パーティクルフィルタとニューラルネットワークを用いた 視線方向推定に関する一構成法

柴田 論^{a,*} 呉 志強^a 山本 智規^a

A Method to Estimate Eye Gaze Direction using Particle Filter and Neural Network

Satoru Shibata^{a,*}, Zhiqiang Wu^{a,*}, Tomonori Yamamoto^a

(Received December 8, 2016; revised February 15, 2017; accepted February 24, 2017)

Abstract

In conventional eye gaze input system under natural light, since eye gaze direction is estimated by extracting center of mass of black eye (iris and pupil), the accuracy of the estimation is not satisfactory because part of black eye is hidden behind eyelid. In this study, we propose a practical method to calculate eye gaze direction using a particle filter and a neural network under the condition that a computer and a USB camera are used to cut costs and easy to set up. In the proposed method, outer corner and inner corner of eye are detected, and position of black eye is projected by the particle filter taking the effect of eyelid into consideration. In addition, the neural network is used to estimate eye gaze direction from positions of outer corner and inner corner of eye, and black eye. To examine the effectiveness of the proposed method, experiments in which human subjects follow a moving target on the screen by eye movement are conducted, and accuracies of the estimation are compared between the proposed method and the conventional method. As the results, it is confirmed that the accuracy of the proposed method is superior to that of the conventional method, which shows the effectiveness of the proposed method.

キーワード：視線方向推定，パーティクルフィルタ，ニューラルネットワーク，視線入力インタフェース

Keywords : estimation of eye gaze direction, particle filter, neural network, interface of eye gaze input.

1. はじめに

近年，ALS（筋萎縮性側索硬化症）をはじめとする重度肢体不自由者のコミュニケーションのために，比較的最後まで随意運動の残る眼球運動に着目し，人間の視線方向の変化を捉え，コンピュータへの文字入力，ソフトウェアの起動，実行などを行うことが可能な視線入力インタフェースが数多く開発されている⁽¹⁾。これまで，視線入力装置における視線方向の検出方法としては，EOG法，強膜反射法，角膜反射法などが報告されている⁽²⁻³⁾が，患者に対して非侵襲ではない点や，赤外線を眼球に長時間照射することに起因する疾病の危険性⁽⁴⁾，赤外線を用いた市販の視線入力装置は高価であるなどの問題点が指摘されている。

一方，自然光の下，上半身（顔）の撮影画像を画像処理し，目を検出して瞳孔を含む虹彩部分（以下，黒目と呼ぶ）の位置から視線推定を行うアプローチも報告されている。落合ら⁽⁵⁾，富本ら⁽⁶⁾は開眼画像と閉眼画像の差を取ることで目の領域を特定し，閾値処理により黒目部分を抽出してその重心点を用いる手法により視線を抽出している。しかしこの方法では，特に下方向に視線を移したときにまぶたの影響が大きくなり，視線方向の推定の誤差が大きくなることが予想される。岸本ら⁽⁷⁾は閾値処理，輪郭線抽出により抽出した黒目領域を円形と仮定し，2次元ハフ変換により中心座標を算出してその時間変化によりカーソルを移動する手法を提案している。しかし黒目の部分が円形であるという仮定はまぶたで隠れている部分を考えると誤差が大きく，個人差が大きく生じる可能性が高い。北川ら⁽⁸⁾は楕円で近似した黒目とベジェ曲線により近似したまぶたのアイモデルを構築し，パーティクルフィルタによるまぶた追跡を行うことで視線方向を推定している。この方法では，目尻と目頭の位

* Corresponding author. E-mail: shibata.satoru.mg@ehime-u.ac.jp

^a 愛媛大学

〒790-8577 愛媛県松山市文京町 3

Ehime University

3, Bunkyo-cho, Ehime, Japan 790-8577

置をあらかじめ指定しておく必要がある。また、中沢ら⁹⁾は閾値処理により検出した黒目部分とベジェ曲線により近似したまぶたの形状をもちいて視線推定を行っている。しかしながら、まぶたの形状は個人差が大きく、安定した検出に課題を有すると考えられる。

