

# 論文

## 連動するパーティクルフィルタによる瞳追跡

大谷 悠祐<sup>†\*</sup> 福井 和広<sup>†</sup>

Pupils Tracking Using Multiple Coordinating Particle Filters

Yusuke OHTANI<sup>†\*</sup> and Kazuhiro FUKUI<sup>†</sup>

あらまし 本論文は、連動する複数のパーティクルフィルタを提案し、それに基づいて顔向き・視線変動に対してロバストな瞳追跡を実現する。瞳の動きは頭部運動と眼球運動の組合せからなる。このような複雑な動きをする対象物を追跡する枠組みとして、パーティクルフィルタが有効である。これを用いた瞳追跡として、左右の瞳をそれぞれ個別のパーティクルフィルタにより追跡するという単純な方法が考えられるが、追跡の安定性に不安が残る。そこで、本論文では両瞳が筋肉により連動して動くことに着目し、通常のパーティクルフィルタの枠組みに、瞳の次時刻の状態を、自身の状態からだけでなく他方の瞳状態からも推定する枠組みを導入する。頭部や視線が大きく変化する状況において、提案法の有効性を検証した。その結果、左右の瞳を個別に追跡する方法に対して、提案法では極めて安定に追跡を行えることを確認できた。

キーワード 瞳追跡, パーティクルフィルタ, 分離度フィルタ

### 1. ま え が き

画像から瞳を抽出・追跡する技術は、顔認識、表情認識、視線推定などの顔画像処理における基本要素であり、人間とロボットとのアイコンタクトやドライバーの状態モニタリングなど、様々な応用が期待されている[1]~[4]。これらのアプリケーションにおいて、瞳追跡を難しくする大きな要因として、「カメラに対する顔の向き」と「視線」による瞳の見え方の複雑な変動がある。本論文ではこれらの変動に対して頑健な追跡に追跡を目指す。

これまでに様々な瞳検出・追跡のための方法[5]~[7]が提案されている。本論文では特に有効性が確認されている方法として、分離度フィルタを用いた方法に注目する[8]~[11]。分離度フィルタは瞳のような円状の領域を高精度・安定に抽出するフィルタであり、瞳検出に適用する場合には、このフィルタにより抽出された複数の瞳候補に対して、パターン照合[8],[10]、ハフ変換[9]、更新型テンプレート[11]などの後処理を

適用して、正しい瞳を選択する。

瞳の動きは頭部運動と眼球運動の複雑な組合せにより形成されている。このような複雑でモデル化が難しい動きを追跡する方法として、パーティクルフィルタ[12],[13]が有効であることが知られている。パーティクルフィルタはベイジアンフィルタ[14]の一種であり、各時刻で更新ステップと予測ステップを繰り返しながら、確率的に対象を追跡する方法である。更新ステップでは、追跡対象の状態の事前確率分布と観測画像に対するゆう度を用いて事後確率分布を求める。予測ステップでは、追跡対象の状態の事後確率分布と状態遷移モデルを用いて次時刻の事前確率を求める。パーティクルフィルタでは、事前分布・事後分布を多数のサンプル(粒子)によって表現する。これによって任意の確率分布を表現でき、線形予測では対応が難しい動きにも対応できる。

パーティクルフィルタを瞳追跡に適用する場合、最も素朴には左右の瞳をそれぞれ独立にパーティクルフィルタを用いて追跡する方法が考えられる。しかしながら、両瞳は筋肉により連動して動作することを考えると、より安定な追跡を実現するためには、両者の拘束関係をパーティクルフィルタの枠組みに反映させることが有効であると考えられる。

両瞳の拘束関係を反映した方法として、頭部と眼球

<sup>†</sup> 筑波大学大学院システム情報工学研究科, つくば市  
Graduate School of Systems and Information Engineering,  
University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, 305-  
8573 Japan

\* 現在, セイコーエプソン株式会社

の幾何学的な関係を反映した三次元モデルを用いた方法 [15], [16] が提案されている。しかしながら、頭部と瞳の拘束関係は複雑であり、両者の関係を完全にモデル化することは難しい。仮にモデル化できたとしても、単純なモデルでは拘束関係を正確に反映させることが難しいと考えられる。

両瞳の拘束関係を反映するために、二つの瞳の状態変数を合わせて一つのパーティクルフィルタで追跡することも考えられるが、その場合には状態ベクトルの次元数が増え、それに伴い膨大な数のサンプルが必要となる。サンプル数を削減する方法として、追跡対象の形状や運動の変動を表す空間を、主成分分析を適用して事前に求めておき、サンプル生成をこの空間内に限定する方法 [17] や、主成分分析によって状態空間の次元を減らす方法 [18] などが提案されている。しかしながら、瞳と頭部の組合せで生じる複雑な動きを、そのような単純な線形空間で高精度に近似することは難しいと考えられる。また、両瞳の状態を一つの状態ベクトルで表現する場合、片目が観測できない場合には、データ欠損の影響が状態ベクトルに直接的に及ぶと考えられ、追跡が不安定になる可能性が高いと推察される。

これらに対して、本論文では両瞳の幾何学的な拘束関係を陽にモデル化せずに確率的に反映させる。具体的には、複数のパーティクルフィルタをお互いに連動させることで、両瞳の拘束関係を追跡処理に反映させる。従来のパーティクルフィルタでは予測ステップにおいて、一方の瞳のサンプル（粒子）は一時点前のその瞳のサンプル状態のみに依存して生成される。これに対して提案法では、一時点前の他方の瞳のサンプル（粒子）の状態にも依存して生成される。また、更新ステップにおいて、左右瞳をセットとして得られる特徴量からゆう度を求めることで、事後分布の生成にも左右瞳の拘束関係を反映させる。

複数の事前確率から生成したサンプルを混合するという点では、提案法は Boosted Particle Filter [19] と同じアイデアに基づく。しかし、この方法では、パーティクルフィルタによる追跡とは別の手段で追跡対象を検出し、その結果に基づいて追跡対象の付近にサンプルを生成しているために、別手段による追跡の精度・安定性が全体性能に影響する可能性が考えられる。

以降では、2. で提案法の基本アイデアについて述べ、3. において提案法のアルゴリズムを述べる。4. で提案法の有効性を評価する実験を行い、その結果について

考察する。5. では結論と今後の課題について述べる。

## 2. 提案法のアプローチ

本章では、まず提案法の基本となるパーティクルフィルタを用いた瞳追跡の枠組みについて述べる。次にこの枠組みに両瞳の拘束関係を組み入れる方法について述べる。

### 2.1 パーティクルフィルタによる瞳追跡

パーティクルフィルタによる瞳追跡は、前時刻  $t-1$  の瞳状態ベクトル  $\mathbf{x}_{t-1}$  の事前分布  $p(\mathbf{x}_t|\mathbf{Y}_{t-1})$  と現時刻  $t$  の観測画像  $\mathbf{y}_t$  から、現時刻  $t$  の瞳状態  $\mathbf{x}_t$  の事後分布  $p(\mathbf{x}_t|\mathbf{Y}_t)$  を逐次的に求めていく問題である。ここで、 $\mathbf{Y}_t$  は時刻  $t$  までの観測画像の集合  $\{\mathbf{y}_i\}_{i=1}^t$  である。

