検出を実現する．提案手法の性能を評価するため，大阪市茶屋町周辺で撮影した 64 分間の映像を収集し，実験を行った．その結果，前方の歩行者検出の Precision は 74.6%，Recall は 66.6%となり，後方の歩行者検出の Precision は 64.4%，Recall は 73.0%となることが分かった．

Estimating Pedestrian Flows Based on Head Detection Using Drive Recorders

Yusuke Hara[1] Sohei Kojima[1] Akira Uchiyama[1] Takaaki Umedu[2] Hirozumi Yamaguchi[1] Teruo Higashino[1]

1. はじめに

マーケティング，都市計画，安全支援など，様々な目的において都市部における人流（歩行者の分布および移動状況）を把握することは重要である．例えば，把握した人流から人気のあるスポットを検出したり，混雑状況の監視・予測に基づく人流誘導を行う他，災害時の帰宅困難者の人流に応じた救援計画の立案にも活用できる．

このため，これまでに様々な手法が提案されている．例えばモバイル空間統計 [1] では携帯電話の通信統計情報を用いて区画毎の人口推定を行っている．また，混雑度マップ [2] では GPS 対応の携帯電話利用者から許諾を得て送信される位置情報の分布からの人口推定を行っている．しかし，いずれも 250m メッシュなど一定範囲ごとの人密度を推定する手法であり，“ある道路の左側を駅方向に歩く人数” といった詳細な人流を把握する試みは見当たらない．一方，防犯カメラ映像を用いて混雑状況を推定する手法 [3], [4], [5], [6] も存在する．これらの手法では，背景差分法やエッジ検出法などにより抽出された群衆密度と相関のある特徴量を基に混雑状況の推定を行う．また，映像内の人物をトラッキングすることで，人流を推定する手法も存在 [7] する．これらの固定カメラを用いた手法で都市部全体の人流を把握するためには，膨大な数のカメラを設置する必要があり，設置場所やコストの制約上，現実的ではない．

そこで本研究では，近年普及が進んでいる車載カメラの映像を用いた歩道レベルでの人流推定法を提案する．様々な道路を走行している複数の車両で撮影された映像から，歩道の人流を推定し，各車両の位置情報と共にサーバーで集約・統合することで，都市部全体における人流把握を実現する．このためには，以下のような課題を解決しなければならない．まず，都市部では看板などの様々な背景が存在するが，背景画像が用意できないため歩行者検出そのものが困難である．次に，高所に設置される監視カメラと異

[1] 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[2] 滋賀大学 経済学部
Faculty of Economics, Shiga University

なり，車載カメラの設置位置は一般的に歩行者の顔程度の高さであるため，映像内で歩行者同士の重なりが頻繁に起こる．このため，全身が映ることはほとんどない上に，歩行者一人一人が映像内に現れるタイミングは断続的となる．また，車と歩道の間に存在する植え込みや柵などによっても，身体の一部が遮蔽される．

車載カメラを用いた既存の歩行者検出法 [8], [9], [10] は運転支援を目的としているため，全身が映っていない歩行者の検出は考慮していない．提案手法では，遮蔽されにくい頭部を対象に，機械学習により検出器を構築することで，歩行者一人一人の検出を試みる．歩道上で歩行者は前方と後方のいずれかにしか移動しないものと仮定し，歩行者の移動方向推定には前方と後方の 2 種類の頭部検出器を利用する．歩行者同士の重なりに対してロバストな推定を実現するため，時間的に連続する複数フレームの検出結果を利用する．車の移動に伴い，歩行者を撮影する角度が変化するため，単一のフレームで歩行者同士が重なっている場合でも，その前後のフレームでは隠れている歩行者がカメラに映ると考えられる．そこで複数フレームの検出結果に対して断続的に検出される歩行者のトラッキングを行うことで，歩行者数を推定する．走行している車両の移動速度は歩行者よりも非常に速いことから，複数フレーム間で歩行者はほぼ移動しないものと仮定し，頭部検出の位置に基づくトラッキングアルゴリズムを考案した．

性能評価のため，大阪市茶屋町周辺で撮影した 64 分間の映像を用いて実験を行った．その結果，前方の歩行者検出の Precision は 74.6%，Recall は 66.6%となり，後方の歩行者検出の Precision は 64.4%，Recall は 73.0%となることが分かった．

2. 関連研究

2.1 車載カメラを用いた人検出

自動運転車に関連する技術の発展とともに，安全運転支援を目的として，車載カメラを用いた歩行者検出が数多く研究されている．これらの手法は，人の動きを検出する方式と人の形状を検出する方式の 2 種類に大別される．文献 [8], [11], [12], [13] では人特有の動きのパターンを特徴量として歩行者を検出する．しかし，これらの手法は動きのパターンを抽出するために歩行者の足が一定時間見えている必要がある．また，人の動きを用いて検出を行っているため静止している歩行者は検出することができない．

一方，人の形状を特徴量として歩行者を検出する手法は移動している人と静止している人の両方を検出することができる．文献 [9] ではウェーブレット解析 [14] と Support Vector Machine(SVM)[15], [16], [17] を用いて歩行者検出を行っている．これらの手法は運転支援を目的としており，群衆中では人同士の重なり (オクルージョン) が大きく影響し，検出精度が低くなるという問題が生じる．