そこで本報告では、自然光の下、安価かつ設置の容易な市販のパソコンと USB カメラを用いた構成で、実用的な視線方向推定手法を提案する。提案手法では、まず目尻、目頭の検出を行う。次に、自動追跡のアルゴリズムとして動画像中の対象追跡に成果を上げているパーティクルフィルタを黒目部分の認識、運動推定に適用する。特に、パーティクルフィルタの尤度を、瞼によって隠れた黒目部分を補うように算出する点に特徴を有する。そして得られた目尻、目頭の座標、黒目の代表点を入力、画面上の視線位置を出力としたニューラルネットワークを構成し、非線形な写像を精度良く行うことにより、実用的な視線方向の推定を試みる。この提案方法と従来の閾値処理のみによる黒目部分の認識手法を比較し、その有効性について検討する。

2. 視線入力システムの構成

提案する視線入力システムの構成を Fig.1 に示す。ユーザはベッド上の患者を想定しており、顔の向く正面方向にモニタ画面が設置されている。モニタ画面には USB カメラが装着されており、ユーザの顔を撮影する。ここでは、画面の見やすさを考慮してモニタおよび USB カメラからユーザの顔までの距離は 600mm とする。撮影した画像から視線方向の情報を取得し、マウスカーソルを移動させる。なお、本研究では詳述しないが、画面上の項目の選択などは、一定時間の注視や随意的な閉眼により可能であることを確認している。

なお、PC は CPU が Intel core i7, 2.7GHz でメモリは 8GB である。Web カメラは 33fps で解像度は 640×480 となっている。

3. 視線方向推定のための画像処理手順

ALS などの重度肢体不自由者は、症状が進行すると眼球以外の随意運動が困難になることが知られている。ここではそのような状態を仮定し、ユーザは顔や首を動かさない状態で目の動きのみにより視線方向を変化させるものとする。この仮定のもと、視線方向の変化は撮影した目の黒目部分の変化に対応すると考えられ、本報告では目尻、目頭に対する黒目位置の変化により視線方向を推定することを基本とする。

本報告における画像処理手順は以下のとおりである。USB カメラからリアルタイムで上半身の画像が取得され、その画像から顔および目が検出される。その後、検出した目の領域に対しメディアンフィルタ処理、グレイ

Fig. 1. System configuration of eye gaze input.

スケール化処理、二値化処理などの画像変換処理を行うことで目尻、目頭の情報を取得する。そして、パーティクルフィルタを用いて黒目の代表点の検出および運動推定を行う。これらの目尻、目頭、および黒目の位置からニューラルネットワークを用いて視線方向の推定を行う。

3.1 顔の検出

まず、USB カメラにより得られたカラー画像(幅 640pixel×高さ 480pixel)から、OpenCV 付属の顔検出機能 Haar-like を用いて、顔の検出を行う。顔の検出結果(赤色の矩形)を Fig.2 に示す。

本研究では、ユーザとして ALS などの重度肢体不自由者を対象としており、システムを利用する間には顔の位置がほとんど動かないことを仮定している。この仮定のもと、顔領域の抽出は最初の 1 回のみ行っており、その際抽出した領域を記憶しておく。そして以降の処理については、記憶したその領域のもとに処理を繰り返す。

3.2 目の検出

次に、得られた顔の矩形領域から目を検出する。その際、USB カメラからユーザの顔までの距離が変化することにより画像に占める目の割合が変化する。そのような割合の変化により画像処理精度にばらつきが生じる。そこで、USB カメラからユーザの顔までの距離が変化しても安定した処理を行えるように、顔の矩形領域を切り取り 400×400pixel に正規化する。その後、OpenCV 付属の目検出機能 Haar-like を用いて顔の正規化画像から目を検出する。目の検出結果(白色の矩形)を Fig.3 に示す。

3.3 黒目の検出

3.2 節で検出した目の矩形領域を切り取り、そのカラー画像から黒目を検出する。まず、黒目表面にある照明等の反射の影響により白くなった部分を除去するためにメディアンフィルタをかける(Fig.4)。次に、メディアンフィルタ後のカラー画像を OpenCV の関数を用いてグレイスケール画像に変換する(Fig.5)。そして黒目を検出するために、グレイスケール画像を二値化画像に変換する。

Fig. 2. Detection of face.

