# 歩行者の不注意行動認識 - 歩きスマホ検出 -

## Recognize Careless Behavior of Walker –Detect Texting While walking-

皆本　光[†1]，佐野 睦夫[†2]

Hikaru MINAMOTO[†1]，　Mutsuo SANO[†2]

†1, †2 大阪工業大学情報科学部　　†1, †2Osaka Institute of Technology

E-mail:　†1 e1c11086@st.oit.ac.jp,　†2 sano@is.oit.ac.jp

### 1. はじめに

近年では福祉介護へのサポートになるロボットの研究開発が盛んであり，ロボットに対して高度な処理が求められるようになった．また，現在盲導犬の実働数は 1,010 頭（2015 年 2 月 1 日時点）であり，全国で盲導犬を希望者数約 1 万人と比べて明らかに不足している[1]. 盲導犬が不足している原因には盲導犬の育成費用や育成期間，実働期間の問題が挙げられる．そのため，盲導犬の機能を工学的に実現し，視覚障がい者の支援を行う盲導犬ロボットの研究が行われている．このような背景から我々も盲導犬ロボットの開発を行っている．

### 2. 盲導犬ロボットの不注意行動認識

我々が開発している盲導犬ロボットには安全な歩行支援とコミュニケーション支援という 2 つの目的がある．この 2 つの機能については以下の表 1 に示す．

表 1　盲導犬ロボットの機能

| 機能 | 項目 | 事例 |
|---|---|---|
| 安全な歩行支援 | 歩行者の不注意行動 | スマートフォンを見ながら歩く |
| | | よそ見をしながら歩く |
| | 自転車の不注意行動 | よそ見をしながら走行する |
| | 路面状況 | 段差 |
| | | 落ち葉 |
| | | 水たまり |
| | | 積雪 |
| | 歩行環境の変化 | 人ごみ |
| | | 駐車車両 |
| | | 交通規制 |
| コミュニケーション支援 | 知人の行動・状態確認 | 人物特定 |
| | | 服装の色 |
| | | ジェスチャーを行っている |
| | | 表情の変化 |
| | | こちらに向かっている |
| | 景色の変化 | イベントが開かれている |
| | | 植物が変わっている |
| | | 建造物が立っている |

今回は安全な歩行支援における不注意行動認識として，近年問題視されている歩きながらスマートフォンの画面を見るスマホ歩きの認識を行う．このスマホ歩きを行っている人は注意散漫になり，他の歩行者と衝突してしまう恐れがあるため視覚障がい者にとっては特に危険である．

スマホ歩きを認識するためには，スマホ歩きの行動特性を把握し学習する必要がある．以下にスマホ歩きの行動特性として考えられるものを列挙する．

① 顔，視線がスマートフォンの画面に向いている
② スマートフォンを持っている腕を胸の前付近まで挙げている
③ 時折前方を確認するために顔または視線を前方に向ける
④ 持ち上げている腕には必ずスマートフォン（板状の物体）を持っている

以上のようなものが考えられる．これらの行動特性の特徴量を計算，識別することでスマホ歩きを認識することが可能となる．
今回の認識手法では先に列挙した行動特性のうち，②に着目して認識を行おうと考えた．

### 3. スマホ歩き認識に用いる技術

#### 3.1 Hog 特徴量

Hog（Histogram of Oriented Gradients）[2] 特徴量とは画像特徴量のひとつであり 2006 年にフランスの研究所 INRIA で提案されたものである．Hog はある局所領域における輝度の勾配方向をヒストグラム化した特徴量である. SIFT と呼ばれる特徴量と類似した特徴量であるが，相違点として SIFT は特徴点に対して特徴量を記述するのに対し，Hog ではある一定領域に対して特徴量の記述を行う. 以下の図 1 に Hog 特徴量を疑似的に可視化したものを示す.

