パーティクルフィルタを用いた頭部姿勢の実時間推定

岡 兼司　佐藤 洋一　中西 泰人　小池 英樹

1　はじめに

ユーザの意図の推定にもとづくインタフェースは Attentive User Interface（AUI）と呼ばれ，今後のインタフェースに関する研究の方向性の一つとして注目を集めている [5].

ユーザの意図を推定するための手がかりとして，しばしばユーザの注視点に関する情報が利用される．そこで我々は，ユーザの注視点を実時間で得ることを目的として，コンピュータビジョンによりユーザの頭部の 3 次元的な姿勢を実時間推定するための手法について開発する．

一般に，インタフェースに利用するためには，入力画像に対象物体の遮蔽やノイズが含まれるような状況においても安定した推定が可能でなければならない．近年，この問題を解決可能な技術としてパーティクルフィルタ [2] が注目されており，頭部姿勢の推定にも利用されてきた [4, 1].

その一方で，実際のユーザの動きを考慮すると，次の 2 点も重要な要件として挙げられる．ユーザの突発的な動作にも対象を見失うことなく追従可能であることと，ユーザがある点を注視しているときには十分に高い精度で頭部姿勢を推定可能なことである．しかし，過去のパーティクルフィルタによる頭部姿勢推定手法では，これらを十分に満たしているとは言い難い．

そこで本研究では，突発的な動作への対応と十分な推定精度の両方を実現した実時間頭部姿勢推定手法を開発する．特に，パーティクルフィルタで使用されるサンプルの拡散を適応的に制御することにより，この問題の解決を目指す．

2　パーティクルフィルタ

パーティクルフィルタは，各時刻における状態ベクトルを逐次的に推定するための手法の一つとして，Isard と Blake がコンピュータビジョンに適用した技術である [2].

ここで，パーティクルフィルタについて簡単に説明する．t 番目の画像フレームにおける状態ベクトルを $\boldsymbol{x}_t$，観測ベクトルを $\boldsymbol{y}_t$ とする．そして，観測ベクトルを $\boldsymbol{y}_t$ の履歴を $\boldsymbol{Y}_t = \{\boldsymbol{y}_1 \dots \boldsymbol{y}_t\}$ とする．一般に，状態の推定とは，状態ベクトル $\boldsymbol{x}_t$ の確率密度関数 $p(\boldsymbol{x}_t|\boldsymbol{Y}_t)$ を推定する問題として定式化される．パーティクルフィルタの場合には，確率密度 $p(\boldsymbol{x}_t|\boldsymbol{Y}_t)$ は多数の離散的なサンプルのセット $\{(\boldsymbol{s}_t^{(i)}; \pi_t^{(i)})\}$ ($i = 1 \dots N$) によって表現される．ただし，$\boldsymbol{s}_t^{(i)}$ は状態空間 $\boldsymbol{x}_t$ 中の離散的なランダムサンプルであり，このサンプルは重み $\pi_t^{(i)}$ に比例する確率を持っている．また，N はサンプルの総数である．それゆえ，$p(\boldsymbol{x}_t|\boldsymbol{Y}_t)$ は任意の非ガウシアンな確率密度関数を近似することが可能となっている．各サンプルは次の予測ステップと観測ステップを経て更新される．

1. 予測：まず，サンプルセット $\{(\boldsymbol{s}_{t-1}^{(i)}; \pi_{t-1}^{(i)})\}$ から，重み $\pi_{t-1}^{(i)}$ に比例する確率にもとづいて，ベースとなるサンプル $\boldsymbol{s}_{t-1}^{(\xi)}$ を選択する．その後，動きモデル $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1} = \boldsymbol{s}_{t-1}^{(\xi)})$ から新たなサンプル $\boldsymbol{s}_t^{(i)}$ を生成する．

2. 観測：現時刻での観測ベクトル $\boldsymbol{y}_t$ が与えられたもとで，観測密度 $p(\boldsymbol{y}_t|\boldsymbol{x}_t = \boldsymbol{s}_t^{(i)})$ にもとづいて，各サンプル $\boldsymbol{s}_t^{(i)}$ の尤度を評価し，その尤度に比例する重み $\pi_t^{(i)}$ を決定する．