Fig. 3. Detection of eyes.

(a) white part influenced by light

(b) Image after median filter processing

Fig. 4. Removal of white part using median filter processing.

Fig. 5. Image after gray scaling with median filter processing.

Fig. 6. Detection of black eye by selecting optimum threshold.

その際、照明などの明るさを考慮してユーザ毎に望ましい二値化処理用の閾値を設定しなければならない。そこで、望ましい閾値(黒目の検出用)を設定するために以下の処理を行う。

まず、目のグレースケール画像を閾値 1 から黒目部分が表出するまで値を上昇させながら二値化画像に変換する。次に、黒目部分が確認されたら、黒目の上端(Fig.6のA点)と下端(B点)の検出を行う。そして、矩形の上から下方向に1行ずつ黒点の検索をかけ、初めて出てきた黒点を上端とする。次に、矩形の下から上方向に1行ずつ黒点の検索をかけ、初めて出てきた黒点を下端とする。そして、上端と下端の座標値から黒目の高さ(長さ H)を算出する。黒目の高さの望ましい値は被験者ごとにあらかじめ求めておき、この値に達するまで閾値を上げて二値化処理を繰り返す。黒目の高さが望ましい値に達した際の閾値 Y を望ましい閾値とする。

また、黒目の検出結果から黒目の重心を算出する(Fig. 6のE点)。さらに、黒目を円と考えたときの半径を求めるため、黒目の左端と右端を検出する。検出では、正面を向いた際の二値化画像(Fig.6)の左から右方向に1列ずつ黒点の検索をかけ、初めて出てきた黒点を左端(C点)とする。次に、矩形の右から左方向に1列ずつ黒点の検索をかけ、はじめて出てきた黒点を右端(D点)とする。その後、黒目の上端の y 座標を Ay 、下端の y 座標を By 、左端の x 座標を Cx 、右端の x 座標を Dx とし黒目を円と考えたときの半径 r を式(1)により算出する。なお、座標系は矩形の左上を原点とし、水平方向を x 軸、鉛直方向を y 軸とする。

$$r = (By - Ay) / 2 + (Dx - Cx) / 2 \quad (1)$$

3.4 目尻と目頭の検出

本研究では、視線方向の推定に目尻と目頭の位置情報が用いられる。これらの検出のために以下の処理を行う。まず、3.2 節で検出した目の矩形領域を切り取り、カラー画像をグレースケール画像(Fig. 7)に変換する。その後、閾値処理により、二値化画像への変換を試みる。Fig.8は(a)が閾値 24 のときの画像であり、(b)が閾値 62 のときの画像である。閾値を小さな値から上昇させていくと最初黒目の部分が検出され、続いて目尻、目頭の部分が検出される。検出された領域の幅と高さを図のように定義し、両者の差をグラフ化すると Fig. 9 のように変化する。

Fig. 7. Image after gray scaling.

(a) result when threshold is 22

(b) result when threshold is 50

Fig. 8. Detected areas with different threshold.

Fig. 9. Difference between width and height under different threshold.