パーティクルフィルタでは、瞳状態の事前分布と事後分布を、それぞれ異なる状態ベクトルを保持している多数のサンプル（粒子）集合  $\mathbf{S}$  で表現する。瞳形状を楕円とする場合には、状態ベクトル  $\mathbf{x}_t = (x, y, a, b, \theta)^T$  は楕円の中心座標  $(x, y)$ 、長軸・短軸の長さ  $(a, b)$ 、楕円の水平軸に対する傾き角  $\theta$  の五つのパラメータから構成されることになる（図 4 参照）。

追跡処理は予測ステップと更新ステップから構成される。まず予測ステップでは、前時刻  $t-1$  の瞳状態ベクトル  $\mathbf{x}_{t-1}$  の事後分布を表すサンプル集合  $\mathbf{S}_{t-1|t-1}$  に状態予測関数を適用して、現時刻  $t$  の事前分布を表すサンプル集合  $\mathbf{S}_{t|t-1}$  を生成する。次に更新ステップでは、観測画像にゆう度関数を適用することで各サンプルのゆう度（重み）を求める。そして、 $\mathbf{S}_{t|t-1}$  から各サンプルのゆう度（重み）に比例する割合で復元抽出することで、現時刻  $t$  における事後分布を表すサンプル集合  $\mathbf{S}_{t|t}$  を生成する。現時刻  $t$  における瞳状態ベクトルの期待値  $\hat{\mathbf{x}}_t$  は事後分布  $\mathbf{S}_{t|t}$  の平均として求まる。

### 2.2 両瞳の拘束関係の組み込み

両瞳の拘束関係をパーティクルフィルタの枠組みに組み込むために、先に述べたように、予測ステップにおける瞳状態ベクトルの推定を、前時刻における自身のサンプル集合と他方の瞳のサンプル集合に基づいて推定されたサンプル集合の両方を用いて行う。前者は従来のパーティクルフィルタで用いられるサンプル集合である。後者は提案法において新たに導入するサンプル集合であり、他方の瞳の位置・形状に基づいて推定（生成）される。このような推定は、両眼球は筋肉によって連動して動くことから、他方の状態が分かれ

ば、それに応じてもう一方の状態も推定できるだろうという期待に基づく。以上のように提案法は、二つのパーティクルフィルタが相補的に互いの状態を反映しながら状態遷移していく枠組みを提供する。

図1は従来法と提案法により生成された二つのサンプル集合を示している。図中で青色と赤色の点は、それぞれ一時刻前の右瞳の状態に基づいて生成されたサンプルと左瞳の状態に基づいて生成されたサンプルを示している。従来法では各瞳の周辺にのみサンプルが生成されている（同左図）。これに対して、提案法では他方の瞳状態に基づいて推定されたサンプルも同じ周辺に混在していることが分かる（同右図）。

他方のサンプル状態に基づいて、もう一方のサンプル状態を推定（生成）することを、“相補的なサンプル推定”と呼ぶことにする。つまり、右（あるいは左）瞳の状態（位置と楕円形状）に基づいて、そのとき、それに対応するもっともらしい、左（あるいは右）瞳の状態（位置と楕円形状）を推定するのである。提案フィルタでは、図2に示すように、まずこの相補的なサンプル推定を行い、続いて通常の状態遷移関数に基づく時間的な予測を行う。なお以降では相補的なサンプル推定と時間的な予測を合わせて状態遷移関数とみなす。

相補的なサンプル推定では、瞳サンプル  $\mathbf{x}_{\text{from}}$  と対

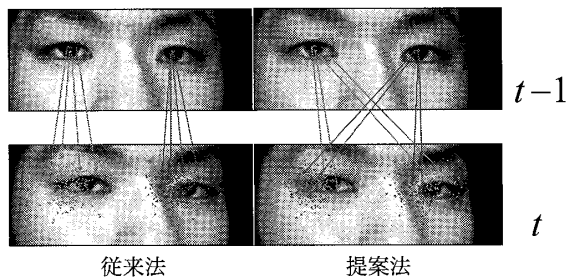


図1 パーティクルフィルタの予測ステップにおける従来法と提案法の比較

Fig. 1 Predict steps of conventional particle filter and proposed particle filter.

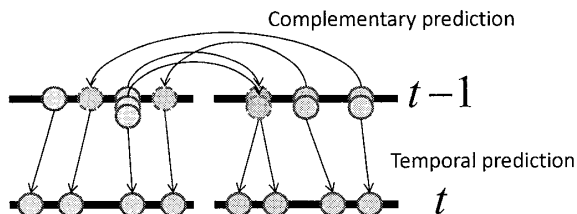


図2 予測ステップにおけるサンプルの動きの概念図

Fig. 2 Schematic diagram of behavior of samples in predict step.

応する他方の瞳サンプル  $\mathbf{x}_{\text{to}}$  を直接関連づけるのではなく、 $\mathbf{x}_{\text{from}}$  が表す輪郭情報に基づいて切り出された瞳周辺の正規化画像と、 $\mathbf{x}_{\text{to}}$  を対応づける。これは、 $\mathbf{x}_{\text{from}}$  には瞳の完全な情報が含まれず、輪郭周辺の瞳近傍画像にも含まれるからである。このように情報量の多い瞳近傍画像と状態ベクトル  $\mathbf{x}_{\text{to}}$  を対応づけることで、相補的な推定を高精度に行う。対応付けは大量の学習サンプルを用いた重回帰分析により実現する。この際、顔の位置・スケールの変化に対応するために、 $\mathbf{x}_{\text{to}}$  をそのまま対応づけるのではなく、 $\mathbf{x}_{\text{from}}$  を基準とした  $\mathbf{x}_{\text{to}}$  のパラメータ  $\dot{\mathbf{x}}_{\text{to}} = (\dot{x}, \dot{y}, \dot{a}, \dot{b}, \dot{\theta})^T$  と  $\mathbf{x}_{\text{from}}$  に基づいて切り出した画像を対応づける。ここで、状態ベクトル  $\dot{\mathbf{x}}_{\text{to}}$  の各成分は以下のように定める。

$$\begin{aligned}\dot{x}_{\text{to}} &= (x_{\text{to}} - x_{\text{from}}) / (a_{\text{from}} + b_{\text{from}}) \\ \dot{y}_{\text{to}} &= (y_{\text{to}} - y_{\text{from}}) / (a_{\text{from}} + b_{\text{from}}) \\ \dot{a}_{\text{to}} &= a_{\text{to}} / (a_{\text{from}} + b_{\text{from}}) \\ \dot{b}_{\text{to}} &= b_{\text{to}} / (a_{\text{from}} + b_{\text{from}}) \\ \dot{\theta}_{\text{to}} &= \theta_{\text{to}}\end{aligned}\quad (1)$$