最後に，すべてのサンプルを統合することにより，現時刻での推定値が計算される．

3　提案手法

本節では，複数台のカメラからの入力画像をもとに実時間で頭部姿勢を推定するための手法について説明する．なお，本稿では 2 台のカメラの場合を説

明するが，この台数は理論的な拡張なしに増設することが可能である．

本手法は，頭部の 3 次元モデルを自動的に獲得する初期化部と，獲得された 3 次元モデルと入力画像列からパーティクルフィルタを利用して逐次的に頭部姿勢推定を行う追跡部から構成される．

3.1 初期化部

提案手法では K 個の特徴点を持つ 3 次元モデルを利用する．本稿では，$K = 10$ に固定されており，各点は両目の両端と口の両端，両鼻孔，そして両眉の内側の端点に対応している．

各特徴点はモデル座標系内での 3 次元位置 $\boldsymbol{M}_k(k = 1 \ldots K)$ を持っている．なお，モデル座標系は頭部に固定されている．この位置 $\boldsymbol{M}_k$ はオムロンで開発された OKAO ビジョンライブラリ [3] を用いて自動的に決定される．さらに，各特徴点は左右のカメラに対する画像テンプレート $T_{L,k}$，$T_{R,k}$（例えば，目の端点の画像）も保持している．

3.2 追跡部の概要

初期化部で計算された 3 次元モデルと入力画像列から，パーティクルフィルタを利用して，逐次的に頭部姿勢を推定する．ここでは，t 番目の画像フレームにおける頭部姿勢を 6 次元ベクトル $\boldsymbol{p}_t = (x_t, y_t, z_t, \phi_t, \theta_t, \psi_t)^{\mathrm{T}}$ により表現する．なお，$(x_t, y_t, z_t)^{\mathrm{T}}$ と $(\phi_t, \theta_t, \psi_t)^{\mathrm{T}}$ は，それぞれ世界座標系からモデル座標系への並進と回転を示す．

本手法で利用するパーティクルフィルタの各サンプルの状態ベクトル $\boldsymbol{s}_t^{(i)}$ は 12 次元ベクトル $(\boldsymbol{p}_t^{(i)\mathrm{T}}, \boldsymbol{v}_t^{(i)\mathrm{T}})^{\mathrm{T}}$ である．なお，$\boldsymbol{v}_t^{(i)}$ は $\boldsymbol{p}_t^{(i)}$ の速度を表す 6 次元ベクトルである．

本手法では，以下の 3 ステップにより頭部姿勢の推定を行う．

1. 第 1 段階推定として，状態ベクトルの確率密度 $\{(\boldsymbol{s}_t^{(i)}; \pi_t^{(i)})\}\,(i = 1 \ldots N)$ を粗く推定し，その結果から頭部姿勢 $\hat{\boldsymbol{p}}_t$ を推定する．

2. 第 2 段階推定として，再サンプリングを通して細かく確率密度 $\{(\boldsymbol{s}_t^{\prime(j)}; \pi_t^{\prime(j)})\}\,(j = 1 \ldots N)$ を推定し，その結果から頭部姿勢 $\hat{\boldsymbol{p}}_t'$ を推定する．

3. $\hat{\boldsymbol{p}}_t$ と $\hat{\boldsymbol{p}}_t'$ のうち，より最適な方をその時刻における頭部姿勢推定結果として出力する．

以下に，各段階の具体的な処理内容について説明する．

3.3 第 1 段階推定

本段階では，直前の画像フレームにおけるサンプルセット $\{(\boldsymbol{s}_{t-1}^{(i)}; \pi_{t-1}^{(i)})\}\,(i = 1 \ldots N)$ から現在の状態ベクトルを粗く推定する．

最初に，サンプルセット $\{(\boldsymbol{s}_{t-1}^{(i)}; \pi_{t-1}^{(i)})\}$ から，重み $\pi_{t-1}^{(i)}$ にもとづいてベースとなるサンプル $\boldsymbol{s}_{t-1}^{(\xi)}$ を選択する．そして，選ばれたサンプル $\boldsymbol{s}_{t-1}^{(\xi)}$ を次式にしたがって $\boldsymbol{s}_t^{(i)}$ にシフトする．このとき，隣り合う画像フレーム間では等速直線運動を行うことを仮定している．

$$\boldsymbol{s}_t^{(i)} = \begin{bmatrix} \boldsymbol{I}_{6\times6} & \tau\boldsymbol{I}_{6\times6} \\ \emptyset & \boldsymbol{I}_{6\times6} \end{bmatrix} \boldsymbol{s}_{t-1}^{(\xi)} + \begin{bmatrix} \boldsymbol{\omega}_t^{(i)} \\ \emptyset \end{bmatrix} \quad (1)$$