Fig.9を見ると、閾値 26 以下ではほぼ横ばいの 10 未満の値を示している。これは黒目の部分のみが現れていることを示している。27 から 28 のときに値が大きく変化し 30 付近の値になっているが、これは目尻が現れたことを示している。さらに閾値 37 から 38 のときに値が大きく変化し 50 付近の値になっているが、これは目頭が現れたことを示している。その後、ほぼ一定に変化している。値の大きな変化が生じる閾値は個人差や使用環境によってやや異なるが、これが生じることは共通して得られる特徴であった。このことより、現れた領域の幅と高さの差を調べ、値が大きく変化してその後ほぼ一定の値になるときの閾値を用いると、目尻・目頭を含む画像を得ることが可能である。

Fig. 10. Detection of outer corner and inner corner of eye.

Fig. 11. Trimmed area of pupil.

目尻と目頭の検出では、矩形領域の左から右方向に 1 列ずつ黒点の検索をかけ、初めて出てきた黒点を目尻 (Fig. 10 の F 点) とする。次に、右から左方向に 1 列ずつ黒点の検索をかけ、初めて出てきた黒点を目頭 (G 点) とする。また次節の処理のため目の横幅 (長さ W) を算出しておく。

3.5 目領域の切り取り

3.3 節と 3.4 節で得られた座標 (黒目の上端と下端、目尻と目頭) と目の高さ and 幅をもとに、目の二値化画像 (Fig. 6) の目領域を切り取る (Fig. 11)。

3.6 黒目の輪郭検出

後述するパーティクルフィルタによる黒目の運動推定では黒目を円と捉え、まぶたにより隠された部分を補った推定を実現する。ここでは、瞼により隠されていない黒目部分の輪郭を検出する方法について説明する。

まず、ユーザの視線方向を水平方向について、左方向、中央方向、右方向に分類する。具体的には、3.3 節で得られた閾値を用いて黒目の二値化画像 (Fig. 12) を作成し、黒目重心を算出する。その後、3.3 節での正面を向いた際の黒目重心と比較することで方向判別を行う。方向判別では、本節での黒目重心が左に 10pixel 以上離れている場合を左方向、右に 10pixel 以上離れている場合を右方向、それ以外を中央方向とする。

方向判別の結果をもとに目の二値化画像から黒目の輪郭を検出する。輪郭の検出方向は視線方向が左である場合は、黒目の右側の輪郭線を検出し、視線方向が右である場合は、黒目の左側の輪郭線を検出する。また視線方向が中央である場合は、視線方向が右側であるときと同じ処理を行う。

Fig. 12. Detection of outline on black eye.

輪郭の検出は、3.4節の目尻と目頭の検出方法を応用して行う。まず、3.4節と同様に端点の検出を行う。その後、さらに検出する列を進め、端点から検出方向へ上と下から黒点を検索し輪郭を検出する(Fig. 12)。

4. パーティクルフィルタ処理

目の固視微動や視線を移動させたときの目の運動に対応する黒目の検出を行うために、自動追跡のアルゴリズムとして動画像中の対象追跡に成果を上げているパーティクルフィルタ処理を行う。パーティクルフィルタ処理の流れは、まずパーティクルの初期化を行い目のカラー画像上にパーティクルを撒く。次にパーティクルの予測を行い、予測値をもとに尤度を計算する。その後、パーティクルの重み付け、重み付き平均の算出を行う。そして、パーティクルの重みをもとにリサンプリングを行う。これらの処理により、パーティクル群の重心算出、すなわち黒目の代表点算出を行う。

4.1 初期化

m 個の初期パーティクル $s_0 = \{x_0, v_0, w_0\}$ を一様乱数で初期化し、画像上に撒く。ただし、 x_0 は座標の初期値、 v_0 は速度の初期値、 w_0 は重みの初期値とする。本研究では、パーティクルの数を 50 個とする。