2.2 固定カメラによる群衆人数推定

監視カメラなどの固定カメラの映像から様々な群衆人数を推定する手法として，画像から特徴量を抽出して検出を行う方法がある．文献 [3] では，これらの手法をピクセル解析法，テクスチャ解析法，オブジェクト解析法，の 3 種類に大別している．ピクセルレベル解析法は背景差分法やエッジ検出などの非常に局所的な特徴を用いて推定を行う．文献 [4] では背景差分法により人の存在しうる領域を抽出し，前景の物体のピクセルと，エッジ検出により抽出されたエッジを入力として線形モデルを適用し，人数推定を行う．文献 [5] では前景の物体のピクセル数に基づき群衆の密度を推定する．この手法は幾何補正を適用した画像では画像中の人数とピクセル数は線形関係にあることを利用している．しかし，これらの手法は線形モデルを用いているためにオクルージョンが発生した場合に人数の増加に伴い線形性は低くなるため，推定精度が低下する．

一方，テクスチャ解析では画像における一定の大きさの領域に対する解析を行う．文献 [18] ではテクスチャの特徴量であるグレーレベル同時生起行列 [19] を用いて群衆の密度を SVM により分類する手法が提案されている．文献 [6] では人物同士の重なりを含めた群衆の状態を複数の人物画像を合成した局所画像 (歩行者領域画像) によって表し，Convolutional Neural Network[20] によって学習することで群衆人数を推定する．

オブジェクト解析法では，画像中の人を識別することで群衆人数を推定する．文献 [21] では一定期間内において人の動きが検出された領域を人が存在しうる領域として，その領域内において HOG 特徴量を用いた上半身の検出器を用いて人を識別して群衆の人数を推定する．

以上の手法では，いずれもカメラが広範囲を見渡せる高所にて設置されていることを想定しており，車載カメラ映像にそのまま適用した場合，頻繁なオクルージョンの影響を受けやすい．また，街中では様々な背景が存在するため，背景画像の無い状況への適用は難しい．

2.3 モバイル端末による群衆人数推定

スマートフォンなどのモバイル端末を用いて群衆密度を推定する手法も検討されている．文献 [22], [23] では，近隣のスマートフォン間で Bluetooth による無線通信を行い，通信範囲内の端末数や電波強度のばらつきをもとにユーザーの周囲の群衆密度推定を行う．モバイル端末間で音声ビーコンによる通信を行うことにより，環境内に存在する端末の数を高精度に推定するアプローチも提案されている [24]．端末間の通信を用いた手法は，端末密度が高い環境でもロバストな推定を行うことができるが，端末を保持していない人やサービスに参加していない人の存在を認識できない．文献 [25], [26] ではスマートフォンから得られた GPS 情報や加速度情報を基に得られた歩行者の移動速度

から群衆密度を推定を行っている．スマートフォンに搭載された加速度センサおよびマイクの計測値から，端末周辺の混雑状況を把握する手法も提案されている [27]．この手法では，周囲の端末との通信を必要としないため端末を保持しているユーザが少ない環境でも推定が可能であるが，混雑度合いを 3 段階で分類しており，より詳細な人数の推定を行うことはできない．文献 [28] では携帯端末が接続している基地局情報を利用して，都市間の人の移動モデルを作成することで，群衆密度を推定している．端末間の近接通信やセンシングなどを行うことなく低コストに群衆密度を推定できるが，基地局測位の推定精度は数百 m 程度であり，歩道単位での群衆人数の推定を目的とする提案手法とは，想定するスケールが大きく異なる．また文献 [29] ではスマートフォンによって比較的高所から撮影された俯瞰画像を用いて群衆人数推定を行う方法が提案されている．非常に短い時間間隔で撮影された 2 枚の画像の差分をとることで群衆が存在する動体領域を抽出し，得られた人が存在し得る動体領域面積と歩行者 1 人あたりの占有面積に基づき群衆人数の推定を行っている．しかし，この手法は人同士の重なりが比較的少ない俯瞰画像を用いることを想定しており，本研究のような車載カメラ映像を用いた場合，性能が大きく低下する．また，カメラ自体が移動するため，画像差分による動体領域の検出が難しいという問題もある．

3. 提案手法

3.1 システム概要

提案システムの概要を図 1 に示す．システム参加ユーザは車載カメラを搭載した車両で道路を走行する．車載カメラとしては，ダッシュボードにマウントされたスマートフォンやスマートフォンと接続可能な後付けのドライブレコーダーなどを想定している．撮影された映像は，各車両のスマートフォンなどにより処理され，方向別の歩行者数が推定される．推定結果は歩道の位置と撮影時刻とともにサーバーに送信される．サーバーでは，複数車両から送られてきた各地の歩道における人流を統合し，地図上にマッピングする．

歩道の位置は GPS などにより得られた車両位置と映像内の歩行者の検出位置（左側または右側）から容易に把握可能である．ただし提案手法では頭部のみを検出対象とすることから，映像内の歩行者が小さいと検出に十分な特徴量が得られない．このため本研究では，走行車線に近い側の歩道（日本では左側）でかつ車両から一定距離内のみを推定対象とし，車両が歩道に最も近い車線を走行している時のみ，人流の推定を行う．このため，映像中の頭部が存在しうる範囲（推定対象範囲）を図 2 のように定め，この範囲があらかじめ分かるものとした．また，交差点では前方，後方以外にも様々な方向に歩行者が移動する可能性があり，その区別は困難であるため，交差点間の歩道のみを