ここで，$\boldsymbol{I}_{6\times6}$ は 6 次元単位行列，τ は画像フレーム間隔，$\boldsymbol{\omega}_t^{(i)}$ は頭部姿勢 $\boldsymbol{p}_{t-1}^{(\xi)}$ に加えられるシステム雑音であり，白色雑音を利用している．また，速度成分 $\boldsymbol{v}_t^{(i)}$ はこの段階では更新されず，3.5 節で述べる処理で更新されることとなる．

本手法では，システム雑音 $\boldsymbol{\omega}_t^{(i)}$ を適応的に制御することにより，ユーザの突発的な動きと注視時の精度との両方を向上させることを目指す．ユーザが突発的な動きをした場合には等速直線運動の仮定を著しく外れてしまうことが多く，このときにはシステム雑音を大きくする必要がある．一方で，ユーザが注視している場合にあまりシステム雑音を大きくすると，推定精度が劣化することが経験的に確認されている．そこで，$\boldsymbol{\omega}_t^{(i)}$ の取り得る範囲を姿勢運動の速度に応じて適応的に変化させることとする．すなわち，運動速度が速いほど，$\boldsymbol{\omega}_t^{(i)}$ が広範囲な値を取るようにする．このような適応的制御による性能向上については，後で述べる評価実験を通して確認する．

新たに $\boldsymbol{s}_t^{(i)}$ が計算されると，次に各サンプルの重み $\pi_t^{(i)}$ が現在の観測にもとづいて計算される．各サンプル $\boldsymbol{s}_t^{(i)}$ に対応する姿勢 $\boldsymbol{p}_t^{(i)}$ のもとで，モデルの各特徴点 $\boldsymbol{M}_k(k = 1 \ldots K)$ は，左カメラの画像平面と右カメラの画像平面上に 2 次元座標 $\boldsymbol{m}_{L,t,k}^{(i)}$，$\boldsymbol{m}_{R,t,k}^{(i)}$

として投影されるものとする．この投影作業は関数 $\mathcal{P}_h(h \in L, R)$ により，$\boldsymbol{m}_{h,t,k}^{(i)} = \mathcal{P}_h(\boldsymbol{p}_t^{(i)}, \boldsymbol{M}_k)$ と表現される．

さらに，投影点 $\boldsymbol{m}_{h,t,k}^{(i)}$ の周辺領域と対応する画像テンプレート $T_{h,k}$ との正規化相関を $\mathcal{N}_h(T_{h,k}, \boldsymbol{m}_{h,t,k}^{(i)})(h \in L, R)$ と定義する．このとき，正規化相関 $\mathcal{N}_h$ の出力値は-1から1の間の値を取る．

これらの定義を利用して，各サンプルに対するスコア $c_t^{(i)}$（$-2K \leq c_t^{(i)} \leq 2K$）が計算される．その後，重み $\pi_t^{(i)}$ がスコア $c_t^{(i)}$ をもとにガウス関数を使って計算され，この重み $\pi_t^{(i)}$ からこの段階での姿勢 $\hat{\boldsymbol{p}}_t$ が計算される．

$$c_t^{(i)} = \sum_{k=1}^{K} \sum_{h \in \{L,R\}} \mathcal{N}_h(T_{h,k}, \mathcal{P}_h(\boldsymbol{p}_t^{(i)}, \boldsymbol{M}_k)) \quad (2)$$

$$\pi_t^{(i)} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(2K - c_t^{(i)}\right)^2}{2\sigma^2}} \quad (3)$$

$$\hat{\boldsymbol{p}}_t = \frac{\sum_{i=1}^{N} \boldsymbol{p}_t^{(i)} \pi_t^{(i)}}{\sum_{i=1}^{N} \pi_t^{(i)}} \quad (4)$$

3.4 第2段階推定

この段階では，再サンプリングを通して推定精度を向上させることを目指す．

まず，以下の式にしたがって，第1段階で推定された姿勢 $\hat{\boldsymbol{p}}_t$ の周辺に新たなサンプルを発生させる．

$$\boldsymbol{p}_t^{\prime(j)} = \hat{\boldsymbol{p}}_t + \boldsymbol{\mu}_t^{(j)} \quad (5)$$