ここで、 $\mathbf{x}_{\text{to}} = (x_{\text{to}}, y_{\text{to}}, a_{\text{to}}, b_{\text{to}}, \theta_{\text{to}})^T$ ,  $\mathbf{x}_{\text{from}} = (x_{\text{from}}, y_{\text{from}}, a_{\text{from}}, b_{\text{from}}, \theta_{\text{from}})^T$  である。

以上を整理すると、左瞳サンプルの状態ベクトル  $\mathbf{x}_t^L$ （位置と形状情報）に対して右瞳サンプルの状態ベクトル  $\mathbf{x}_t^R$  を出力する推定関数  $\mathbf{f}_{\text{pred}}^{LR}$ 、及び右瞳の状態ベクトル  $\mathbf{x}_t^R$  に対して左瞳の状態ベクトルを出力する推定関数  $\mathbf{f}_{\text{pred}}^{RL}$  は、以下ようになる。

$$\begin{aligned}\mathbf{f}_{\text{pred}}^{LR}(\mathbf{x}_t^L) &= \mathbf{f}_{\text{rel}}(\mathbf{B}^{LR} \mathbf{f}_{\text{cut}}(\mathbf{x}_t^L, \mathbf{y}_t) + \mathbf{b}_0^{LR}, \mathbf{x}_t^L) \\ \mathbf{f}_{\text{pred}}^{RL}(\mathbf{x}_t^R) &= \mathbf{f}_{\text{rel}}(\mathbf{B}^{RL} \mathbf{f}_{\text{cut}}(\mathbf{x}_t^R, \mathbf{y}_t) + \mathbf{b}_0^{RL}, \mathbf{x}_t^R)\end{aligned}\quad (2)$$

ここで、 $\mathbf{f}_{\text{cut}}(\mathbf{x}_t^L, \mathbf{y}_t)$ ,  $\mathbf{f}_{\text{cut}}(\mathbf{x}_t^R, \mathbf{y}_t)$  はそれぞれ、左と右の状態ベクトルに基づいて切り出した画像輝度をベクトル化する関数である。行列  $\mathbf{B}^{LR}$ ,  $\mathbf{B}^{RL}$ 、ベクトル  $\mathbf{b}_0^{LR}$ ,  $\mathbf{b}_0^{RL}$  は線形重回帰分析で求まる偏回帰係数を並べた行列、定数ベクトルである。この重回帰式を用いて切り出した画像から他方の瞳の相対パラメータを推定する。更に推定された相対パラメータ  $\dot{\mathbf{x}}_{\text{to}}$  を  $\mathbf{f}_{\text{rel}}(\dot{\mathbf{x}}_{\text{to}}, \mathbf{x}_{\text{from}})$  によって、最終的に必要な状態パラメータ  $\mathbf{x}_{\text{to}}$  に変換する。 $\mathbf{f}_{\text{rel}}(\dot{\mathbf{x}}_{\text{to}}, \mathbf{x}_{\text{from}})$  は具体的に以下のような式になる。

$$\begin{aligned}
\mathbf{f}_{\text{rel}}(\dot{\mathbf{x}}_{\text{to}}, \mathbf{x}_{\text{from}}) &= (x_{\text{to}}, y_{\text{to}}, a_{\text{to}}, b_{\text{to}}, \theta_{\text{to}})^T \\
x_{\text{to}} &= (a_{\text{from}} + b_{\text{from}})\dot{x}_{\text{to}} + x_{\text{from}} \\
y_{\text{to}} &= (a_{\text{from}} + b_{\text{from}})\dot{y}_{\text{to}} + y_{\text{from}} \\
a_{\text{to}} &= (a_{\text{from}} + b_{\text{from}})\dot{a}_{\text{to}} \\
b_{\text{to}} &= (a_{\text{from}} + b_{\text{from}})\dot{b}_{\text{to}} \\
\theta_{\text{to}} &= \dot{\theta}_{\text{to}}
\end{aligned} \quad (3)$$

時間的な予測としては単純なランダムウォークを用いる。以上により、相補的な位置推定と時間的な予測を合わせた状態遷移関数は次のようになる。

$$\begin{aligned}
\mathbf{f}_t^{RR}(\mathbf{x}_t^R, \mathbf{v}_t^{RR}) &= \mathbf{x}_t^R + \mathbf{v}_t^{RR} \\
\mathbf{f}_t^{LR}(\mathbf{x}_t^L, \mathbf{v}_t^{LR}) &= \mathbf{f}_{\text{pred}}^{LR}(\mathbf{x}_t^L) + \mathbf{v}_t^{LR} \\
\mathbf{f}_t^{LL}(\mathbf{x}_t^L, \mathbf{v}_t^{LL}) &= \mathbf{x}_t^L + \mathbf{v}_t^{LL} \\
\mathbf{f}_t^{RL}(\mathbf{x}_t^R, \mathbf{v}_t^{RL}) &= \mathbf{f}_{\text{pred}}^{RL}(\mathbf{x}_t^R) + \mathbf{v}_t^{RL}
\end{aligned} \quad (4)$$

ここで、 $\mathbf{f}_t^{RR}$  は、右瞳の状態から次時刻の右瞳の状態を予測する関数である。同様に、 $\mathbf{f}_t^{LR}$ ,  $\mathbf{f}_t^{LL}$ ,  $\mathbf{f}_t^{RL}$  はそれぞれ左瞳から右瞳を、左瞳から左瞳を、右瞳から左瞳を予測する関数である。 $\mathbf{v}_t^{RR}$ ,  $\mathbf{v}_t^{LR}$ ,  $\mathbf{v}_t^{LL}$ ,  $\mathbf{v}_t^{RL}$  はシステムノイズである。

### 2.3 両瞳の拘束関係を考慮したよう度

両瞳の拘束関係を事後確率分布の推定にも反映させるために、両瞳を含むように切り出された画像パターンを用いて、以下の手順でよう度を求める。

あらかじめ、学習フェーズにおいて、様々な人物、照明条件における顔画像から、左右瞳を含んだ切り出し画像（正解パターン）を多数用意しておき、これから辞書パターンを作成しておく。この切り出しは、各瞳の座標とサイズを基準に行う。よう度を求める際に

