

## 道路を横断する歩行者認識技術の研究

### A Study on Automatic Recognition of Pedestrians

#### Crossing a Street

柳原 卓哉<sup>†</sup> 江口 一彦<sup>††</sup>

Takuya YANAGIHARA , Kazuhiko EGUCHI

**Abstract** Elder drivers are increasing in Japan in accordance with the progress of graying society. Risk of traffic accident caused by elder drivers may be higher than younger age drivers. Therefore a system and/or apparatus to safely support elder driver are highly expected. This paper discusses a method how to recognize pedestrians who are crossing a street. A template consist of the lower part of human body is proposed for template matching and optical flow method is applied in combination. Experimental results show possibility of increasing recognition rate and avoiding wrong recognition.

#### 1. はじめに

高齢化する現代社会において高齢者が運転を強いられる状況が増加している<sup>[9][10][11]</sup>。高齢者は年齢からくる認識力の低下、それに伴う判断力の低下が起これ交通事故を起こす危険性が高い。交通事故総合分析センターITARDAによると図1に示す通り、高齢者ドライバーは年々増加しておりそれに伴い死傷者の発生する事故も増え続けている。また高齢者が事故を起こす時間の特徴として、図2のように年齢を重ねるにつれ午前から午後の昼間に、交通事故が起き易いことが分かる。高齢になるほど生活スタイルは昼間に集中していることが原因であると推測できる。

本研究では高齢者の認知と判断を補助する目的で、高齢者運転支援技術の研究を行う。運転支援をするためには様々な物体を検出し適切な処理が必要となる。その中でも本研究では昼間の横断している歩行者の検出を行う。画像処理には車の運転席上部に web カメラを取り付け、撮影した動画を用いる。画像処理には、テンプレートマッチングとオプティカルフローを用いて人物領域の検出を行い高齢者運転支援システム開発への足がかりになるように研究を進めた。

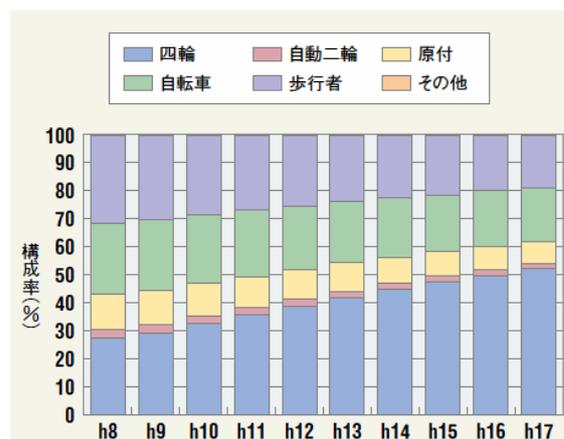


図1 高齢者ドライバー増加の推移<sup>[9]</sup>

<sup>†</sup>愛知工業大学 工学研究科電気電子工学専攻 (豊田市)

<sup>††</sup>愛知工業大学 工学部電気学科 (豊田市)

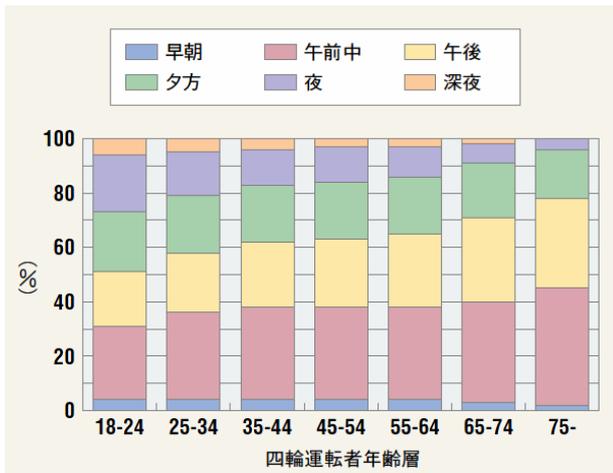


図2 年代別時間帯による事故発生率<sup>[9]</sup>

## 2. 横断歩行者の特徴

道路を横断している歩行者について、歩行者が歩く姿を横方向から見ると、

- ・秒速 0.8~1.1m/s で移動
- ・上半身はほぼ平行移動
- ・下半身は足を交互にしながら移動
- ・腕を振る

という特徴がある。従来は全身の特徴から人物領域の検出を行う事が多いが、本研究ではこのような特徴から下半身の動きと形に着目した。

## 3. テンプレートマッチング

テンプレートマッチング<sup>[1][2][3][4]</sup>とは認識しようとする対象をテンプレートとしてもち、対象画像の各部分との類似度を計算することにより、対象物の画像中の位置を求める手法のことを言う。画像から対象物を探すのに適した方法でその簡単な例を図3に示す。