4.2 予測

各パーティクルをモデルに基づいて前時刻の状態から現時刻の状態へと移動させ予測値とする。動作モデルには、ランダムウォークを用いる。状態ベクトルを画像上の座標 $X_t = [x_t, y_t]$ とし、システムノイズを共分散 Σ_2 の 2 次元正規分布とすると

$$X_t = X_{t-1} + v_t \quad (2)$$

で表される。ただし $v_t \sim N(0, \Sigma_2)$ である。

4.3 尤度計算

現在の観測とパーティクルの予測値をもとにパーティクルの尤度を計算する。尤度の計算について Fig.13 を例

に説明する。3.6 節で検出した黒目輪郭の点(青色の点)の数を n 個とする。あるパーティクルと黒目輪郭の点との距離を d_i ($i=1,2,3,\dots,n$) とし、式(2)を用いて 3.3 節で算出した半径 r との差 E_i ($i=1,2,3,\dots,n$) を算出する。そして、式(3)を用いてパーティクルの尤度 L_j ($j=1,2,3,\dots,m$) を算出する。 ΣE が小さい、つまり黄色の破線円と黒目輪郭の点が近いほど尤度は大きくなる。

$$E_i = |d_i - r| \quad (3)$$

$$L_j = 1 - \frac{\sum_{i=1}^n E_i}{r \cdot n} \quad (4)$$

4.4 重み付け・重み付き平均の算出

4.3 節で計算した尤度をもとに式(4)を用いて、パーティクルの重み w_t を算出する。

$$w_t = \frac{L_j}{\sum_{j=1}^m L_j} \quad (5)$$

算出した重みをもとにパーティクル群の重心を算出する。本研究プログラムでは、重み付き平均をパーティクル群の重心とする。

Fig. 13. Calculation of likelihood.

(a) Results when moving eye-gaze at higher part of the monitor.

(b) Results when moving eye-gaze at lower part of the monitor.

Fig. 14. Estimated black eye.

パーティクルフィルタ処理を繰り返すことで、Fig.13の黄色の破線円が黒目輪郭の点に近づく、つまりまぶたで隠れた部分を補った黒目の代表点を算出することができる(Fig.14)。Fig.14での赤色の点はパーティクル群の重心であり、黄色の円はパーティクル群の重心から半径 r (3.3節で算出)で描いたものである。また、(a)はある被験者が画面上部において視線を右から左に移したものであり、(b)は別の被験者が画面下部において視線を右から左に移したものである。

5. ニューラルネットワークを用いた視線方向推定

本研究においては、重度肢体不自由者を対象に考えており視線入力を行う際に顔がほとんど動かないことを仮定しているため、顔画像の取得は最初の1回のみ実施し、以降の処理ではその領域内で目尻、目頭の認識およびパーティクルフィルタによる黒目の推定を行っている。認識した目尻、目頭の位置およびパーティクルフィルタにより推定された黒目の位置を用いて、ユーザの視線方向の推定を行うことが有効であるが、前者を入力、後者を出力と考えた場合、それらの関係は非線形になる。そこで、それらの関係を表現するのに、学習性に優れ、任意の入出力関係を設定することのできるニューラルネットワークを導入する。

5.1 ニューラルネットワークの構成

本研究で採用したニューラルネットワークは、Fig.15に示されるように3層構造とし、入力層は目尻、目頭、黒目の推定位置の左右の目に対する座標値の12個のユニット、中間層は10個のユニット、出力層はモニタ画面上の視線位置座標値の2個のユニットにより構成する。

Fig. 15. Construction of neural network.

入力の教師信号を T_i とすると、入力値 I_i は、入力の教師信号をそのまま出力するので式(6)で表される。

$$I_i = T_i \quad (6)$$

なお、添字 i は入力層 i 番目を意味する。

入力層の出力 x_i は、入力値をそのまま出力するので次式で表される。

$$x_i = I_i \quad (7)$$

入力層 i 番目と中間層 j 番目間の重み係数を w_{ji} とすると中間層の出力 h_j は、次式で表される。

$$h_j = f \left(\sum_i x_i w_{ji} \right) \quad (8)$$

なお、添字 j は中間層 j 番目を意味し、 f は次式に示されるシグモイド関数である。

$$f(x) = 0.5 X_g \frac{\{1 - \exp(-4.0x/X_g)\}}{\{1 + \exp(-4.0x/X_g)\}} \quad (9)$$