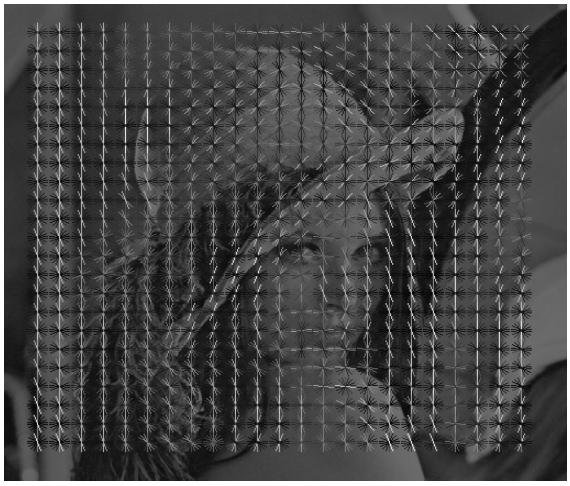

図 1　Hog 特徴量のイメージ

図 1 で表されるように，この Hog 特徴量を用いることで対象の大まかな形状や姿勢を表現することが可能となる.よってスマホ歩きの姿勢の Hog 特徴量を学習し判定を行うことができる.

### 3.2 SVM 識別器

今回 Hog 特徴量の学習に用いたのはサポートベクターマシン(以下，SVM と表記)[3][4][5]は 2 クラスの分類を行う機械学習の一種である．学習データの中でサポートベクトルと呼ばれるクラス境界近傍に位置するデータと識別面との距離が最大となるように分離超平面を構築しクラス分類を行っている．この各クラスにおけるサポートベクトルと決定境界との距離をマージンと呼び，この距離を最大にすることをマージン最大化と呼ぶ．例として SVM 識別器のイメージを図 2 に示す.

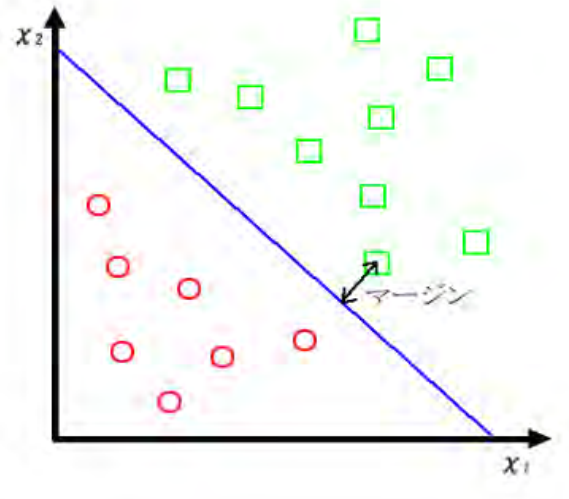


図 2　SVM 識別器のイメージ

図 2 のように正例領域と負例領域の境界を適切に分割する（マージンを最大化する）ことによって SVM 識別器を作成することができる．完成した SVM 識別器に未知データを入力した際，未知データがどの領域に入るかを識別することで正負の判定が可能となる.

今回はスマホ歩きの姿勢を正例，それ以外の姿勢を負例として学習を行った．学習に用いた画像枚数は正例，負例ともに 1700 枚の計 3400 枚である．以下の図 3 に正例，図 4 に負例のサンプルを示す.