は、まず左右の瞳候補セットからそれぞれ瞳候補を一つずつ選択し、選択された両瞳候補を基準に、両瞳近傍パターンを切出し、これと先に学習した正解辞書パターンとの類似度を求めてよう度とする。正しい左右瞳の組合せに基づいて切出された近傍パターンは正解辞書パターンとの類似度、つまりよう度が高くなる。後で述べるように本論文ではパターン認識の代表的な方法である部分空間法 [20] を用いてパターン類似度を求める。

ただし、左右瞳候補の全ての組合せに対して、上記のよう度計算を行うことは、組合せ数が膨大になるために、計算困難となる。そこで更新ステップにおいて、左右瞳単独のよう度を求め、それに基づいて組合せ数を絞り込む。以降、左右瞳を組み合わせたよう度を複合よう度、左右瞳単独のよう度を独立よう度と呼ぶことにする。これらのよう度の具体的な計算法については、3.3で説明する。

## 3. 提案法を用いた瞳追跡

本章では、提案法における予測ステップと更新ステップの詳細について図3に沿って説明する。

時刻  $t$  における右瞳と左瞳の事前分布に従うサンプル集合を  $\mathbf{S}_{t|t-1}^R = \{\mathbf{s}_{t|t-1}^{(i),R}\}_{i=1}^N$ ,  $\mathbf{S}_{t|t-1}^L = \{\mathbf{s}_{t|t-1}^{(i),L}\}_{i=1}^N$ 、事後分布に従うサンプル集合を  $\mathbf{S}_{t|t}^R = \{\mathbf{s}_{t|t}^{(i),R}\}_{i=1}^N$ ,  $\mathbf{S}_{t|t}^L = \{\mathbf{s}_{t|t}^{(i),L}\}_{i=1}^N$  とする。ここで  $N$  は追跡に用いるサンプルの数である。なお、以下で用いる各パラメータの具体的な値については、4. の評価実験で述べる。

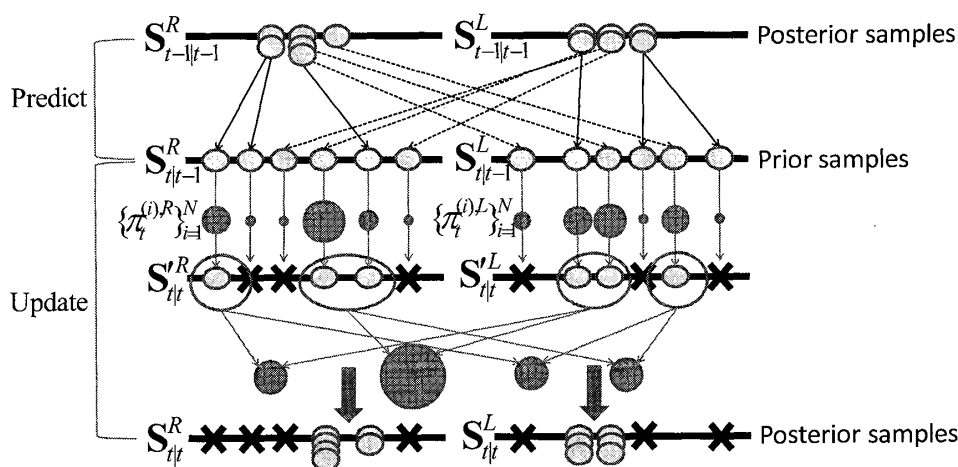


図3 提案パーティクルフィルタのアルゴリズム  
Fig.3 Algorithm of proposed particle filter.

### 3.1 予測ステップ

予測ステップでは、時刻  $t-1$  の事後サンプルから時刻  $t$  の事前サンプルを次の手順で生成する。

(1)  $i = 1, \dots, N$  に対して、一様乱数  $r^{(i)} \in [0, 1]$  を生成する。

(2) (1) で求めた乱数  $r^{(i)}$  に対して、以下のルールを適用した上で、式 (2) と式 (4) に基づいて、事後サンプルから事前サンプルを生成する。

$$\mathbf{s}_{t|t-1}^{(i),R} = \begin{cases} \mathbf{f}_t^{RR}(\mathbf{s}_{t-1|t-1}^{(i),R}, \mathbf{v}_t^{(i),R}) & r^{(i)} \geq \mu \\ \mathbf{f}_t^{LR}(\mathbf{s}_{t-1|t-1}^{(i),L}, \mathbf{v}_t^{(i),R}) & r^{(i)} < \mu \end{cases}$$

$$\mathbf{s}_{t|t-1}^{(i),L} = \begin{cases} \mathbf{f}_t^{LL}(\mathbf{s}_{t-1|t-1}^{(i),L}, \mathbf{v}_t^{(i),L}) & r^{(i)} \geq \mu \\ \mathbf{f}_t^{RL}(\mathbf{s}_{t-1|t-1}^{(i),R}, \mathbf{v}_t^{(i),L}) & r^{(i)} < \mu \end{cases} \quad (5)$$

ここで、 $\mu$  はあらかじめ与える値 ( $0 < \mu < 1.0$ ) で、他方の瞳の事後サンプルから生成される事前サンプルの割合を示している。 $\mathbf{v}_t^{(i),R}, \mathbf{v}_t^{(i),L}$  ( $i \in \{1, \dots, N\}$ ) は平均  $\mathbf{0}$ 、共分散  $\Sigma$  の多次元ガウス分布に従う乱数である。

### 3.2 更新ステップ

更新ステップでは、各サンプルのゆう度に基づいて事前サンプルから事後サンプルを次の手順で生成する。

(1) 各事前サンプルについて、独立ゆう度  $\pi_t^{(i),R}, \pi_t^{(i),L}$  を求める。

$$\pi_t^{(i),R} = \pi_{\text{like}}^R(\mathbf{s}_{t|t-1}^{(i),R}, \mathbf{y}_t)$$

$$\pi_t^{(i),L} = \pi_{\text{like}}^L(\mathbf{s}_{t|t-1}^{(i),L}, \mathbf{y}_t) \quad (6)$$

ここで、 $\pi_{\text{like}}^R, \pi_{\text{like}}^L$  は独立ゆう度関数である。

(2) 独立ゆう度が大きい順に  $\mathbf{S}_{t|t-1}^R, \mathbf{S}_{t|t-1}^L$  からそれぞれ  $N_k$  個ずつサンプルを抽出し、 $\hat{\mathbf{S}}_{t|t}^R = \{\hat{\mathbf{s}}_{t|t}^{(i),R}\}_{i=1}^{N_k}, \hat{\mathbf{S}}_{t|t}^L = \{\hat{\mathbf{s}}_{t|t}^{(i),L}\}_{i=1}^{N_k}$  とする。