ただし、 $X_g = 2.0$ である。

中間層 j 番目と出力層 k 番目間の重み係数を w'_{kj} とすると、出力層の出力 O_k は、式(10)で表される。

$$O_k = f \left(\sum_j h_j w'_{kj} \right) \quad (10)$$

なお、添字 k は出力層 k 番目の意味、 f は式(9)と同じシグモイド関数である。

出力層の出力 O_k の教師信号を To_k とすると、出力層の逆伝播量 ΔO_k は、次式で表される。

$$\Delta O_k = (O_k - To_k) \left(1 - \frac{4O_k^2}{X_g^2} \right) \quad (11)$$

中間層の逆伝播量 Δh_j は、式(12)で表される。

$$\Delta h_j = \left(\sum_k \Delta O_k w'_{jk} \right) \left(1 - \frac{4h_j^2}{X_g^2} \right) \quad (12)$$

得られた逆伝播量を用いて、入力層と中間層間の重み係数を式(13)、中間層と出力層間の重み係数を式(14)のように修正する。

Fig. 16. Targets for teaching.

$$w_{ji} = -\eta x_i \Delta h_j \quad (13)$$

$$w'_{kj} = -\eta h_j \Delta O_k \quad (14)$$

なお、重み修正係数 $\eta = 0.01$ とした。

5.2 ニューラルネットワークの学習

教師信号を用いたニューラルネットワークの学習は以下のように行った。まず、Fig.16 に示されるようにモニタ画面上に 9 つの点を描画する。9 点の座標は、それぞれ、(50,50), (960,50), (1870,50), (50,520), (960,520), (1870,520), (50,1000), (960,1000), (1870,1000) とした。ユーザには 9 点をそれぞれ 100 サンプルずつ見てもらい、入力素子 12 個それぞれの座標を算出する。算出した値を入力教師信号として与えた。また、出力教師信号は描画した 9 点の座標値とした。なお、学習の繰り返し回数は 20000 回とした。

本提案手法では、画像取得の周波数である 33fps のフレームレート間隔 (30msec) 未満で、以上のすべての処理を完了することが可能となっている。

6. 視線推定性能の比較実験

ここでは、パーティクルフィルタを用いて黒目位置を推定する提案手法と、パーティクルフィルタを使用せずに黒目重心を検出する手法について、画面上の移動目標点をトラッキングする実験を行い、その追従特性について検討する。ただしこの実験は、視線方向の推定が比較的容易といわれる水平方向の移動に対する追従特性と、視線方向の推定が難しいといわれる縦方向の移動における追従特性についてそれぞれ行う。

まず、水平方向の移動に対する追従特性に関する実験は以下のように行う。Fig.17 に示す目標点(半径 20pixel の赤色の円)を水平方向に動かし、その目標点にマウスカーソルを追従させ、その際のマウスカーソルの挙動を調べる。まず、目標点をモニタ画面の中心点($x=960$, $y=540$)から右方向へ($x=1870$, $y=540$)の位置まで移動させる。

Fig. 17. Experiment for tracking a moving target by eye-gaze input using particle filter.

Fig. 18. Comparison of the method using particle filter with the method without particle filter when moving in horizontal direction.

Table 1. Cumulative error in horizontal movement.