図 1 システム概要



図 2 推定対象範囲

対象とする．

実際には，歩道に最も近い車線から歩道までの距離は柵や植え込みなどの配置によって様々であり，推定対象範囲を常に固定することは適切でない．このため，提案システムの実現には，走行車線の認識や歩道までの距離を認識する手法も必要となるが，本研究の対象とはしない．直感的な実現方法としては，文献 [35] で提案されている方式によって横断歩道や車線といった白線を検出したり，広範囲に対して顔検出を行い，頻繁に顔が検出される範囲の周辺を当該道路走行時の対象範囲とする，といったことが考えられる．

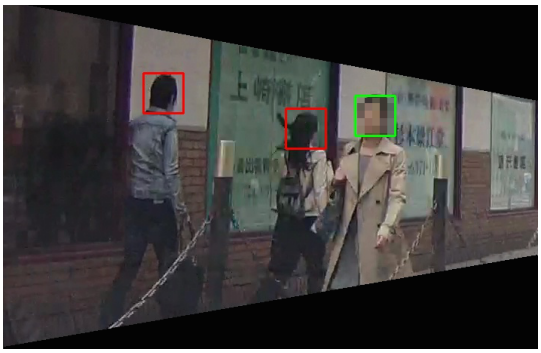


図 3　頭部検出結果の例

3.2　人流推定法の概要

地理的に隣接している 2 つの交差点 i, j を直接結ぶ道路沿いにある歩道 r_{ij} の i, j 間を人流推定の空間的な単位とする．歩道は道路の両側にあるが，以降では車両の走行車線に近い側の歩道のみを対象として説明する．単位時間あたりに歩道 r_{ij} 上を i から j（j から i）へ移動している人数 $f_{ij}(f_{ji})$ を求めることが，本研究の目的である．

車載カメラ映像の 1 フレームには，r_{ij} の一部のみが推定範囲 r'_{ij} に収まる．したがって，車両が交差点 i から j に向かって走行している場合，連続するフレームで r_{ij} の異なる範囲が撮影される．r_{ij} 全体の人流 f_{ij} は r_{ij} の長さ $|r_{ij}|$ と 1 フレームの推定結果から得られる r_{ij} の一部の範囲あたりの人流の平均値 f'_{ij} より，以下の式で求める．

$$f_{ij} = \frac{f'_{ij}|r_{ij}|}{|r'_{ij}|} \qquad (1)$$

$|r'_{ij}|$ は 1 フレームで推定範囲に収まる歩道 r_{ij} の一部の長さである．逆方向の人流 f_{ji} も同様に求められる．以降では，頭部検出器を用いたフレームごとの人流推定法について説明する．

3.3　フレーム内人流推定

歩行者同士の重なりや障害物による遮蔽が存在する都市環境においてロバストな人流推定を実現するため，提案手法では撮影された映像のうち，時間的に連続する w フレームに対して歩行者頭部のトラッキングを行う．頭部が映像中に断片的に現れる場合であっても，複数フレームを利用すれば一人一人の歩行者を軌跡として検出できる可能性がある．

提案手法では，まず既存の物体検出法を用いて $[1, w]$ 内の各フレームに対し，頭部の前方（顔）および後方をそれぞれ検出する．検出結果は図 3 に示すような矩形領域の座標として得られる．以降では，この矩形領域の中心座標を検出結果の位置座標とする．前述のように車両の移動速度が歩行者と比べて非常に速い状況を想定しているため，一人の歩行者が推定対象範囲に映る期間は短く，歩行者はほぼ同一地点にいると考えられる．一方，車両はフレーム間に一定距離を移動するため，映像中の歩行者の位置が車両の移動距離に応じて変化する．

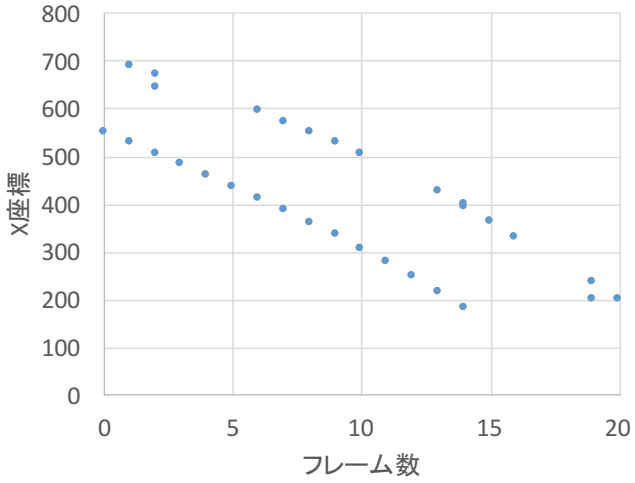


図 4　頭部 x 座標の軌跡

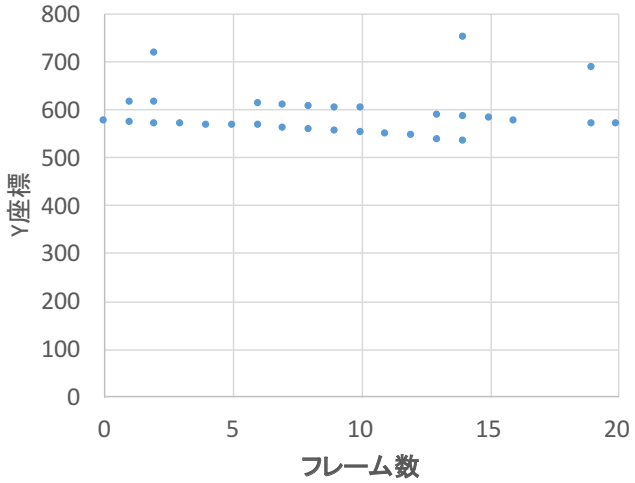


図 5　頭部 y 座標の軌跡



図 6　瞬間的な誤検出の例

この事実を確認するため，画像の横（x 軸），縦（y 軸）方向について，連続する 20 フレーム（0.67 秒間）それぞれで検出された前方頭部の位置座標を図 4，5 に示す．なお，画像の左上が原点であり，右方向，下方向が正の方向