(3) 集合  $\hat{\mathbf{S}}_{t|t}^R, \hat{\mathbf{S}}_{t|t}^L$  を 5 次元状態ベクトル空間におけるユークリッド距離を基準にそれぞれクラスタリングし、クラスタ  $\{C_t^{(i),R}\}_{i=1}^{N_t^R}$  と  $\{C_t^{(j),L}\}_{j=1}^{N_t^L}$  を得る。 $N_t^R, N_t^L$  はそれぞれクラスタの数である。クラスタリングには Mean-Shift 法 [21] を用いる。

(4) 各クラスタの全ての組合せ  $i, j \in \{1, \dots, N_t^R\} \times \{1, \dots, N_t^L\}$  について複合ゆう度  $\pi_t^{(i,j)}$  を計算する。

$$\pi_t^{(i,j)} = \pi_{\text{like}}^C(C_t^{(i),R}, C_t^{(j),L}, \mathbf{y}_t) \quad (7)$$

ここで、 $\pi_{\text{like}}^C$  は複合ゆう度関数である。

(5) 複合ゆう度が最大となるクラスタの組合せ

$(C_t^{(\hat{i}),R}, C_t^{(\hat{j}),L})$  を求める。

$$\hat{i}, \hat{j} = \underset{i,j}{\operatorname{argmax}} \pi_t^{(i,j)} \quad (8)$$

(6) サンプル集合  $\{\mathbf{s} | \mathbf{s} \in C_t^{(\hat{i}),R}\}$  と  $\{\mathbf{s} | \mathbf{s} \in C_t^{(\hat{j}),L}\}$  から、各サンプルの独立ゆう度に比例する割合で  $N$  個ずつ復元抽出し、 $\mathbf{S}_{t|t}^R, \mathbf{S}_{t|t}^L$  とする。

(7) 事後分布  $\mathbf{S}_{t|t}^R, \mathbf{S}_{t|t}^L$  の平均を時刻  $t$  における瞳の推定状態  $\hat{\mathbf{x}}_t^R, \hat{\mathbf{x}}_t^L$  とする。

$$\hat{\mathbf{x}}_t^R = \frac{1}{N} \sum_{i=1}^N \mathbf{s}_{t|t}^{(i),R}$$

$$\hat{\mathbf{x}}_t^L = \frac{1}{N} \sum_{i=1}^N \mathbf{s}_{t|t}^{(i),L} \quad (9)$$

### 3.3 ゆう度関数

#### 3.3.1 独立ゆう度

形状情報として分離度 [8]、パターン情報としてパターン照合結果を用いて独立ゆう度を定義する。

分離度は二つの領域に属する輝度などの情報がどれだけ分離されているかを表す値であり、ノイズや照明変動に対してロバストに求まる。二つの領域を図 4 のように楕円形状に設定することで、楕円形状の瞳輪郭に対する分離を求めることができる。具体的には、領域  $C$  を分割した二つの領域  $C_1, C_2$  に対して分離度  $\eta$  は以下のように定義される。

$$\eta = \frac{\sigma_b^2}{\sigma_T^2}$$

$$\sigma_b^2 = n_1(m_1 - m)^2 + n_2(m_2 - m)^2 \quad (10)$$

$$\sigma_T^2 = \sum_{x_i \in C} (x_i - m)^2$$

ここで  $x_i$  は画素  $i$  の輝度値、 $n_1, n_2$  は領域  $C_1, C_2$  に含まれる画素数、 $m_1, m_2, m$  は  $C_1, C_2$  及び全体の平均輝度値である。

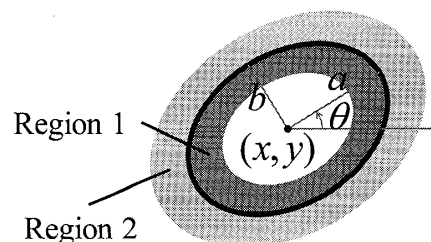
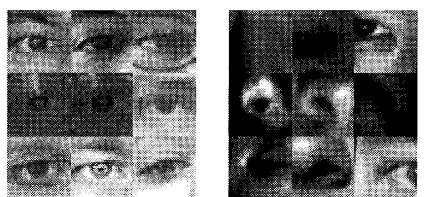


図 4 楕円分離度フィルタ

Fig. 4 Elliptical separability filter.



(a) Eye images (b) Non eye images

図 5 切り出された方形画像パターンの例

Fig. 5 Examples of cropped image patterns.

サンプルの状態ベクトルが与えられると、それが表す楕円形状を分離度フィルタのマスクとして分離度を求める。更に形状情報に加えてパターン情報もゆう度に考慮するために、状態ベクトルが表す楕円形状を基準に近傍パターンを切り出し、これが瞳か否かの判別を行う。画像パターンは、円形分離度が局所最大となる点を中心に設定し、楕円分離度フィルタの長軸と短軸を基準にして、 $15 \times 15$  に正規化した後に切り出す。判別空間はこのようにして収集した瞳画像パターン（図 5a）と非瞳画像パターン（図 5b）に線形判別分析を適用して生成しておく。

以上で述べた分離度とパターン照合の結果に基づいて、独立ゆう度関数  $\pi_{\text{like}}^R, \pi_{\text{like}}^L$  を以下のように定義する。

$$\pi_{\text{like}}^R(\mathbf{s}, \mathbf{y}) = \begin{cases} \eta(\mathbf{s}, \mathbf{y}) & (f_{\text{disc}}^R(\mathbf{s}, \mathbf{y}) = 1) \\ 0 & \text{else} \end{cases} \quad (11)$$

$$\pi_{\text{like}}^L(\mathbf{s}, \mathbf{y}) = \begin{cases} \eta(\mathbf{s}, \mathbf{y}) & (f_{\text{disc}}^L(\mathbf{s}, \mathbf{y}) = 1) \\ 0 & \text{else} \end{cases}$$

ここで、 $\eta$  は分離度を返す関数である。 $f_{\text{disc}}^R, f_{\text{disc}}^L$  は、それぞれ右瞳と非瞳、及び左瞳と非瞳の判別を行う関数で、瞳と判別したら 1 を、そうでない場合には 0 を返す。

### 3.3.2 複合ゆう度

複合ゆう度関数は、右瞳のサンプルのクラスタ  $C^R$ 、左瞳のサンプルのクラスタ  $C^L$ 、及び観測画像が与えられると複合ゆう度を出力する（式 (7) 参照）。

複合ゆう度は図 6 に示す右瞳画像、左瞳画像、及び両瞳を含む方形局所画像パターンに対するパターン照合から得られる三つのパターン類似度と、先に求めた独立ゆう度を用いて定義する。