図3 テンプレートマッチングの例

画像から抜き出したテンプレートならば確実に対象物を検出することが出来る。しかしテンプレートを概略の形状や色にして不特定多数の物を探索すると誤認識は増加する。

歩行者の認識にテンプレートを用いる場合、対象画像から切り取ったテンプレート画像を用いる場合、その1フレームでは適合するが他のケースでは類似度の計算結果が適合しにくい問題点がある。

類似度の計算方法には2乗残差法と相互相関法がある。

### 3.1 2乗残差法

2乗残差法は式(3.1)で表される。 $F(i, j)$ は対象の画像を表し、 $A(i, j)$ はテンプレートを表している。

$$E(x, y) = \sum_i \sum_j [F(i, j) - A(i - x, j - y)]^2 \quad (3.1)$$

テンプレート画像の位置を $(x, y)$ だけ移動した時の画像との整合性は $E(x, y)$ によって与えられる。単純に対象画像とテンプレート画像の輝度値を、各ピクセル同士で差を取りそれを2乗する計算である。計算結果の $E$ の値が最小となる位置を探す。計算速度は高速だが、画像間の明るさの変化には弱い。

### 3.2 相互相関法

相互相関法の相関係数 $\delta(x, y)$ は式(3.2)で表され、画素の位置差を平均移動量とする方法である。

$$\delta(x, y) = \frac{\sum_i \sum_j \{F(i, j) \cdot A(i - x, j - y)\}}{\sqrt{\sum_i \sum_j F^2(i, j)} \cdot \sqrt{\sum_i \sum_j A^2(i - x, j - y)}} \quad (3.2)$$

線形的な明るさの変化(画像の輝度及びコントラストの線形変化)とノイズに影響されないが、正規化処理のため処理時間は長くなる。相関係数は-1~1までの値で変化して、値が大きいくほど画像が類似していることを表す。この手法は $\delta(x, y)$ が最も相関係数が1に近くなる値を探す処理をする。

## 4. オプティカルフロー

時間的に連続した画像列を使い画像の速度場を求め、速度ベクトルにより物体の運動を検出する方法である。大きく分けて2種類の方法がある<sup>[1][2][5][6][7][8]</sup>。

### 4.1 勾配法

画像の時空間微分の拘束方程式による条件から速度ベクトルを出すオプティカルフローで、拘束式は画像中の点 $(x, y)$ の時間 $t$ における輝度を $E(x, y, t)$ とする。

$u(x, y), v(x, y)$ がオプティカルフローベクトルの $x$ 成分、

y 成分であるとした時、時刻  $t$  における輝度は時刻  $t+\delta$  の点  $(x+u\delta, y+v\delta)$  の輝度と等しい。よって、(3.3)式が小さい時間間隔  $\delta t$  に対して成り立つとみなす。

$$E(x+u\delta t, y+v\delta t, t+\delta t) = E(x, y, t) \quad (3.3)$$

もし輝度が  $x, y, t$  に関して滑らかに変化するならば(3.3)式の左辺をテイラー級数に展開でき(3.4)式となる。

$$\begin{aligned} \frac{dx}{dt} &= u \\ \frac{dy}{dt} &= v \end{aligned} \quad (3.4)$$

$u$  は画素  $(x, y)$  の速度ベクトルの  $x$  成分,  $v$  は画素  $(x, y)$  の速度ベクトルの  $y$  成分である。それぞれ  $x, y$  の時間微分を式(3.4)のようにすると未知数  $u, v$  に対する線形方程式となり式(3.5)で表される。

$$E_x u + E_y v + E_t = 0 \quad (3.5)$$

$E_x u, E_y v, E_t$  は  $x, y$  方向の空間微分値及び時間微分値で計算される。この式は一般的にオプティカルフローの拘束方程式と呼ばれる。