Subjects	a	b	c	d	e
With PF	15200	17463	15524	7016	4925
Without PF	33026	22129	22249	16247	7035

次に、目標点を($x=1870$, $y=540$)から左方向へ移動させ、($x=50$, $y=540$)の位置まで移動させる。その後、($x=50$, $y=540$)から右方向へ移動させ、中心点まで移動させる。そして、その目標点を追従するようにユーザはマウスカーソルを操作する。その際、サンプリング数ごとの目標点とマウスカーソルの座標誤差を計測する。

Fig. 18にパーティクルフィルタを使用して視線方向推定を行った場合の追従誤差と、パーティクルフィルタを用いず単純に黒目重心を検出し、それを基に視線方向推定を行った場合の追従誤差の結果の代表例を示す。Fig. 18を見ると、パーティクルフィルタを使用しない手法では目標点とマウスカーソルの座標誤差が 200pixel を超える点が多く、パーティクルフィルタを使用した手法では座標誤差が 50 から 150pixel の点が多い。また、パーティクルフィルタを用いた手法は、用いない手法に比べて、有意に追従精度が高いことを確認している (有意水準 5% の t 検定)。

Fig. 19. Comparison of the method using particle filter with the method without particle filter when moving in vertical direction.

Table 2. Cumulative error in vertical movement.

Subjects	a	b	c	d	e
With PF	3611	3717	3205	3966	3374
Without PF	5753	5399	4428	5794	5144

また、5 人の被験者に対して実験を行った時の累積誤差を Table 1 に示す。表を見てわかるように、すべての被験者についてパーティクルフィルタを用いたほうが用いない場合に比べて累積誤差が小さくなっている。この結果からわかるように、対象となる被験者（顔や目の特徴）が異なっても提案した手法は良好に視線の推定を行うことができ、パーティクルフィルタを用いることの有効性を確認することができる。

次に、縦方向の移動に対する追従特性に関する実験は以下のように行う。Fig. 17 に示す目標点を縦方向に動かし、その際の仮想マウスカーソルの挙動を調べる。まず、目標点をモニタ画面の中心点($x=960$, $y=540$)から上方向へ($x=960$, $y=50$)の位置まで移動させる。次に、目標点を($x=960$, $y=50$)から下方向へ移動させ、($x=960$, $y=1030$)の位置まで移動させる。その後、($x=960$, $y=1030$)から上方向へ移動させ、中心点まで移動させる。そして、その目標点を追従するようにユーザはマウスカーソルを操作する。その際、サンプリング数ごとの目標点とマウスカーソルの座標誤差を計測する。Fig. 19 にパーティクルフィルタを使用して視線方向推定を行った場合の追従誤差と、黒目重心を検出しそれを基に視線方向推定を行った場合の追従誤差の結果を示す。

Fig. 19 を見ると、パーティクルフィルタを使用しない手法では目標点とマウスカーソルの座標誤差が 200 から 350pixel の点が多く、パーティクルフィルタを使用した

手法では座標誤差が 50 から 200pixel の点が多い。また、パーティクルフィルタを用いた手法は、用いない手法に比べて有意に追従精度が高いことを確認している（有意水準 5% の t 検定）。

また、5 人の被験者に対して実験を行った時の累積誤差を Table 2 に示す。表を見てわかるように、すべての被験者についてパーティクルフィルタを用いたほうが用いない場合に比べて累積誤差が小さくなっている。この結果からわかるように、垂直方向の移動においても、対象となる被験者が異なっても提案した手法は良好に視線の推定を行うことができ、パーティクルフィルタを用いることの有効性を確認することができる。

実用的な観点に立つと、ウィンドウズのアプリケーションを視線入力により選択・起動することが可能であることが重要である。本手法のアプリケーションとして、視線の移動によりウィンドウズ画面上の望ましいアイコンやボタン選択し、随意的閉眼（瞬きではなく意図的に目を閉じること）により選択したアイコンやボタンを開くという機能を開発しているが、アイコンやボタンの大きさが 20mm×20mm 以上であれば問題なく動作することを確認している。