パターン類似度は次のように求める。まず特徴量として、クラスタ平均の瞳位置を基準に切り出した画像から得る SIFT-like 特徴量を用いる [22]。この特徴量は、単一瞳から切り出した画像では  $4 \times 4$  に、両瞳か

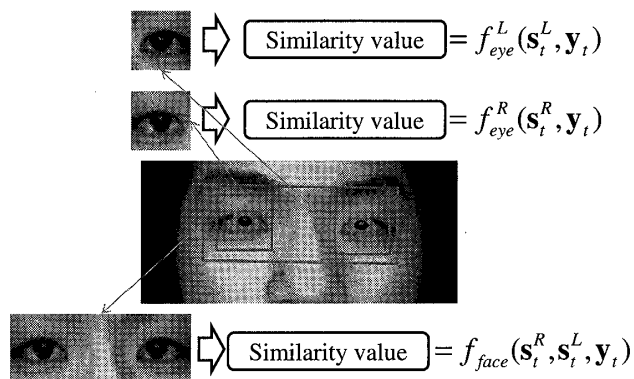


図 6 切り出された画像パターンに対するパターン照合

Fig. 6 Calculation of the similarity for cropped image.

ら切り出した画像では  $3 \times 8$  に領域分割し、領域ごとの 8-bin のこう配方向ヒストグラムを連結したベクトル量である。パターン照合には部分空間法を用いる。部分空間法では、あらかじめ学習用のパターン集合を KL 展開（主成分分析）して基底を求めておき、この基底によって張られる部分空間と入力パターンとの角度  $\theta$  を求める。この角度に対する  $\cos^2 \theta$  をパターン類似度とする。

上記のようにして得られた三つのパターン類似度と、その候補クラスタに含まれる全サンプルの独立ゆう度を用いて、複合ゆう度  $\pi_{\text{like}}^C$  を以下のように定義する。

$$\pi_{\text{like}}^C(C^R, C^L, \mathbf{y}) = F_1 \times F_2 \times F_3 \quad (12)$$

ここで、

$$F_1 = f_{\text{eye}}^R(\hat{\mathbf{s}}^R, \mathbf{y}) \times \sum_{\mathbf{s} \in C^R} \pi_{\text{like}}^R(\mathbf{s}, \mathbf{y})$$

$$F_2 = f_{\text{eye}}^L(\hat{\mathbf{s}}^L, \mathbf{y}) \times \sum_{\mathbf{s} \in C^L} \pi_{\text{like}}^L(\mathbf{s}, \mathbf{y}) \quad (13)$$

$$F_3 = f_{\text{face}}(\hat{\mathbf{s}}^R, \hat{\mathbf{s}}^L, \mathbf{y})$$

とする。 $\hat{\mathbf{s}}^R$  と  $\hat{\mathbf{s}}^L$  は各クラスタ  $C^R$  と  $C^L$  に属する全サンプルの重み付き平均である。この重みには各サンプルの独立ゆう度を用いる。 $f_{\text{eye}}^R$  と  $f_{\text{eye}}^L$  はそれぞれ右瞳と左瞳の近傍パターンから得られるパターン類似度である。 $f_{\text{face}}$  は両瞳を含んだ近傍パターンから得られる類似度である。

## 4. 実験及び考察

### 4.1 実験 1：基本性能の評価

#### 4.1.1 実験条件

提案法の性能を定量評価するために、

- (a) 視線をカメラに向けたまま顔を真横に振る.
  - (b) 顔は正面を向いたまま視線のみ左右に動かす.
  - (c) 視線を顔向き方向に合わせて顔を左右に振る.
- という 3 種類の動画 (図 7) に対して瞳の追跡実験を行った.

本論文の冒頭で挙げたロボットのアイコンタクトやドライバーモニタリングなどのアプリケーションでは, 視線の左右の動きが基本となる場合が多いので, まずは, 上記のような動画を対象として評価を行った. フレームレート 30 [fps] で合計 9607 フレームを取得した. また, 顔とカメラの距離については, 被験者が正面を向いたときに, 瞳の直径が 60 [pixel] 程度になるように設定した.

この動画セットに対して, 3 種類の提案法と従来法との比較実験を行った. 3 種類の提案法の違いは表 1 に示すように相補的なサンプル推定 (Complementary Prediction) と複合ゆう度 (Compound Likelihood) の組合せによる. 各方法のアルゴリズムは以下のとおりである.

**提案法 1 (CP):** 通常のパーティクルフィルタの予測ステップに相補的なサンプル推定を導入した方法. ゆう度は 3.3 で述べた独立ゆう度を使用.

**提案法 2 (CL):** 通常のパーティクルフィルタの更新ステップに複合ゆう度を導入した方法.

**提案法 3 (CP+CL):** 通常のパーティクルフィルタの予測ステップと更新ステップにそれぞれ相互予測と複合ゆう度を導入した方法.

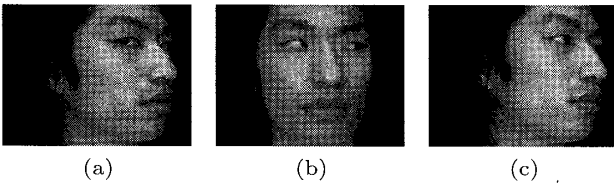


図 7 評価に用いた各動画像の先頭画像: (a) 顔を横に振る場合, (b) 視線を横に振る場合, (c) 顔と視線を横に振る場合

Fig. 7 Examples of the image sequence used in the evaluation.

表 1 提案法と比較法

Table 1 Proposed methods and conventional methods.

	相補的なサンプル推定 (CP)	複合ゆう度 (CL)
提案法 1 (CP)	○	×
提案法 2 (CL)	×	○
提案法 3 (CP+CL)	○	○
C-PF	×	×
Sep+PM	—	

**C-PF (Conventional Particle Filter):** 左右の瞳をそれぞれ独立に通常のパーティクルフィルタを適用する方法. ゆう度は 3.3 で述べた独立ゆう度を使用.

**Sep+PM (Separability and Pattern Matching):** 円形分離度フィルタとパターン照合を組み合わせた方法 [8]. 円形分離度フィルタにより検出された分離度の局所最大点を瞳の候補とし, 各候補に対してパターン照合を行うことで瞳検出する. この方法ではパーティクルフィルタは使われておらず, 各フレームごとに独立に瞳検出を行う.

パーティクルフィルタの各パラメータについては, システムノイズの共分散行列  $\Sigma = \text{diag}(30^2, 30^2, 1.5^2, 1.5^2, 0.1^2)$ , ここで  $\Sigma$  は対角行列で,  $\text{diag}$  は対角成分を表す. 各分散は予備実験によりそれぞれ最適な値に設定した. 独立ゆう度に基づいて抽出するサンプル数  $N_k = 0.01 \times N$ , 他方の瞳のサンプルから生成する事前サンプルの割合  $\mu = 0.5$  とした.