拘束方程式は一画素につき一つの拘束方程式しか得られないので、一意にフローベクトルを決定できない。よって2種類の付加条件を加えフロー推定を行う。

- 1) 「ある注目画素の近傍では動きは滑らかである」という条件を作り、注目画素近傍の拘束方程式からその最小2乗法の解を求めフローとする。
- 2) 「物体内の動きは滑らかに変化する」という条件を作り、発生するオプティカルフローの画像上での全変化量を最小にする拘束条件の繰り返し計算により、フローを滑らかに連結させる。

勾配法は対応点探索を行う必要がなく、計算が高速という長所がある。しかし輝度の急激な変化や雑音に弱いという短所がある。

#### 4.2 ブロックマッチング法

画像中の大きさの領域をテンプレートのブロックとし、次フレームのテンプレートのブロックと、次フレームの注目ブロックとの差分評価関数の値を算出する。その値が最小となる点を対応点とする手法である。類似度を評価する関数としては(3.6)式のような絶対値差分和がある。

$$e_B(x, y) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} |f_i(x+m, y+n) - f_{i-1}(x+m-i, y+n-j)| \quad (3.6)$$

ブロックマッチング法を用いた手法は輝度値の急激な変化や雑音に強いという長所があるが、全探索を行うため計算時間が膨大となり処理速度が遅くなる。また縮小や回転といった動きにも弱いと言う短所がある。

#### 4.3 ピラミッド構造を利用した特徴点に対する勾配法

勾配法を利用したオプティカルフローで、画像のピラミッド構造を使う方法である。図4に構造の概念を示す。原画像を順次縮小してピラミッド形状の階層構造を作成すると、画像上の動きはそれに応じて小さくなる。これを利用して最もサイズの小さい画像の特徴点同士をオプティカルフローで推定し、その結果を次の階層の画像間での初期値とする。この処理を原画像のサイズまで繰り返し処理するので、フローの急激な変化に対する信頼性が向上する。また各々特徴点に対して計算を行う事により無駄を省き処理速度も速くなる。

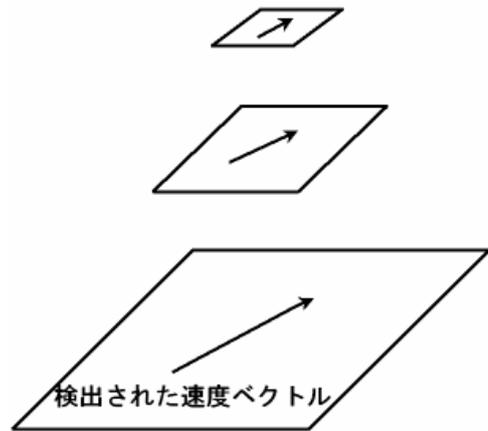


図4 ピラミッド構造

### 5. 実験結果

#### 5.1 横断歩行者の認識に最適なテンプレートの検証

テンプレートマッチングにおいて横断する人間にはどのようなテンプレート画像が適するか検証する。テンプレートマッチングには相互相関法を用いる。試したパターンは、全身・上半身・下半身の3通りで動画から無作為に10フレームの画像を抜き出し、それぞれ歩行者検出を試みた。検証に使用したテンプレート画像は図5の3枚の画像を用いた。



図5 テンプレート画像(全身・上半身・下半身)