図 3　.正例画像のサンプル

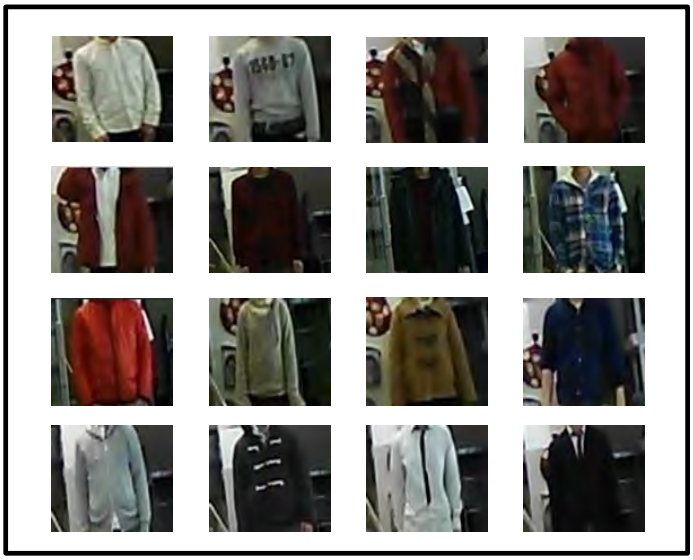

図 4　負例画像のサンプル

### 3.3 Haar-Like 特徴

Haar-Like 特徴は 2 節で挙げたスマホ歩きの行動特性の①について，簡単な判定を行うために正面顔を検出する際に利用した．顔検出に使われた特徴量の Haar-like 特徴量には，Paul Viola, Michael Jones により開発[6]され，Rainer Lienhart, Jochen Maydt によって改良[7]された Haar-like 特徴[8]を利用した Adaboost アルゴリズムを用いた．AdaBoost は Yoav Freund と Robert Schapire によって考案された機械学習アルゴリズムで，単純な特徴の利用による高速性と，環境変化に対するロバスト性を比較的よく兼ね備えた検出手法である．

Haar-like 特徴は図 2.12 のような 14 種類のパターンを水平方向，垂直方向にスケーリングした特徴を利用する．図 2.13 に示すように，探索窓の任意の位置に Haar-like 特徴を置き，白い領域を正，黒い領域を負として両領域の輝度の合計値を計算し，これを特徴量とする．

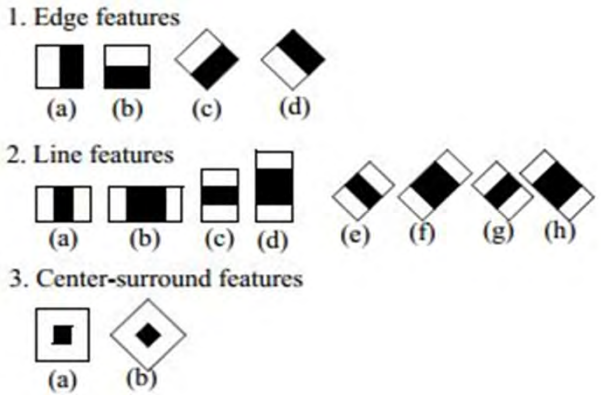


図 5 Haar-like 特徴

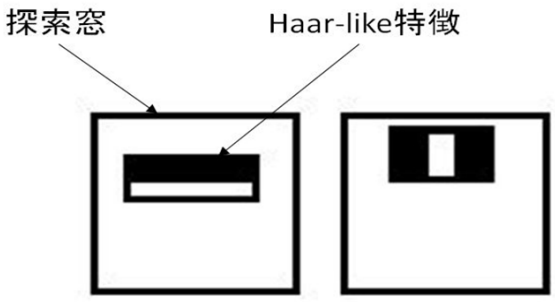


図 6 探索窓

この Haar-like 特徴を用い，カスケード分類器に通すことにより，顔画像が識別される．AdaBoost アルゴリズムは，単純な特徴を検出する弱い検出力の分類器をたくさん繋ぎ合わせることで，最終的には検出力の強い分類器を作るという考え方である．拒否率が低い分類器により，顔の検出漏れが少なくなるよう設計されている．図 2.14 に検出処理のカスケード構造のイメージを示す．識別機では，T(True:顔である)か，F(False:顔でない)で判断され，T と判断された候補だけが次の処理に進むため，効率がよく，計算時間が遅くならないようになっている．

### 3.4 パーティクルフィルタ

パーティクルフィルタとは画像上の物体を追跡する手法のひとつである．今回パーティクルフィルタは手の停滞や位置を把握するために用いた．以下にパーティクルフィルタについて説明を行う．パーティクルフィルタでは，時刻$t$ における追跡対象の状態ベクトルを$x_t$，観測ベクトルを$y_t$とし，観測値の集合を

$$Y_t = \{y_1, \ \ldots, \ y_t\}$$

としたとき，追跡問題は$x_t$ の事後確率分布 $p(x_t|Y_t)$として推定できる．事後確率$p(x_t|Y_t)$は，ベイズの定理により式 1 のように事前分布と尤度の積に置き換えられる．