#### 4.1.2 実験結果

各方法の追跡精度を比較するために, 追跡した瞳座標と真の瞳座標との誤差を調べた. サンプル数  $N$  を変化させたときの, 誤差が 15pixel 以下のフレームの割合と平均誤差の変化を表 2 と表 3 に示す. 真の瞳座標は追跡対象の動画から目視により得た. 更に各方法の処理速度を表 4 に示す. 処理時間の計測は, Intel Core2 プロセッサ 6700 (2core, 2.66 GHz), MATLAB (R2009b) を用いて行った.

#### 4.1.3 考察

単純に左右瞳にパーティクルフィルタを適用した C-PF に対して, 提案法である CP と CL の追跡性能の方が優れていることが示された. このことから相互予測と複合ゆう度がそれぞれ有効に働いているといえる. また両者を組み込んだ CP+CL の性能が, CP と CL の性能を更に改善していることから, 相補的なサンプル推定と複合ゆう度を組合せが効果的であることが確認できた.

また, 処理速度に関して, 提案法は従来のパーティクルフィルタと同等程度の性能を得ることができた. この処理速度ではリアルタイム処理を行うには問題があるが, 提案法は通常のパーティクルフィルタと同様, 処理の並列化が容易であるため, GPU による並列処理などを行えば, リアルタイム処理も可能であると考えられる.



表 2 誤差が 15pixel 以下のフレームの割合 (左右の平均) [%]

Table 2 Success rate of tracking [%].

サンプル数	200	400	600	800	1000	2000	4000	6000
提案法 (CP+CL)	72.78	84.83	90.37	92.90	93.17	95.43	94.89	96.02
提案法 (CP)	61.65	81.39	86.07	88.65	90.48	89.89	94.67	93.01
提案法 (CL)	63.26	81.28	88.97	90.10	89.35	89.73	90.53	91.72
C-PF	49.33	61.54	70.68	77.25	78.32	80.58	86.01	84.02
Sep+PM	60.68							

表 3 平均誤差 (左右の平均) [pixel]

Table 3 Mean tracking error [pixel].

サンプル数	200	400	600	800	1000	2000	4000	6000
提案法 (CP+CL)	57.1	27.0	14.8	13.0	13.7	10.1	11.9	9.0
提案法 (CP)	84.9	32.6	29.2	19.8	17.5	21.2	10.9	13.4
提案法 (CL)	56.3	28.0	17.3	14.8	17.3	17.6	12.5	12.8
C-PF	89.6	62.9	44.6	29.0	29.5	24.7	13.6	17.5
Sep+PM	29.2							

表 4 処理速度 [fps]

Table 4 Processing speed [fps].

サンプル数	200	400	600	800	1000	2000	4000	6000
提案法 (CP+CL)	7.62	6.77	6.13	5.47	4.52	3.24	1.96	1.28
提案法 (CP)	7.80	6.96	6.21	5.55	5.06	3.26	1.98	1.29
提案法 (CL)	7.81	7.26	6.60	6.08	5.66	3.75	2.36	1.60
C-PF	8.11	7.38	6.79	6.09	5.77	3.77	2.41	1.60
Sep+PM	0.54							

## 4.2 実験 2: 一般的な動きに対する動作検証

### 4.2.1 実験条件

より一般的な瞳の動きに対応できるかを確認するために, 被験者が上下, 斜め方向を顔を向けたり, 首を傾げたりする状況, 及び被験者が片手で目を擦ることで片目が見えなくなる状況において, CP+CL に基づく追跡を行った. これらの動画はフレームレート 30fps で取得し, 画像サイズは  $640 \times 480$  pixel である. 先の実験と同様に, 被験者が正面向きのときに, 瞳の直径が 30pixel 程度になるように顔を撮影した. サンプル数  $N = 6000$ , 各パラメータは  $\Sigma = \text{diag}(15^2, 15^2, 1.5^2, 1.5^2, 0.1^2)$ ,  $N_k = 0.01 \times N$ ,  $\mu = 0.5$  とした.

### 4.2.2 実験結果

図 8 は, 5 人の被験者が様々な方向に顔を向けた動画に対する追跡結果を示している. 図 9 は片目が隠れた動画に対する追跡結果を示している. また, 様々な方向に向く動画に対する追跡に関して, 手動で与えた真の瞳座標と追跡結果の瞳の座標との誤差の平均は, 右瞳が 3.1 [pixel], 左瞳が 2.6 [pixel] であった. 片手で目をこする動画に対する追跡では, 左瞳の手動で与えた真の座標と追跡座標との誤差の平均は, 右目が見えている間では 2.1 [pixel], 右目が隠れている間では

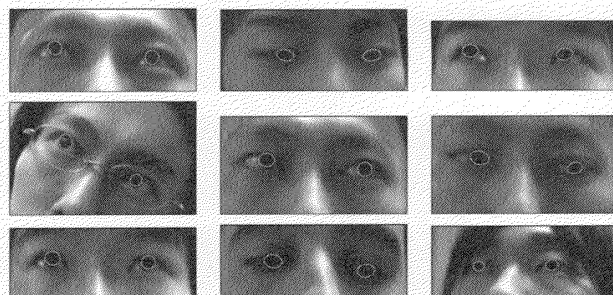


図 8 追跡結果の例: 両目の付近を拡大して表示している  
Fig. 8 Examples of the detected and tracked pupils.

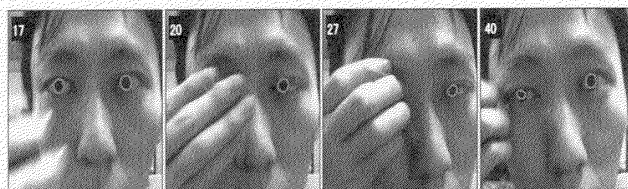


図 9 片目が隠れいされる動画に対する追跡の様子  
Fig. 9 Tracking results during other pupuil is hidded by a hand.

2.5 [pixel] であった.