図 6 3 パターンのマッチングに成功

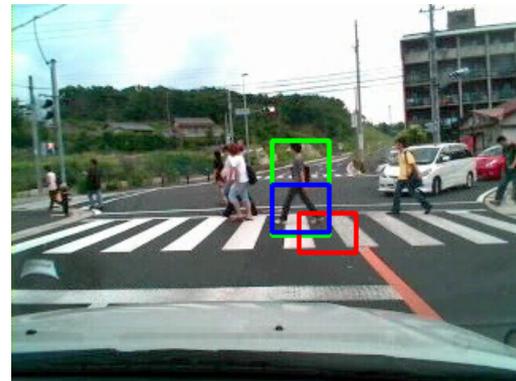


図 7 全身下半身のマッチングに成功

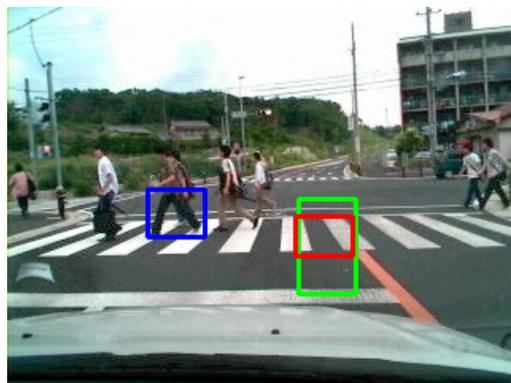


図 8 下半身のみマッチングに成功

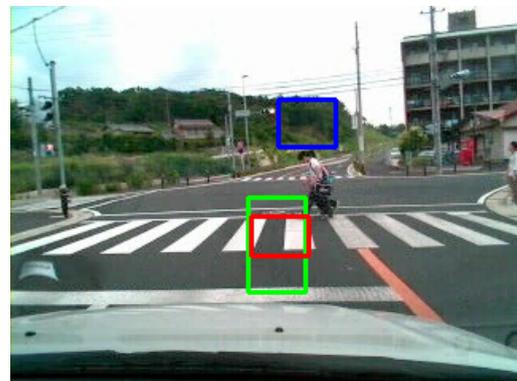


図 9 全てのマッチングに失敗

代表的な 4 パターンの検証結果を図 6~9 に示す。マッチングした箇所が緑枠の時は全身、赤枠は上半身、青枠は下半身で表す。

図 8 が 10 回試行した結果、6 回あった下半身のみマッチングしている例である。図 9 は背景にマッチングして、全てのマッチングに失敗した例である。

結果をまとめた物を表 1 に示す。表から、下半身が一番マッチングしやすいことを確認した。これは横断している歩行者の足の形状が特徴的な形をしている事から、マッチングし易いと考えられる。全身のテンプレートを使うとマッチングしなくてはならない領域が広いので、認識を失敗しやすい。また Case5 のように上半身のみマッチングしなかったケースに於いても、足の形状の影響が出ているため、全身では認識可能でも、上半身のみでは認識出来ないと考えられる。全てのマッチングに失敗した Case6 や Case9 は認識対象の歩行者が重る、足が揃うなどの原因で足の特徴が消えているのが原因と考えられる。

表 1 テンプレートマッチングの結果

	上半身	下半身	全身
Case1	○	○	○
Case2	×	○	×
Case3	×	○	×
Case4	×	○	×
Case5	×	○	○
Case6	×	×	×
Case7	×	○	×
Case8	×	○	×
Case9	×	×	×
Case10	×	○	×
合計	1	8	2

## 5.2 テンプレート画像と処理速度の関係

テンプレート画像の大きさ 36×30, 処理動画 320×240 で処理速度の計測を行った。その結果を図 10 のグラフに示す。

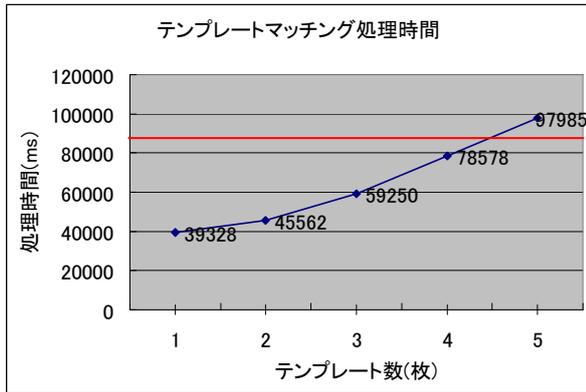


図 10 処理速度とテンプレート枚数の関係

グラフから読み取れる通り、テンプレートの枚数が増えると処理時間が増加する。この動画の時間が 83000ms で図中の赤線の部分である。リアルタイムな処理速度を目指すと、テンプレート画像の枚数は 4 枚位が限界だろうと思われる。また他の処理を追加する事を考慮すると 2, 3 枚が実用的な限度と考えられる。