7. おわりに

本報告では、自然光の下、安価かつ設置の容易な市販のパソコンと USB カメラを用いた視線認識システムを対象に、目尻、目頭の部分の認識、および自動追跡のアルゴリズムとして有用なパーティクルフィルタを用いた黒目部分の認識を行い、それらからニューラルネットワークを用いて視線方向を推定する新たな方法を提案した。そして、この提案方法と従来の閾値処理のみによる黒目部分の認識にもとづく視線方向推定手法を比較し、その有効性について検討した。その結果、パーティクルフィルタに基づく提案手法を用いた方が、従来の方法に比べて高精度に画面上の目標点の移動に追従することが可能であることを確認した。

照明の影響について考察すると、室内で蛍光灯や白熱灯が天井にある一般的な環境では、明るさの違いは閾値の変化によって対応でき、本システムを利用することが可能である。しかしながら光源が極端に利用者に近い場合や限られた方向からのみ顔に光が当たる場合、部屋が暗い場合には利用することが難しく、今後の課題である。

さらに、メガネを着用している人が本システムを利用した場合についてであるが、メガネのレンズに反射した環境の像の数や種類、フレームに対する目の大きさなどの条件により正しく視線が抽出できない場合があり、これらの条件でも望ましい視線抽出が行えるよう、今後取り組んでいきたい。

- (1) Tobii Pro Japan:「視線制御(入力)によるコミュニケーション」,
<http://www.tobii.com/ja-JP/eye-tracking-research/japan/about-tobii/history-at/eye-controlled-communication/>
- (2) 久野悦章, 八木透, 藤井一幸, 古賀一男, 内山嘉樹:「EOG を用いた視線入力インタフェースの開発」, 情報処理学会論文誌, Vol.39, No. 5, pp. 1455-1462, 1998
- (3) 伊藤和幸, 数藤康雄, 伊福部達:「重度肢体不自由者向けの視線入力式コミュニケーション装置」, 電子情報通信学会論文誌, Vol.J83DI, No. 5, pp. 495-503, 2000
- (4) 石原忍, 小眼科学改訂第 22 版, 金原出版, 1991
- (5) 落合積, 石松隆和, 高見修, 松井稜治:「目の動きを利用した身障者用文字入力装置の試作」, 日本機械学会論文集 (C 編), Vol.63, No. 609, pp. 140-143, 1997
- (6) 富本悠公, 田中幹也, 桐原正敬, 中島翔太, 春山和男:「目で操作する上肢障害者食事支援ロボット」, ライフサポート, Vol.27, No. 4, pp. 132-137, 2015
- (7) 岸本研志, 米村貴裕, 広瀬健一, 長江貞彦:「カーソル移動方式による視線入力システムの開発」, 映像情報メディア学会誌, Vol.55, No. 6, pp. 917-919, 2001
- (8) 北川洋介, 加藤丈和, 呉海元, 和田俊和:「アイモデルを用いた ConDensation による視線推定」, 情報処理学会研究報告, 2005-CVIM-150, pp. 17-24, 2005
- (9) 中沢信明, 瀬川辰乃佑, 村上裕紀, 松井利一:「瞼の形状近似を利用した操作インタフェースの開発」, ライフサポート, Vol.27, No. 4, pp. 124-131, 2015

柴田 論

1994 年 3 月東北大学大学院工学研究科博士後期課程機械工学専攻修了。2007 年 4 月愛媛大学大学院理工学研究科准教授, 2013 年 7 月同大学大学院理工学研究科教授, 現在に至る。人間工学, 福祉工学, 制御工学の研究に従事。

呉 志強

1995 年 3 月豊橋技術科学大学工学研究科総合エネルギー工学専攻博士後期課程修了。同年 4 月愛媛大学助手, 2004 年 10 月愛媛大学講師, 現在に至る。最適化, 福祉工学, 計算力学, 機械力学の研究に従事。

山本 智規

2002 年 3 月愛媛大学大学院理工学研究科博士後期課程生産環境工学専攻。2004 年 4 月愛媛大学助手, 2015 年同大学大学院理工学研究科准教授, 2016 年同大学社会共創学部, 現在に至る。ロボット工学, 福祉工学の研究に従事。