である．図から分かるように，y 軸方向については時間の経過にともなう変化がほとんど見られないものの，x 軸方向については連続するフレームで車両の移動に合わせて映像中の検出結果の座標が移動している．これは，歩道上のほぼ同一地点で複数フレームにわたって前方頭部が検出されていることを意味する．検出結果の座標が線形ではなく緩やかにカーブしているが，この理由はレンズの歪みや，カメラからの距離によって x 座標が原点に近付く（車両に近付く）ほど，1 ピクセルあたりの解像度が高く（距離が短く）なるためである．一方，瞬間的に特定のフレームのみで検出され，前後のフレームでは同一地点での検出結果が得られない場合も存在するが，これらは瞬間的な誤検出の場合がほとんどである．実際，図 6 は図 4 のフレーム 2 における検出結果を表しており，$x = 640$ 付近に検出された頭部は誤検出であることが分かる．

以上より，提案手法では連続する w フレームにおける前方，後方それぞれの頭部検出結果に対し，歩道上の同一地点にあると考えられる検出結果を対応づけることによって，歩行者のトラッキングを行う．以降の節では，各フレームにおける頭部検出法およびトラッキングのアルゴリズムの詳細を述べる．

3.3.1 頭部検出

提案手法では前方と後方の 2 種類に分け，それぞれの頭部検出器を作成する．特徴量には Haar-Like 特徴量 [36], [37] を使用し，Adaboost[32] で学習を行う．Haar-Like 特徴量では，局所的な明暗差の組み合わせによって検出対象を学習する．与えられた画像中に存在する可能性のある検出対象は学習した対象と大きさが異なるため，ウィンドウサイズをあらかじめ定めた範囲で変化させながら網羅的に画像を走査し，学習した対象との類似度が一定以上あれば，そのウィンドウを仮検出結果として保存する．最終的に，全てのウィンドウサイズでの走査が完了した後，位置的にわずかでも重複する仮検出結果が一定数 $N_{windows}$ 以上あれば，その平均位置とサイズを最終的な検出結果とする．

3.3.2 歩行者トラッキングアルゴリズム

歩行者トラッキングアルゴリズムの擬似コードを Algorithm 1 に示す．歩行者の場合，数フレームが撮影される間にほぼ動かないことから，複数フレームにわたる同一歩行者の検出結果は，映像内では車の移動分だけ見かけ上移動することになる．したがって，提案するトラッキングアルゴリズムでは，複数フレームにわたって車の移動分だけ画像内の位置が変化している頭部検出結果を同一歩行者と見なし，トラッキングを行う．なお，提案手法では前方，後方それぞれの頭部検出結果が得られるが，本稿ではそれぞれ独立に歩行者軌跡を推定するものとしている．実際には，同一人物が両方の検出器で検出されてしまう場合も起こり得るため，より尤度の高い検出結果を採用することで，どちらか一方を選択する必要があるが，本稿では対象とせず，個別の軌跡検出に焦点を当てている．

Algorithm 1 x 軸上の歩行者トラッキング

Input: x-coordinates X^i of detected points in each frame i
Output: set T of estimated trajectories T_n

```
T = φ
for each point x ∈ X^1 on x-axis do
    add x to T as a new trajectory
end for
for each frame i from 2 to w do
    for each newest point t̂_j ∈ T_j do
        for each point x ∈ X^i do
            if |D − |x − t̂_j| / (i − F(t̂_j))| < ε ∧ i − (F(t̂_j)) < F then
                add x to T_j after t_j
                remove x from X^i
            end if
        end for
    end for
    while X^i ≠ φ do
        add x ∈ X^i to T as a new trajectory
        remove x from X^i
    end while
end for
for each T_i ∈ T do
    if number of points in T_i < L then
        remove T_i from T
    end if
end for
```

まず，連続する w フレームのうち，1 番目のフレームに対する頭部検出結果の x 座標の集合 X^1 を歩行者軌跡 T_i として保存する．これらの歩行者軌跡に対し，次フレームの検出結果 X^2 の中で，X^1 の歩道上での位置とほぼ同じ位置 x に検出された結果があれば，同一人物に対する検出結果と見なし，軌跡 T_i の最後尾に追加する．実際には，障害物や他の歩行者によって同一人物が検出されない場合もあるため，1 番目のフレームから順に w 番目のフレームまで走査していき，F フレーム以内に歩道上での同一位置に検出された結果が存在すれば，同一人物に対する検出結果と見なす．Algorithm 1 中の $\mathcal{F}(\hat{t_j})$ は検出結果の x 座標 $\hat{t_j}$ が得られたフレーム番号を表す．D は 1 フレームあたりの車両の移動によって生じる同一人物の映像内座標での変位である．D は車両の移動速度や映像内の人物の位置によって変化するが，本稿ではほぼ同一の速度で走行している場合のみを想定し，$D = -18$pixel とした．ただし，実際には検出位置がずれたり，人物がわずかに移動するため，これらを許容するパラメータ ϵ を導入している．ϵ は経験的に 20pixel とした．フレーム w までの走査が完了した後，最後に極端に短い歩行者軌跡は誤検出と見なして除外する．本稿では，検出数の閾値 $L = 5$ と定め，軌跡中の検出数が L に満たないものは除外することとした．