このとき，$\boldsymbol{p}_t^{\prime(j)}$ は新たに発生させたサンプルの姿勢成分であり，$\boldsymbol{\mu}_t^{(j)}$ が零ベクトルを平均値とする6次元のガウス雑音である．

その後，第1段階と同様の方法で，姿勢 $\boldsymbol{p}_t^{\prime(j)}$ に対応する重み $\pi_t^{\prime(j)}$ を計算し，第2段階での姿勢推定値を計算する．

$$c_t^{\prime(j)} = \sum_{k=1}^{K} \sum_{h \in \{L,R\}} \mathcal{N}_h(T_{h,k}, \mathcal{P}_h(\boldsymbol{p}_t^{\prime(j)}, \boldsymbol{M}_k)) \quad (6)$$

$$\pi_t^{\prime(j)} = \frac{1}{\sqrt{2\pi}\sigma'} e^{-\frac{\left(2K - c_t^{\prime(j)}\right)^2}{2\sigma'^2}} \quad (7)$$

$$\hat{\boldsymbol{p}}_t' = \frac{\sum_{j=1}^{N} \boldsymbol{p}_t^{\prime(j)} \pi_t^{\prime(j)}}{\sum_{j=1}^{N} \pi_t^{\prime(j)}} \quad (8)$$

3.5 推定値の最終決定

最後に，第1段階と第2段階の結果から，以下のようにして，最終的な推定値を決定する．

$$c_t = \sum_{k=1}^{K} \sum_{h \in \{L,R\}} \mathcal{N}_h(T_{h,k}, \mathcal{P}_h(\hat{\boldsymbol{p}}_t, \boldsymbol{M}_k)) \quad (9)$$

$$c_t' = \sum_{k=1}^{K} \sum_{h \in \{L,R\}} \mathcal{N}_h(T_{h,k}, \mathcal{P}_h(\hat{\boldsymbol{p}}_t', \boldsymbol{M}_k)) \quad (10)$$

$$\boldsymbol{p}_t = \begin{cases} \hat{\boldsymbol{p}}_t' & \text{if} \quad c_t' \geq c_t \\ \hat{\boldsymbol{p}}_t & \text{else} \end{cases} \quad (11)$$

さらに，各サンプルの速度成分 $\boldsymbol{v}_t^{(i)}$ を以下のように計算する．

$$\boldsymbol{v}_t = \frac{\boldsymbol{p}_t - \boldsymbol{p}_{t-1}}{\tau} \quad (12)$$

$$\boldsymbol{v}_t^{(i)} = (1 - \lambda_t)\left(\boldsymbol{v}_{t-1}^{(i)} + \boldsymbol{\nu}_t^{(i)}\right) + \lambda_t \boldsymbol{v}_t \quad (13)$$

このとき，システム雑音 $\boldsymbol{\nu}_t^{(i)}$ は零ベクトルを平均値とする6次元のガウス雑音である．また，λ_t は状態空間内でのサンプルの収束度合にもとづいて決定される0と1の間の値であり，具体的にはサンプルセットの加重分散から計算される．

4 評価実験

本節では，提案手法に関する評価実験の概要と結果について述べる．実験ではLinuxをOSとするPCを利用した．PCにはCPUとしてIntel Pentium IV 2.8GHzが搭載されている．入力画像は2台のモノクロカメラとアナログキャプチャボードにより取り込まれ，その解像度は 640×240（ノンインタレース）である．また，画像テンプレートのサイズは 12×6，パーティクルフィルタのサンプル数は1000である．このシステムは毎秒30フレームで動作する．

入力画像シーケンスとしては2種類用意した．シーケンス#1では，ユーザは比較的ゆっくり動作し，時折注視のために静止している．一方，シーケンス#2では，図1に示すように，ユーザは比較的高速に動作している．いずれのシーケンスも10秒（300フレーム）分のデータを含んでいる．また，各シーケンスに対する姿勢推定の正解値は，磁気センサであるPolhemus社のFASTRAKを利用して計測された．

	x	y	z	roll	yaw	pitch
#1(slow)	1.42	2.25	3.09	0.58	1.94	2.43
#2(fast)	5.06	6.41	5.34	0.79	2.52	3.43