### 5.3 オプティカルフローの手法の違いにおける結果と処理速度

4章で検討した勾配法、ブロックマッチング法、ピラミッド構造を利用した特徴点に対する勾配法による、3種類の手法で処理した結果を、図 10、図 11、図 12 に示す。赤い線は算出された速度ベクトルである。勾配法は細かな雑音などに敏感に反応している。処理速度は速いが、図中に見られる本来起こりえない、長いフローベクトルが得られるなどフローの精度が悪い。

ブロックマッチング法は勾配法より雑音などに反応しないので、フローの精度が改善されている。しかし動画にこの処理を適用すると処理速度が非常に遅くなる。

ピラミッド構造を利用した特徴点に対する勾配法を用いると、特に大きく動く物体にのみ反応し、細かな雑音は無視できる。また特徴点を割り出してから速度ベクトルの計算をするので、処理速度も高速である。



図 10 勾配法



図 11 ブロックマッチング法



図 12 ピラミッド構造を利用した特徴点に対する勾配法

3種類の手法の 1 フレーム当たり処理速度を比較した物を表 2 に示す。

表 2 解像度別の処理速度の比較

解像度	勾配法	ブロック マッチング	ピラミッド 勾配
320×240	375ms	1281ms	266ms
640×480	719ms	4968ms	312ms

### 5.4 人物領域抽出とオプティカルフローの効果の検証

テンプレートマッチングでは、相関係数を閾値にしてマッチングを行う。

相関係数を低く設定・・・背景などの誤認識が増加

相関係数を高く設定・・・歩行者を認識しない

以上のような特徴がある。これより本実験ではまず相関係数を低めに設定した。誤認識に対してはオプティカルフローを適用することによりその軽減を試みた。処理の流れを図 13 に示し、画像に対する表示方法を図 14 に示す。

本実験ではテンプレートマッチングに相互相関法を、テンプレート画像は下半身のテンプレート画像を用いた。誤

認識除去をするオプティカルフローは、ピラミッド構造を利用した特徴点に対する勾配法を用いた。動画 1000 フレーム間のオプティカルフロー適用前、適用後の誤認識の数を計測し、それぞれ相関係数 (0.80・0.82・0.85) と変えて実験した。