$$p(x_t|Y_t) = k_t p(y_t|x_t) p(x_t|Y_{t-1}) \qquad (1$$

ただし，$k_t$は正規化定数である．また$p(y_t|x_t)$は，ある状態$x_t$のときに，観測値$y_t$を得る確率(尤度)である．$p(x_t|Y_{t-1})$は，時刻$t$ における$x_t$の事前確率であり，$x_t$ のマルコフ性により，式 2 のように与えられる．

$$p(x_t|Y_{t-1}) = \int p(y_t|x_t) p(x_t|Y_{t-1}) d \qquad (2$$

パーティクルフィルタのポイントは，2 種類のサンプル集合$s_{t|t-1}$, $s_{t|t}$をそれぞれ事前分布 $p(x_t|Y_{t-1})$，事後分布$p(x_t|Y_t)$に従って生成することである．これらの分布は式(1，(2 に従って推定される．パーティクルフィルタでは，これらの式を，次のような手順でサンプルに適用しながら，逐次的にサンプル集合を生成する．

(1) 初期化

$i = 1, \ \ldots, \ N$について$s_{0|0}^{(i)} \sim p_0(x)$を生成する．

ただし$p_0(x)$は，あらかじめ与えた初期分布である．$t: = 1$として以下の手順を実行する．

（2）予測

各サンプルについて，時刻$t$における予測サンプル

$$s_{t|t-1}^{(i)} \sim p\left(x_t | x_{t-1} = s_{t-1|t-1}^{(i)}\right)$$

を次の手順で生成する．

(a) $i = 1, \ \ldots, \ N$について，$l$ 次元の乱数としてシステムノイズ$v^{(i)} \sim Q(v)$ を生成する．

ただし，$q(v)$ はあらかじめ設定されたシステムノイズ$v$ の分布である．

(b) 時刻$t-1$ の各サンプル$s_{t-1|t-1}^{(i)}$ を遷移させて予測サンプルを生成する．

$$s_{t|t-1}^{(i)} = f_t\left(s_{t-1|t-1}^{(i)}, \ v_t^{(i)}\right) \qquad (3$$

(3) 尤度設定

各サンプル$s_{t|t-1}^{(i)}$ について，重み$\pi_t^{(i)}$ を推定する．

$$\pi_t^{(i)} = \frac{p\left(y_t | x_t = s_{t|t-1}^{(i)}\right)}{\sum_{i=1}^{N} p\left(y_t | x_t = s_{t|t-1}^{(i)}\right)} \qquad (4$$

$p\left(y_t | x_t = s_{t|t-1}^{(i)}\right)$ は，状態$x_t$ が$s_{t|t-1}^{(i)}$ であったときに，観測$y_t$ を得る確率(尤度)である．

(4) フィルタ

フィルタ：$s_{t|t-1} = s_{t|t-1}^{(1)}, \ \ldots, \ s_{t|t-1}^{(N)}$ からそれぞれ$s_{t|t-1}^{(i)}$ を重み$\phi_t^{(i)}$ に比例する割合で$N$個復元抽出し，$s_{t\,t} = s_{t|t}^{(1)}, \ \ldots, \ s_{t|t}^{(N)}$ とする．