表 1: 適応的制御を適用した場合の最小二乗誤差の平方根（x，y，z[mm] and roll，yaw，pitch[degree]）

	x	y	z	roll	yaw	pitch
#1(slow)	3.39	2.68	4.90	0.71	3.87	2.52
#2(fast)	7.29	6.89	6.36	1.74	6.40	5.73

表 2: 適応的制御を適用しない場合の最小二乗誤差の平方根（x，y，z[mm] and roll，yaw，pitch[degree]）

図 1 はシーケンス#2 の実験結果である．太い実線（*Est.*）が本手法による推定値，破線（*GT*）が正解値である．また，細い実線（*Const.*）はシステム雑音の適応的制御を行わなかった場合の推定値である．この図からわかるように，適応的制御を行った本手法では，ユーザが高速に動いた場合でも，高い追従性を維持しながら追跡することが可能である．

また，表 1 はシステム雑音の適応的制御を適用した場合の最小二乗誤差の平方根，表 2 はそれを適用しない場合の最小二乗誤差の平方根である．この表からも，システム雑音の適応的制御が精度の上で有効に作用していることがわかる．

さらに，図 2 は追跡結果の画像である．1 行目と 2 行目は入力画像シーケンス#2 に含まれる画像である．また，3 行目は様々な状況下（眼鏡の着用，顔の一部の遮蔽，照明・背景の動的変化）での追跡結果である．このような状況下でも，本手法では安定に頭部姿勢を推定することが可能である．

5 おわりに

本手法では，複数のカメラからの入力画像をもとに実時間で頭部の 3 次元的な姿勢を推定するための手法について提案した．特に，システム雑音の適応的制御という要素を取り入れたパーティクルフィルタを利用することにより，ユーザが突発的に動作する場合やユーザが注視している場合の追跡・推定性能を向上させることに成功した．

今後は，頭部モデルとして変形可能なモデルを導入することにより，さらなる精度の向上を計画している．

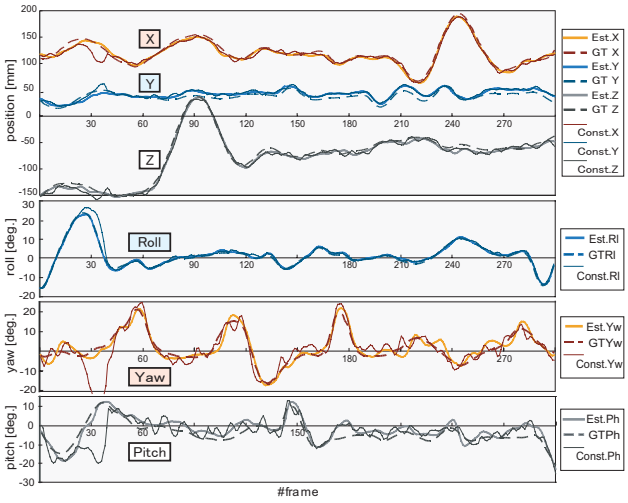

図 1: シーケンス#2 の推定結果

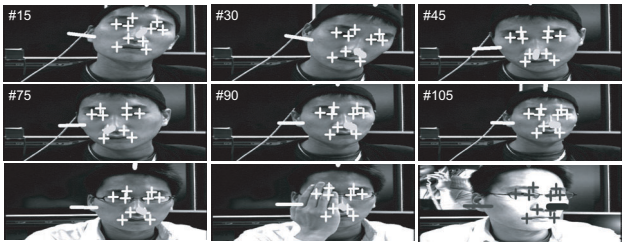

図 2: 推定結果画像の一例

謝辞

本研究の一部に（株）オムロンの顔検出・顔器官検出技術を利用している．

参考文献

[1] B. Braathen, et al., “An approach to automatic recognition of spontaneous facial actions,” *Proc. FG 2002*, pp. 360-365, 2002.

[2] M. Isard and A. Blake, “Condensation– conditional density propagation for visual tracking,” *IJCV*, Vol. 29, No. 1, pp. 5-28, 1998.

[3] S. Lao, et al., “A fast 360-degree rotation invariant face detection system,” *ICCV 2003*, Demo Session, 2003.

[4] J. Sherrah and S. Gong, “Fusion of perceptual cues for robust tracking of head pose and position,” *Pattern Recognition*, Vol. 34, No. 8, 2001.

[5] R. Vertegaal, “Attentive user interfaces,” *CACM*, Vol. 46, No. 3, pp. 30-33, 2003.