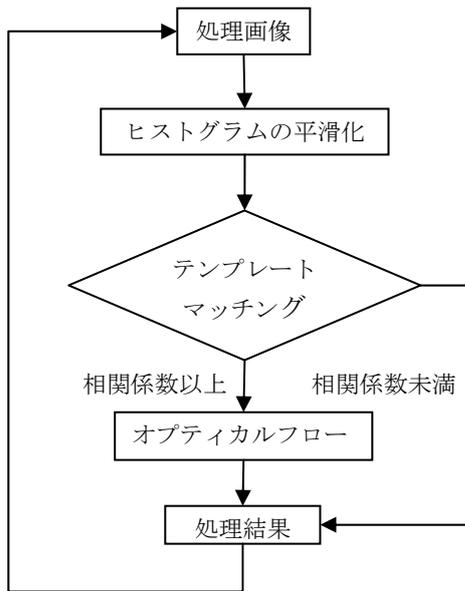


図 13 処理の流れ



図 15 人物領域抽出の成功例



図 16 背景に対する誤認識を除去した例

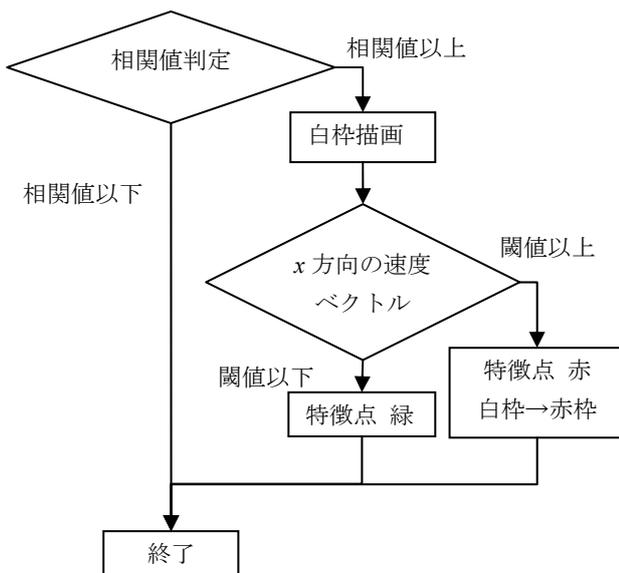


図 14 表示方法



図 17 人物領域抽出の失敗例

図 15 は歩行者の認識に成功した例である。この場合下半身の特徴をマッチングしてから  $x$  方向に一定以上動きがあるので、歩行者の判定に成功している。図 16 のケースはテンプレートマッチングでは認識したが  $x$  方向に一定以上動きがないため、オプティカルフローによって「歩行者ではない」と判断した例である。頻繁に雑音の影響でこのような背景に対する誤認識が増える。図 17 は歩行者以外の物体を、歩行者として認識した失敗例である。テン

プレートマッチングの誤認識が背景以外の動物体にマッチングした場合、誤認識が除去出来ていない。ただし、この現象は  $x$  方向のベクトル変化量の閾値を歩行者の歩く速度に合わせて変更する事により改善が可能であると考えられる。

誤認識の改善結果を表 3 にまとめる。1000 フレーム間の動画を処理し、誤認識の減少量を調べた。表から適用前から適用後では 70~80%減少していることが分かる。背景などの静止物に対する誤認識は除去し易いが、車などの動物体への誤認識を減少させることは困難であることが分かった。

表 3 オプティカルフローによる誤認識の改善

相関係数	適用前	適用後
0.85	3	1
0.82	8	3
0.80	10	2

## 6. 結言

本研究では画像処理による歩行者認識の研究を行った。横断歩行者に最適なテンプレートの検証では、下半身のテンプレート画像が最もマッチング結果が良好にであることを確認した。テンプレートの枚数による処理速度の検証では、枚数が多ければ多いほど処理速度は遅くなり、リアルタイムで処理するには約 4 枚が限界ということが分かった。各種オプティカルフローによるフロー結果と、処理速度の比較では、ピラミッド構造を利用した特徴点に対する勾配法が、フローの精度が高く処理速度も速い事が分かった。動画中でオプティカルフローを適用することにより誤認識を 70~80%減少させる事ができた。

## 7. 今後の課題

- ・今回は昼の横断報道と限定されたケースでのみ実験であったが、夜間や悪天候など様々なケースでの検証を行う
- ・テンプレートマッチングにおけるテンプレートの種類を増やせば認識率は向上するが、リアルタイムな画像処理は困難になる。テンプレートマッチングの探索領域を絞り込むなどの処理速度向上策が必要である。
- ・今回は単眼の web カメラを用いたが、ステレオカメラを用いて対象物との距離を推定し、認識率を向上させる。

## 8. 参考文献

- [1] 双目出来太他： ロボットの目をつくる，トランジスタ技術増刊，CQ 出版社，2006，5
- [2] 奈良先端科学技術大学院大学 OpenCV プログラミングブック製作チーム： OpenCV プログラミングブック，2007，9
- [3] 武田大祐，高鹿陽介：“OpenCV を利用した動画画像処理プログラミング前編”，interface2006 年 11 月号 CQ 出版社，2006，11
- [4] 武田大祐，高鹿陽介：“OpenCV を利用した動画画像処理プログラミング中編”，interface2006 年 12 月号 CQ 出版社，2006，12
- [5] 武田大祐，高鹿陽介：“OpenCV を利用した動画画像処理プログラミング後編”，interface2007 年 1 月号 CQ 出版社，2007，1
- [6] 今村弘樹：“フロー外挿予測を用いた遮蔽，発生領域におけるオプティカルフロー推定”，北陸先端科学技術大学修士論文
- [7] 渡辺睦：“オプティカルフローによる動物体の認識アルゴリズムの研究”，鹿児島大学研究報告
- [8] 行動知能システム学研究室ホームページ，<http://www.aisl.ics.tut.ac.jp/index.html>
- [9] (財) 交通事故総合分析センター，ITARDA <http://www.itarda.or.jp/>
- [10] HONDA (本田技研株式会社) ホームページ <http://www.honda.co.jp/>
- [11] jieee spectrum online <http://www.spectrum.ieee.org/>

(受理 平成 20 年 3 月 19 日)