図 7　実験で走行した道路 (大阪市茶屋町周辺)

4.　性能評価

4.1　実験環境

提案手法の性能評価を行うため，大阪市茶屋町周辺の道路をドライブレコーダ (ユピテル社製 DRY-WiFiV5c) を設置した自動車 3 台で複数回通行し，映像を撮影した．撮影は，図 7 の地点 1 から地点 2 間の水色で示されている道路で 2016 年 3 月 20 日 (日) と 2016 年 3 月 21 日 (月) の正午頃に行った．撮影した映像から交差点付近や歩道に最も近い車線を走行していない場合の映像を目視で確認し除外した後，各フレームにおける人の頭部を前方と後方に分類してクリップし，それぞれ 500 枚ずつ学習データに用いた．また，BingAPI を用いて道路や風景など人の写っていない写真を 3000 枚収集し，学習用の非正解データとして用いた．

また，学習データとは別の映像に対し，30 フレームごとに切り出した 68 フレームをテストデータとし，各フレームにおける前方，後方それぞれの頭部位置を手動で指定することで，方向別の頭部位置の真値を得た．これらのフレームに存在する歩行者数の合計は前方 75 名，後方 89 名である．これに基づき，検出結果の正しさを示すため，評価指標としては人流（方向別人数）ではなく，Precision, Recall, F 値を用いる．F 値は Precision と Recall の調和平均であり，

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (2)$$

と定義される．

これらの評価指標を算出するため，テストデータのフレームを中心とする w フレームに対して提案手法を適用し，得られた軌跡から補間したテストデータのフレームに

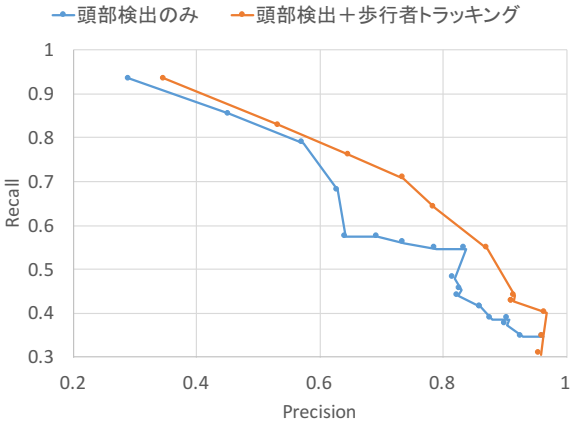


図 8　$N_{windows}$ を変化させたときの前方頭部検出の結果

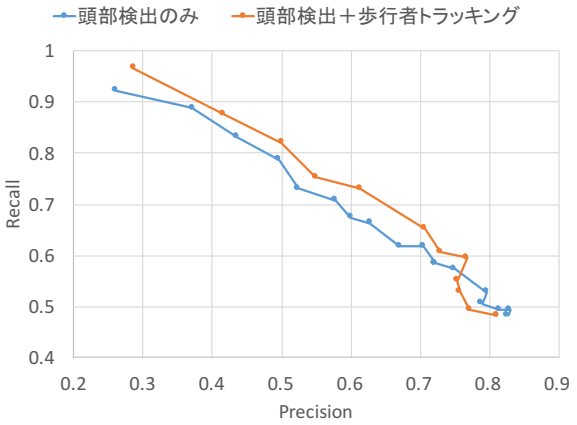


図 9　$N_{windows}$ を変化させたときの後方頭部検出の結果

おける頭部位置に対し，評価を行った．軌跡推定に利用するフレーム数 w は 10 とした．テストデータでクリップした頭部領域と検出結果の矩形領域が小さい方の領域の面積の 20%以上重なっていれば，検出結果は正しいものと定義する．頭部検出器での走査ウィンドウサイズは，対象範囲において想定される頭部の大きさの範囲内（30×30[pixel] から 80×80[pixel]）で，ウィンドウサイズを 10%ずつ拡大させた．画像処理の実装には Python および画像処理ライブラリ OpenCV2.4.12 を用い，機械学習には OpenCV3.1.0 のツールを使用した．実行環境は OS X 10.10.5 である．

4.2　軌跡による補間の効果

提案手法の軌跡による補間の効果を調べるため，頭部検出器のみを用いた場合と，軌跡による補間を行った場合の Precision, Recall を比較した．Precision, Recall を調整するパラメータとして，頭部検出器におけるの近傍矩形数の最小値である $N_{windows}$ を 1 から 19 まで変化させる．一般的に $N_{windows}$ の値が小さいと検出結果の Precision は低いが Recall は高くなり，値が大きくなると Precision は高く，Recall は低くなる．