$t := t+1$ として，(2)の予測へ戻り繰り返す．

図 7 にパーティクルフィルタのアルゴリズムを示す．

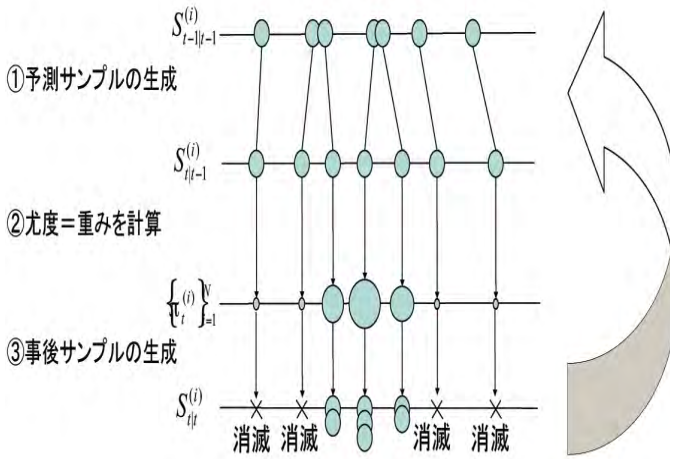


図 7 パーティクルフィルタのアルゴリズム

以下にパーティクルフィルタを適用し手を追跡した例を以下の図 8 に示す．

図 8　パーティクルフィルタによる手の追跡

今回手を追跡するために，手は肌色であることを利用し，まず胸元領域から肌色（OpenCV の HSV 表色系にて 6≦H≦18，20≦S）領域を赤色（OpenCV の HSV 表色系にて H=0，S=255，V=255）として抽出する．次に各パーティクルの位置情報から，そのピクセルの赤色らしさを尤度として用いる．具体的には HSV 色空間における赤色からのユークリッド距離 d に対して，0 を平均，分散 σ の正規分布を尤度関数 L(d)（式 5）として設定した．

$$L(d) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{d^2}{2\sigma^2}\right)} \qquad (5$$

今回パーティクルフィルタを作成した判定の流れは以下である．

① パーティクルの重心座標をフレーム毎に求め，フレーム間の平均座標の移動距離を求める．
② 移動距離が一定以下であれば停滞フレーム数のカウントを増加させ，移動距離が一定以上であれば停滞フレーム数のカウントをリセットする．
③ この処理をフレーム毎に行い，停滞フレーム数のカウントが一定以上であればスマホ歩きと認識する

という流れである．このフローで用いた重心座標の求め方は，重心座標の x 成分を Gx，y 成分を Gy，各パーティクルの x 軸成分を nPx，y 軸成分を nPy，存在するパーティクルの総数を E（1≦n≦E≦1000）とした重心座標の算出方法を式 6 に示す．

$$Gx = \frac{1Px + 2Px + \cdots + nPx + \cdots + EPx}{E} \quad (6$$

$$Gy = \frac{1Py + 2Py + \cdots + nPy + \cdots + EPx}{E}$$

またパーティクルの停滞時間だけではなく，停滞する位置についての判定も作成した．正と判定される領域は抽出した胸元領域について胸部から腹部にあたる部分である．以下の図 9 に示される赤色の領域が正と判定される範囲の一例である．

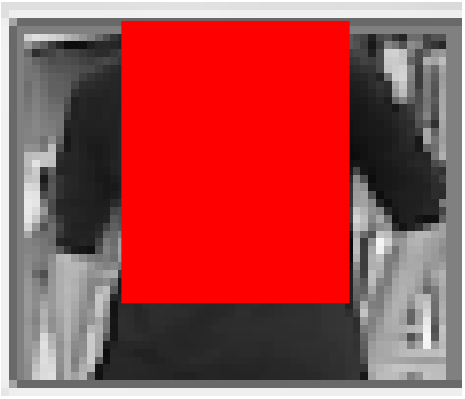

図 9　正判定領域の例

## 4. スマホ歩き認識手法

今回提案する手法は 3 節で述べた Hog 特徴量の識別，パーティクルの停滞時間の判定，顔検出の判定を組み合わせたものである．

これらの判定を組み合わせた認識手法をまとめると以下の 9 つの手法となる．

- 手法 A：パーティクルの停滞時間のみを判定する
- 手法 B：パーティクルの停滞時間と重心座標の位置をそれぞれ判定する
- 手法 C：Hog 特徴量と顔検出をそれぞれ判定する
- 手法 D：Hog 特徴量とパーティクルの停滞時間をそれぞれ判定する
- 手法 E：Hog 特徴量とパーティクルの停滞時間，重心座標の位置をそれぞれ判定する
- 手法 F：顔検出とパーティクルの停滞時間それぞれ判定する
- 手法 G：顔検出とパーティクルの停滞時間，重心座標の位置それぞれ判定する
- 手法 H：Hog 特徴量と顔検出，パーティクルの停滞時間，それぞれ判定する
- 手法 I：Hog 特徴量と顔検出，パーティクルの停滞時間と重心座標の位置，それぞれ判定する