### 4.2.3 考察

被験者が上下や斜め方向に顔を向けても, 両瞳を安定に追跡できていることが分かる. また, 瞳の追跡誤



差は瞳直径の 10 分の 1 程度に収まっており、提案法が瞳の左右の動きだけではなく、より一般的な動きにも対応できることが確認できたと考えられる。

片目を隠す動画に対しては、手が除かれた後、右目の追跡がスムーズに再開できている。追跡できている左目の状態から右目の状態が適切に予測され、次時刻の事前分布が正しい形に形成されているために、追跡のスムーズな再開が可能となったと考えられる。これは相補的な推定の副次的な効果の現れである。また、右目が隠れている間の左目の追跡性能は、見えている間と比較して平均 0.4 [pixel] 程度しか低下しておらず、片目の遮へいによるもう片方の目の追跡への影響はほとんどないといっている。これは、通常のパーティクルフィルタと同様に、サンプル生成を自身の前状態からも行うためと考えられる。二つの瞳を合わせて一つの状態とするような方法では、見えていない右目に引きずられて左目の追跡性能も大きく落ちてしまうと考えられる。

## 5. む す び

本論文では、両瞳の拘束関係を反映するために、連動する複数のパーティクルフィルタを提案し、これに基づいて顔向きや視線の変化に対してロバストな瞳追跡を実現した。提案法の基本アイデアは、パーティクルフィルタの予測ステップにおいて、前時刻における自分自身のサンプル状態に基づくサンプル（粒子）生成に加えて、前時刻の他方の瞳の状態に基づく生成を同時に行うことにある。他方からのサンプル生成を実現するために、一方の瞳形状と他方の瞳の形状・位置の対応関係を重回帰により学習した。

提案法の有効性を検証するために、被験者が顔向きや視線を大きく変化させる動画に対して瞳追跡を行った。その結果、提案法の追跡率は従来法を大きく上回り、平均誤差に関しても従来法の半分程度に抑えることができた。

今後の課題としては、GPU を用いた処理の高速化が挙げられる。提案法はサンプルごとに独立にゆ一度計算を行っているため、並列処理は効率的であると考えられる。瞳追跡では左右二つの状態ベクトル間の相補的な推定を行ったが、この枠組みは、人数数スポーツのプレイヤー追跡など、三つ以上の状態ベクトルに対しても適用できると考えられる。

## 文 献

[1] S.Y. Lee, Y.K. Ham, and R.H. Park, "Recognition of

- human front faces using knowledge-based feature extraction and neurofuzzy algorithm," *Pattern Recognition*, vol.29, no.11, pp.1863–1876, 1996.
- [2] O. Yamaguchi, K. Fukui, and K. Maeda, "Face recognition using temporal image sequence," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp.318–323, 1998.
- [3] 大野健彦, "視線を利用したウインドウ操作環境," *信学技報*. HIP1999-29, 1999.
- [4] 中野 明, 杉山和彦, 水野守倫, 山本 新, "画像による瞬目計測と居眠り検知への応用," *テレビ誌*, vol.50, no.12, pp.1949–1956, 1996.
- [5] K. Peng, L. Chen, S. Ruan, and G. Kukharev, "A robust algorithm for eye detection on gray intensity face without spectacles," *J. Comput. Sci. Technol.*, vol.5, no.8, pp.127–132, 2005.
- [6] A.L. Yuille, P.W. Hallinan, and D.S. Cohen, "Feature extraction from faces using deformable templates," *Int. J. Comput. Vis.*, vol.8, no.2, pp.99–111, 1992.
- [7] 坂本静生, 宮本陽子, 田島譲二, "顔画像中からの目の特徴点抽出," *信学論 (D-II)*, vol.J76-D-II, no.8, pp.1796–1804, Aug. 1993.
- [8] K. Fukui and O. Yamaguchi, "Facial feature point extraction method based on combination of shape extraction and pattern matching," *Systems and Computers in Japan*, vol.29, no.6, pp.49–58, 1998.
- [9] T. Kawaguchi, D. Hidaka, and M. Rizon, "Detection of eyes from human faces by hough transform and separability filter," *Proc. ICIP*, vol.I, pp.49–52, 2000.
- [10] Q. Chen, K. Kotani, F. Lee, and T. Ohmi, "An accurate eye detection method using elliptical separability filter and combined features," *Int. J. Comput. Sci. Netw. Secur.*, vol.9, no.8, pp.5–72, 2009.
- [11] K. Hayashi, M. Hashimoto, and K. Sumi, "Head direction estimation with robust and accurate tracker to facial features," *IEICE Trans. Inf. & Syst.*, vol.J84-D-II, no.8, pp.1762–1771, Aug. 2001.
- [12] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, 2004.
- [13] 加藤丈和, "パーティクルフィルタとその実装法," *情報処理学会研究報告*. CVIM, pp.161–168, 2007.
- [14] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proc.-F*, vol.140, no.2, pp.107–113, 1993.
- [15] H. Wu, Y. Kitagawa, T. Wada, T. Kato, and Q. Chen, "Tracking iris contour with a 3D eye-model for gaze estimation," *Proc. ACCV'07*, pp.688–697, 2007.
- [16] 大谷悠祐, 福井和広, "眼球の位置と姿勢を考慮した顔向き変化に頑健な瞳追跡," *信学技報*, PRMU2009-136, 2009.
- [17] 満上育久, 角所 考, 美濃導彦, "非剛体物体の形状変化と運動変化の主成分分析に基づくパーティクルフィルタの効率化," *信学論 (D)*, vol.J92-D, no.8, pp.1270–1278, Aug. 2009.

- [18] 菅野裕介, 佐藤洋一, “顔変形をとまなう 3 次元頭部姿勢の単眼推定,” 情報処理学会論文誌コンピュータビジョンとイメージメディア, vol.1, no.2, pp.41–49, 2008.
- [19] K. Okuma, A. Taleghani, N.D. Freitas, J.J. Little, and D.G. Lowe, “A boosted particle filter: Multitarget detection and tracking,” Proc. European Conference on Computer Vision (ECCV2004), vol.3021 of LNCS, pp.28–39, 2004.
- [20] 石井健一郎, 上田修功, 前田英作, 村瀬 洋, わかりやすいパターン認識, オーム社, 1998.
- [21] Y. Cheng, “Mean shift, mode seeking, and clustering,” IEEE Trans. Pattern. Anal. Mach. Intell., vol.17, no.5, pp.790–799, 1995.
- [22] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vis., vol.60, no.2, pp.91–110, 2004.

(平成 23 年 10 月 31 日受付, 24 年 2 月 19 日再受付)



大谷 悠祐

2009 筑波大・第三・情報学類卒。2011 同大学院システム情報工学研究科博士前期課程在学。同年セイコーエプソン（株）入社、現在に至る。



福井 和広 （正員）

1986 九大・工・機械卒。1988 同大学院工学研究科了。同年（株）東芝入社。同社研究開発センター主任研究員を経て、現在、筑波大学大学院システム情報工学研究科准教授。パターン認識・コンピュータビジョンの理論と応用に関する研究に従事。

工博。2002 本会論文賞, 2006 本会 ISS 活動功労賞受賞。情報処理学会, IEEE 各会員。