前方，後方それぞれの評価結果を図 8, 9 に示す．どちら

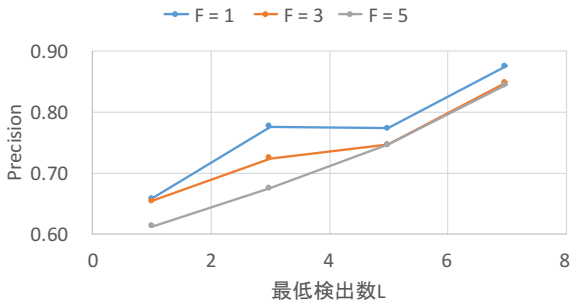


図 **10**　パラメータに対する Precision（前方）

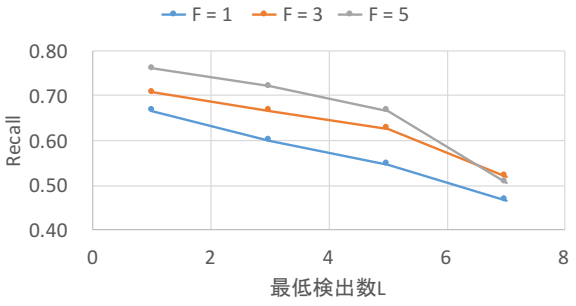


図 **11**　パラメータに対する Recall（前方）

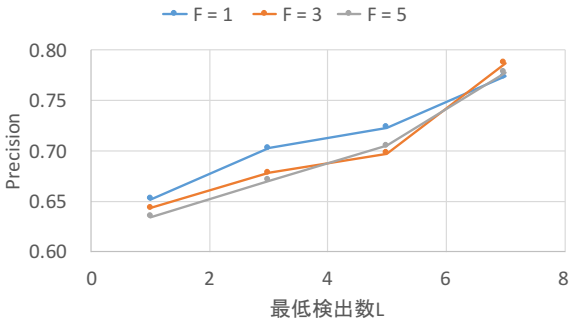


図 **12**　パラメータに対する Precision（後方）

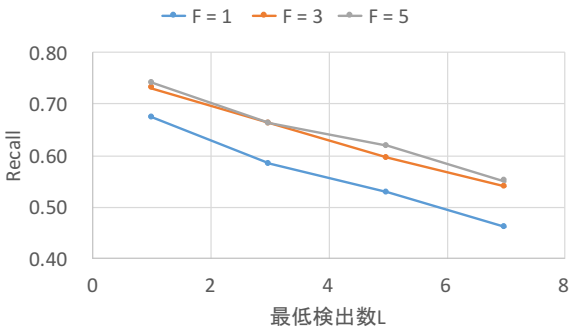


図 **13**　パラメータに対する Recall（後方）

の場合も提案手法により Precision, Recall が向上していることが分かる．それぞれの F 値を算出すると，前方の場合は頭部検出のみの F 値の最大が 0.663 であるのに対し，提案手法は 0.721 であったことから，軌跡推定による補間が有効であることが分かる．一方，後方の場合は，頭部検出のみの F 値の最大は 0.658 であるのに対し，提案手法は 0.678 となり，わずかながら提案手法により F 値が向上している．この原因として，頭部後方は前方と比べて後頭部しか情報量が無く，特徴量が非常に少ないことが挙げられる．このため，たとえ遮蔽されていない頭部後方が映っていても，検出そのものが困難となっている．これを解決するためには，検出器の学習データを増やしたり，物理的に頭部が存在する可能性がある検出対象範囲をより狭め，誤検出を抑えるといった工夫が求められる．

前方頭部の検出結果のみを見ると，Recall が 0.6 付近ではほぼ一定で，Precision が 0.6 から 0.8 程度まで向上している区間がある．これは，頭部前方においては目や鼻，口などの様々な特徴量が検出できるため，誤検出しやすい物体と比べて明確な差異があるためと考えられる．したがって，実際に歩行者が存在する場合には様々なウィンドウサイズで仮検出結果が得られ，$N_{windows}$ を大きくすることで，Recall を保ったまま Precision を高めることができている．ただし，車載カメラで撮影された映像では，遮蔽の影響で顔の特徴が明確に現れない場合も頻発するため，0.6 付近で急激な Recall の低下が見られる．一方，提案手法によりこのような急激な Recall の低下が抑えられており，軌跡による補間の有効性が見て取れる．

以上の結果より，提案手法により特に前方頭部について Precision, Recall を向上できることが分かった．以降の評価では，提案手法での $N_{windows}$ の値は F 値が最大であった時の値とし，（前方 4，後方 8）とする．

4.3　トラッキングパラメータの影響

トラッキングアルゴリズムのパラメータとして，3.3.2 節で述べた欠損許容フレーム数 F，および軌跡中の最低検出数 L が存在する．これらのパラメータの影響を確認するため，それぞれのパラメータを変化させた時の方向別の Precision，Recall を図 10,11,12,13 に示す．

いずれの場合においても L の値が大きくなれば Precision も大きくなり Recall は小さくなる．これは L の値が大きくなれば歩行者と見なすために必要な検出数が多くなるため，多くのフレームで検出されているより信頼性の高いもののみが歩行者であると推定されるためだと考えられる．また，ほとんどの場合において $F = 1$ の場合は $F = 3, 5$ の場合と比べて Precision は高く，Recall は低くなっている．これは時間的に連続する複数フレームで頻繁に検出されるような信頼性が高い結果のみを歩行者の軌跡として推定しているためである．これらのパラメータは求められる Recall, Precision に応じて適切に設定する必要があるが，本研究の目的である人流推定という観点では，Recall, Precision 両方のバランスが求められる．F 値が最大となる場合は，前方については $F = 5, L = 5$，後方について