以下の図 10 に実際の認識フロー例を示す．

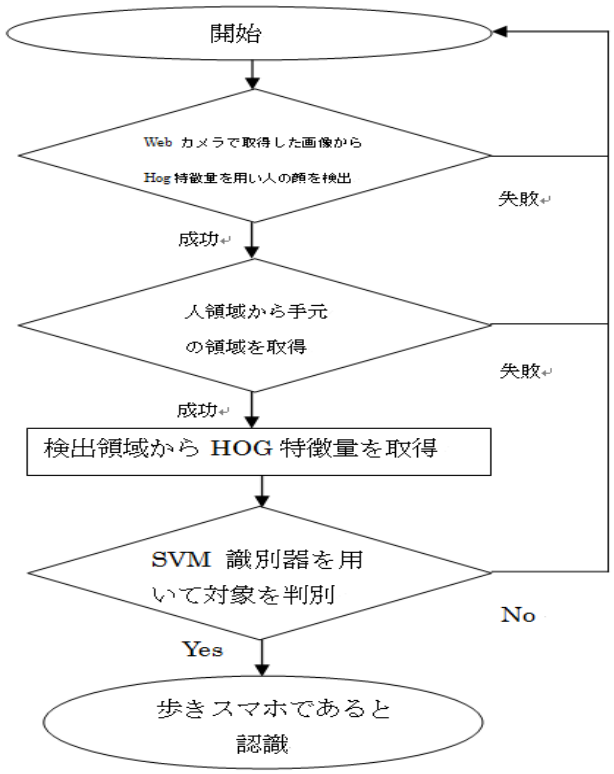


図 10　認識フロー例

## 5. 実験

### 5.1 SVM 識別器の性能評価

Web カメラを使用してスマートフォンを見ながらユーザーの前方から歩いてくる歩行者の認識率を検証する．一般に画像認識の実験では，性能評価は 2 クラス分類の場合，適合率と再現率によって行う．したがって，評価方法の指標は適合率と再現率を用いる．そのため提案する認識法を未知の画像に適応し適合率と再現率を用いて性能を検証した．実験に使用した被験者毎のテスト画像枚数は正例，負例ともに 80 枚の計 160 枚である．以下の図 11 に今回の実験の被験者の胸部付近の画像を示す．


図 11　学習性能評価に用いた被験者毎の画像

### 5.2 認識手法毎の比較

4 節で挙げた認識手法について，どの手法がスマホ歩き認識に適しているかを調べる目的で実験を行う．実験方法は 5.1 節と同様に適合率と再現率をそれぞれの手法で算出する．ただしこちらの実験ではパーティクルの停滞時間を判定に用いているため画像では認識を行えないため，画像を動画に置き換えて算出を行う．手法 C では停滞時間によって判定を行わないため，パーティクルフィルタを用いた手法と同様に 3 秒間続いた場合スマホ歩き認識成功とした．

### 5.2 使用機器

実験に使用した PC は ThinkPad T430s であり，Web カメラは Qcam Orbit AF である．これらの機器の詳細は以下の表 2,表 3 に示す．

表 2　PC の仕様詳細

| | |
|---|---|
| PC | ThinkPad T430s |
| OS | Windows 7 Professional 64bit |
| プロセッサー | インテル® Core™ i7-3520M プロセッサー |
| プロセッサー動作周波数 | 2.90GHz |
| RAM | 4GB(4GBx1)(PC3-12800 DDR3 SDRAM) / 16GB |
| HDD | 320GB(7200rpm) |

表 3　Web カメラの仕様詳細

| | |
|---|---|
| Webカメラ | Qcam Orbit AF |
| 動画フォーマット | HD |
| 画素数 | 200万画素 |
| 解像度 | 1600×1200 |
| フレームレート | 最大30fps |
| 水平視野角 | 189° |
| 垂直視野角 | 102° |
| フォーカス | オートフォーカス可 |