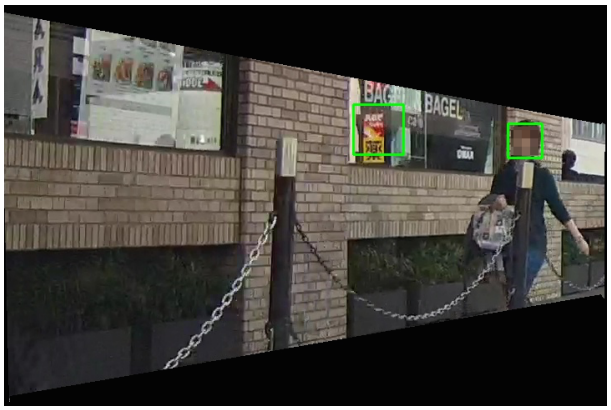


図 **14**　誤検出例

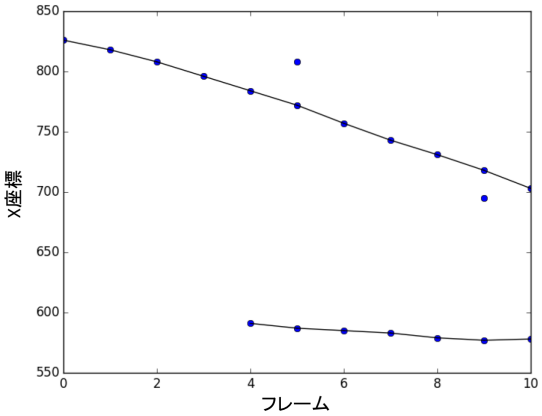


図 **15**　誤検出をした時のトラッキング結果

は $F = 3, L = 1$ であった．このとき，それぞれの F 値は 0.704, 0.684 であり，Precision, Recall はそれぞれ前方が 74.6%, 66.7%，後方が 64.4%, 73.0%であった．

4.4　誤検出の例

図 14 に提案手法による誤検出の例を示す．緑の枠で囲っている領域が前方の頭部と検出されたものだが，ガラスに写った看板が誤検出されている．この時の前後のフレームにおける歩行者トラッキングの結果を図 15 に示す．上部の軌跡は実際に歩行者であるが，下部の軌跡は誤検出された看板の軌跡である．このように連続したフレームで同じ物体を誤検出すると，トラッキングアルゴリズムでは判別ができない．このような誤検出を減らすには，看板などの様々な不正解データをより多く与えて検出器の精度を向上させることが必要不可欠であると考えている．

5.　おわりに

本研究では，街中を走行する車両の車載カメラ映像を利用した歩道レベルでの人流推定法を提案した．提案手法では遮蔽されにくい頭部を対象とし，Haar-Like 特徴量に基づく検出器を構築することで歩行者一人一人の検出を行う．歩行者の移動方向を推定するため，検出器は頭部の前方，後方それぞれに対して構築した．歩行者同士の重なりに対してロバストな推定を実現するため，時間的に連続する複数フレームの検出結果に対して，断続的に検出される歩行者のトラッキングを行うアルゴリズムを考案した．

性能評価のため，大阪市茶屋町周辺で撮影した分間の映像を用いて実験を行った．その結果，前方の歩行者検出の Precision は 74.6%，Recall は 66.6%となり，後方の歩行者検出の Precision は 64.4%，Recall は 73.0%となることが分かった．

今後，歩道単位での人流推定に対する性能を評価するための実験を計画している．また，今後の課題として，標識や服などの誤検出を減らすために，誤検出されやすい物体を不正解データとして学習させ，検出器そのものの精度向上を図ることが挙げられる．車両の速度や歩道までの距離に応じて適切にパラメータを変化させることで，手法の適用可能な対象環境を広げることも検討している．

参考文献

[1] 寺田雅之，永田智大，小林基成：モバイル空間統計における人口推計技術 (社会・産業の発展を支える 「モバイル空間統計」：モバイルネットワークの統計情報に基づく人口推計技術とその活用)，NTT DoCoMo テクニカル・ジャーナル， Vol. 20, No. 3, pp. 11–16 (2012).

[2] 株式会社ゼンリンデータコム：混雑度マップ，`http://lab.its-mo.com/densitymap/`.

[3] Silveira Jacques Junior, J., Musse, S. and Jung, C.: Crowd Analysis Using Computer Vision Techniques, *IEEE Signal Processing Magazine*, Vol. 27, No. 5, pp. 66 – 77 (2010).

[4] Davies, A. C., Yin, J. H., Velastin, S. et al.: Crowd monitoring using image processing, *Electronics & Communication Engineering Journal*, Vol. 7, No. 1, pp. 37–47 (1995).

[5] Ma, R., Li, L., Huang, W. and Tian, Q.: On pixel count based crowd density estimation for visual surveillance, *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems*, pp. 170–173 (2004).

[6] 池田浩雄，大網亮磨：群衆パッチ学習に基づく人数推定，第 12 回情報科学技術フォーラム (FIT2013)， Vol. 3, pp. 129–130 (2013).

[7] Wu, Z., Thangali, A., Sclaroff, S. and Betke, M.: Coupling detection and data association for multiple object tracking, *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1948–1955 (2012).

[8] Wöhler, C., Anlauf, J. K., Pörtner, T. and Franke, U.: A Time Delay Neural Network Algorithm for Real-Time Pedestrian Recognition, *Proceedings of International Conference on intelligent vehicle*, pp. 247–251 (1998).

[9] Papageorgiou, C., Evgeniou, T. and Poggio, T.: A Trainable Pedestrian Detection System, *Proceedings of Intelligent Vehicles*, pp. 241–246 (1998).

[10] Lee, K.-H., Hwang, J. N., Okapal, G. and Pitton, J.: Driving recorder based on-road pedestrian tracking using visual SLAM and Constrained Multiple-Kernel, *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 2629–2635 (2014).

[11] Mori, H., Charkari, N. and Matsushita, T.: On-line vehicle and pedestrian detections based on sign pattern, *IEEE Transactions on Industrial Electronics*, Vol. 41, No. 4, pp. 384–391 (1994).

[12] Niyogi, S. and Adelson, E.: Analyzing and recognizing walking figures in XYT, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 469–474 (1994).

[13] Niyogi, S. and Adelson, E.: Analyzing gait with spatiotemporal surfaces, *Proceedings of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 64–69 (1994).

[14] 山口昌哉：ウェーブレット解析，科学，Vol. 60, pp. 398–405（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/10006233574/⟩ (1990).

[15] Vapnik, V.: *The nature of statistical learning theory*, Springer Science & Business Media (2000).

[16] Bradski, G. and Kaehler, A.: 詳解 OpenCV: コンピュータビジョンライブラリを使った画像処理・認識，O'Reilly Media, Inc. (2009).

[17] Bishop, C. M.: パターン認識と機械学習 (上)，シュプリンガー ジャパン，1 edition (2007).

[18] Wu, X., Liang, G., Lee, K. K. and Xu, Y.: Crowd density estimation using texture analysis and learning, *Proceedings of IEEE International Conference on Robotics and Biomimetics*, pp. 214–219 (2006).

[19] Haralick, R., Shanmugam, K. and Dinstein, I.: Textural Features for Image Classification, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-3, No. 6, pp. 610–621 (1973).

[20] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324 (1998).

[21] Li, M., Zhang, Z., Huang, K. and Tan, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection, *Proceedings of the 19th International Conference on the Pattern Recognition*, pp. 1–4 (2008).

[22] Weppner, J. and Lukowicz, P.: Collaborative Crowd Density Estimation with Mobile Phones, *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pp. 193 – 200 (2011).

[23] Ghose, A., Bhaumik, C. and Chakravarty, T.: BlueEye: A System for Proximity Detection Using Bluetooth on Mobile Phones, *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, pp. 1135–1142 (2013).

[24] Kannan, P. G., Venkatagiri, S. P., Chan, M. C., Ananda, A. L. and Peh, L.-S.: Low cost crowd counting using audio tones, *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '12)*, pp. 155–168 (2012).

[25] 山本友理，屋代智之，重野　寛，岡田謙一：歩行者用道路上におけるリアルタイムな混雑情報の取得・提供手法 (セッション 2：ユビキタス 2)(ユビキタスモバイルサービス)，電子情報通信学会技術研究報告，pp. 37–42 (2004).

[26] 米村淳，大岸智彦，井戸上彰，小花貞夫ほか：スマートフォンを用いた人の混雑度推定手法の提案と評価，情報処理学会研究報告，pp. 1–8 (2013).

[27] Nishimura, T., Higuchi, T., Yamaguchi, H. and Higashino, T.: Detecting smoothness of pedestrian flows by participatory sensing with mobile phones, *Proceedings of the 2014 ACM International Symposium on Wearable Computers (ISWC'14)*, pp. 15–18 (2014).

[28] Isaacman, S., Becker, R., Cáceres, R., Martonosi, M., Rowland, J., Varshavsky, A. and Willinger, W.: Human mobility modeling at metropolitan scales, *Proceedings of the 10th International Conference on Mobile Systems, Applications and Services (MobiSys '12)*, pp. 239–252 (2012).

[29] 小島颯平，内山　彰，廣森聡仁，山口弘純，東野輝夫：俯瞰画像における被写体サイズ推定モデルに基づく群衆人数推定手法の提案，情報処理学会, マルチメディア, 分散, 協調とモバイル（DICOMO2015）シンポジウム論文集，pp. 157–165 (2015).

[30] Dalal, N. and Triggs, B.: Histograms of oriented gradients for human detection, *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR '05)*, pp. 886–893 (2005).

[31] Dollar, P., Tu, Z., Perona, P. and Belongie, S.: Integral Channel Features, *Proceedings of the British Machine Vision Conference*, pp. 91.1–91.11 (2009).

[32] Freund, Y. and Schapire, R. E.: A desicion-theoretic generalization of on-line learning and an application to boosting, *Proceedings of the Computational Learning Theory*, pp. 23–37 (1995).

[33] Schapire, R. E. and Singer, Y.: Improved boosting algorithms using confidence-rated predictions, *Machine learning*, Vol. 37, No. 3, pp. 297–336 (1999).

[34] Friedman, J., Hastie, T., Tibshirani, R. et al.: Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), *The annals of statistics*, Vol. 28, No. 2, pp. 337–407 (2000).

[35] Cheng, H.-Y., Jeng, B.-S., Tseng, P.-T. and Fan, K.-C.: Lane Detection With Moving Vehicles in the Traffic Scenes, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 4, pp. 571–582 (2006).

[36] Viola, P., Jones, M. J. and Snow, D.: Detecting pedestrians using patterns of motion and appearance, *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, pp. 734–741 (2003).

[37] Corvee, E. and Bremond, F.: Haar like and LBP based features for face, head and people detection in video sequences, *Proceedings of International Workshop on Behaviour Analysis and Video Understanding (ICVS 2011)*, pp. 10–20 (2011).