## 6. 結果・考察

### 6.1 学習性能評価結果・考察

学習性能評価結果は以下の表 4 に示し，グラフ化したものを図 12，13 に示す．

表 4　学習性能評価結果

| 被験者 | 画像数 | 適合率 | 再現率 |
|---|---|---|---|
| A | 232 | 97.47% | 50.00% |
| B | 153 | 88.06% | 74.68% |
| C | 113 | 58.33% | 79.55% |
| D | 168 | 65.07% | 95.96% |
| E | 198 | 61.97% | 55.70% |
| F | 113 | 93.88% | 85.19% |
| G | 163 | 100.00% | 29.81% |
| H | 188 | 56.38% | 77.06% |

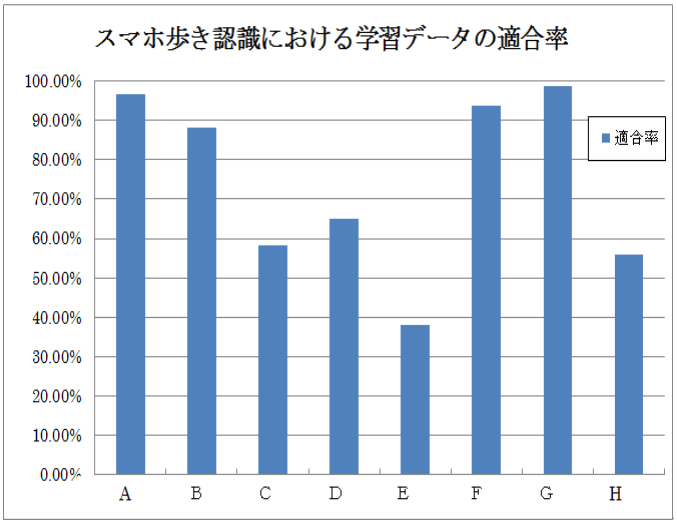


図 12　学習データの適合率

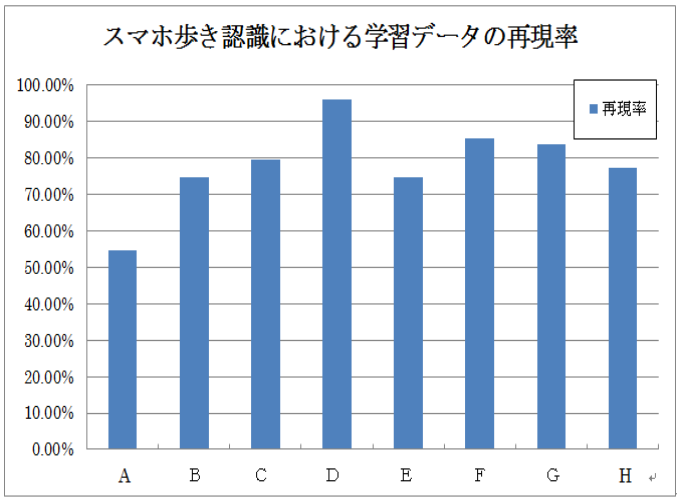


図 13　学習データの再現率

学習結果であるが，表 4 に示すとおり適合率・再現率ともに被験者によって差がみられる結果となった．特に低い適合率となった被験者 E について，以下の図 14 に示すような胸ポケットに物を入れる動作を行っていた．この動きはスマホ歩きではないが，腕が胸部前方付近まで上がっている．この姿勢は学習データに非常に似ているため，誤認識してしまったと考えられる．また表 4 及び図 13 から再現率に関しては，被験者 A が特に低い結果となっている．原因は人検出が被験者 A に対してあまり成功していなかったことが挙げられる．この問題に対しては人検出の精度を上げることによって改善が図れると考えられる．

図 14　誤認識した被験者 E の例

### 6.2　手法毎の比較結果及び考察

手法毎の比較結果を以下の表 5 に示す．

表 5　手法毎の比較結果

| 手法 | 動画数 | 適合率 | 再現率 |
|---|---|---|---|
| A:パーティクル時間 | 30 | 36% | 90% |
| B:パーティクル時間･位置 | | 60% | 90% |
| C:Hog＋顔検出 | | 53% | 100% |
| D:Hog＋パーティクル時間 | | 60% | 90% |
| E:Hog＋パーティクル時間･位置 | | 90% | 90% |
| F:顔検出＋パーティクル時間 | | 56% | 90% |
| G:顔検出＋パーティクル時間･位置 | | 62% | 80% |
| H:Hog＋顔検出＋パーティクル時間 | | 89% | 80% |
| I:Hog＋顔検出＋パーティクル時間･位置 | | 100% | 80% |

表 5 に示すとおり，適合率は手法 I，再現率は手法 C が一番高いという結果となった．手法 C は適合率 53%とほぼ半分が誤認識でありスマホ歩き認識が出来ているとは言えない．一方手法 I はデータが少ないとはいえ誤認識がなく再現率も 80%と悪くない精度となった．また手法 E も適合率・再現率ともに 90%と高い精度になった．さらに手法 E は手法 I と比較して処理が一つ少ないため CPU に対する負荷が小さい．よって今回の提案手法のなかでは手法 E がスマホ歩き認識に適していると考えた．

今回の実験では適合率・再現率ともに良い結果がみられたが，今回の実験データ量ではたまたま良い結果が出たということは否定できない．また今回多くの手法でスマホ歩きを認識すべき入力に対して，正しく認識出来なかった以下の図 15

に示すようなデータであった．


図 15　パーティクル収束失敗例

図 15 のようにパーティクルが手に収束することが全くなかった．原因は肌色が上手く検出されなかったためである．この問題の対策として，肌色ではなく抽出領域から手の質感等を求めて手を検出することが出来れば服の色にも影響されず，パーティクルの収束が上手くいくと考えられる．

## 7. まとめ

本稿では我々が開発している盲導犬ロボットの機能である不注意行動認識において，現在問題視されている歩きスマホの認識を行った．今回提案したスマホ歩きの認識手法のなかでは，パーティクルフィルタと Hog 特徴量を組み合わせた手法が最も適していることが分かった．今回のように不注意行動の行動特性を挙げ，認識をひとつひとつ行うことで注意散漫である人物との衝突を防ぐことができると考えられる．しかし認識する不注意行動を安易に増やすとコンピュータの資源や動作速度にも影響が出ると考えられた．また行動特性を判定する処理を不注意行動毎に作成することは，実際に行うと良い方法ではないと感じた．

今後の課題として，不注意行動とは人物がどのような状態であるかを定義し学習させる必要があると考えられた．今回の目的が注意散漫な人物との衝突を避けるためであるため，人物の視線を正確に認識出来ればこの問題について大きな改善が図れると考えられる．

## 文　献

[1] 厚生労働省 http://www.mhlw.go.jp/topics/bukyoku/syakaihojyoken/html/b04.html

[2] N. Dalai, and B. Triggs, “Histograms of oriented gradients for human detection”, Computer Vision and Pattern Recognition Vol.1, pp.886-893, 2005.

[3] C. Cortes and V. Vapnik, “Support-Vector Networks”, Machine Learning, Vol.20, pp.273-297, 1995

[4] 小野田崇著,”サポートベクターマシン”, オーム社, pp.1-78, 2007

[5] 渡辺澄夫, 萩原克幸, 赤穂昭太郎, 本村陽一, 福永健次, 岡田真人, 青柳美輝 著,”学習システムの理論と実現”, 森北出版, pp.46-73, 2005

[6] P. Viola, M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features“, In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, USA, pp.1-9, 2001

[7] Rainer Lienhart and Jochen Maydt, “An Extended Set of Haar-like Features for Rapid Object Detection”, IEEE ICIP 2002, Vol.1, pp. 900-903, 2002

[8] Y. Freund and R. E. Schapire, “A detection-theoretic generalization of on-line learning and an application to boosting”, Journal of Computer and System Sciences, Vol.55, No.1, pp.119-139, 1997