

分散モンテカルロ追跡法

－ ウェアラブルアクティブビジョンシステムによる実時間人物追跡 －

加藤 丈和 蔵田 武志 坂上 勝彦

産業技術総合研究所 知能システム研究部門
〒305-8568 茨城県 つくば市 梅園 1-1-1 中央第 2

E-mail : t.kato@ieee.org

あ ら ま し

本稿では、ウェアラブルアクティブビジョンシステムによって、実時間かつ高精度な人物追跡を実現する手法として、分散モンテカルロ追跡法を提案する。分散モンテカルロ追跡法は、クライアント/サーバ分散型の追跡アルゴリズムであり、クライアント上では大まかな人物頭部追跡によってアクティブカメラへの高速なフィードバックを実現し、サーバ上ではクライアントの結果を反映しつつ詳細な人物頭部位置、姿勢を推定する。試作したウェアラブルアクティブビジョンシステムに提案手法を実装し、本手法による追跡性能を確認した。

キーワード ウェアラブルシステム, 分散処理, アクティブビジョン, 人物追跡, ConDensation アルゴリズム

Distributed Monte Carlo Tracking

－ Real-time Human Tracking by Wearable Active Vision Systems －

Takekazu Kato, Takeshi Kurata, Katsuhiko Sakaue

Intelligent Systems Institute,
National Institute of Advanced Industrial Science and Technology (AIST),
1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568, Japan

E-mail : t.kato@ieee.org

Abstract

In this paper, we propose distributed Monte Carlo tracking to realize accurate and real-time human tracking by wearable active vision systems. This method is a client-server distributed algorithm for human-head tracking. The client feeds back the tracked head positions to the active camera with minimum delay by the fast and coarse tracking. The server accurately estimates head posture according to the tracking results obtained by the client. We implemented the distributed Monte Carlo tracking on the wearable active vision system, and confirmed the tracking capability in experiments.

Keywords wearable system, distributed processing, active vision, human tracking, ConDensation algorithm

1 まえがき

近年、携帯型コンピュータや携帯型ディスプレイ、デジタルカメラを始めとする各種センサなどの小型化、高性能化にともなって、ウェアラブルコンピューティングが現実的なものとなってきている。また、無線ネットワークの広域・広帯域化によって、いつでも、どこでも、映像情報などの大容量データを送受信することが可能になりつつあり、このような環境に適したウェアラブルアプリケーションの創出が期待されている。

ウェアラブルシステムの利点の1つは、ユーザがシステムを常に身につけることによって、システムがユーザと同じ経験を共有できることである。つまり、ウェアラブルシステムは、ユーザの状況を理解し、適切なフィードバックをユーザに提示することで、適応的にユーザを支援する潜在能力をもつといえる。

このようなユーザの状況理解に基づくウェアラブルアプリケーションを実現するためには、実世界データを収集し、そこからユーザの状況を理解することが重要である。コンピュータビジョンは、それらを実現する手段として有用であり、ウェアラブルシステムへの応用が期待されている [1, 7, 11]。

我々は、ウェアラブルアクティブカメラ [8] を備えた着用型アシスタントである *VizWear-Active* の研究開発を行っている [5, 6]。*VizWear-Active* では、ユーザが出逢った人物に関する情報を自律的に収集し、その状況を把握することを目的としている。ユーザにとって、出逢った人物に関連する情報は最も興味のある事柄の1つであり、これらの情報を自律的に収集することは記憶補助 [2, 9] やセキュリティなどへの応用が期待される。

本稿では、*VizWear-Active* による人物情報の収集のために、ウェアラブルアクティブビジョンシステムによって、高速かつ高精度な人物追跡を実現する手法として、分散モンテカルロ追跡法を提案する。分散モンテカルロ追跡法は、クライアント/サーバ型の分散処理によって、アクティブカメラ制御に必要な高速で遅延の少ない追跡と、状況把握や個人識別に必要な詳細な人物位置、姿勢の推定を同時に実現する。

2 *VizWear-Active*

現在、多くのウェアラブルシステムでは、頭部など体の一部に固定されたカメラを使用している。このようなカ

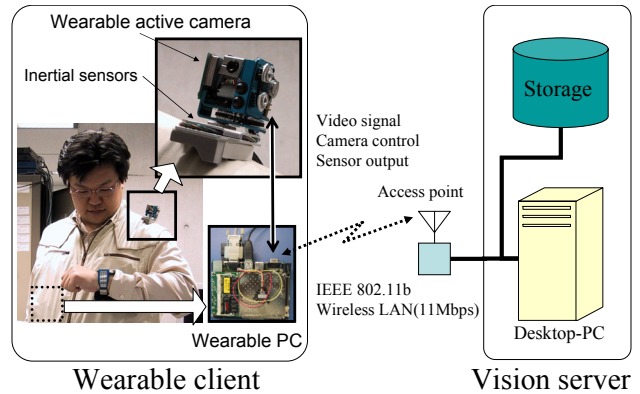


図 1: *VizWear-Active* の試作システム

メラでは観測範囲がユーザの動きに追従するため、ユーザが注目する対象を自然に観測できるという利点を持つ反面、ユーザが注目していない情報を見逃してしまうという欠点がある。*VizWear-Active* においては、このような固定カメラでは、ユーザが注視していない人物の情報を収集することができない。そこで、アクティブカメラを用いて能動的に観測範囲を制御しながら観測を行うことで、ユーザの意図とは独立した情報収集を実現する。

図 1 に *VizWear-Active* の試作システムを示す。本システムで用いるアクティブカメラは、ウェアラブル用に製作したパン・チルト回転雲台付きのカメラであり、軽量 (100g) かつ小型 (40mm × 40mm × 40mm) で、パン方向に約 270°、チルト方向に約 90° の範囲でカメラヘッドを回転させることができる。また、雲台の土台部分にモーションセンサ (InterSense InterTrax²) が固定され、ユーザの動作にともなうカメラの動き (回転角度) を計測することができる。カメラとモーションセンサは、ジャケットに組み込まれたウェアラブル PC (Intel mobile PentiumIII 500MHz) に接続され、カメラで得られた画像やセンサのデータを取り込んで処理することができ、また、ウェアラブル PC からアクティブカメラを制御することができる。本システムでは、これらのユーザが身につけるシステムをウェアラブルクライアントと呼ぶ。

一般にウェアラブル PC は、小型化や省電力化のために十分な計算リソースを持つことができないため、計算コストの高いコンピュータビジョンのタスクを実時間で処理することは困難である。本システムでは、このようなタスクを環境側に設置したビジョンサーバに分散して処理することができる。試作システムではビジョンサーバとしてデスクトップ PC (Intel dual Xeon 1.7GHz) を用い、ウェアラブルクライアントとビジョンサーバの間

は無線 LAN によって接続する．

現在の試作システムでは無線 LAN を利用しているため、アクセスポイントのある場所でのみサーバとの通信が可能であるが、将来、第 4 世代 (4G) 移動通信網や、公衆無線 LAN などの無線ブロードバンドネットワークのインフラが整備されると、ウェアラブルクライアントがどこにいてもサーバと通信することが可能となり、このようなクライアント/サーバによるシステム構成が現実的なものとなると考えられる．

3 分散モンテカルロ追跡法

VizWear-Active のようなアクティブカメラを持つシステムで人物追跡を行う場合、対象人物を見失わないように常にアクティブカメラを制御する必要があるため、追跡結果をアクティブカメラへ高速にフィードバックすることが重要となる．また、人物追跡の結果に基づいて対象人物の状況把握や個人識別を行うためには、対象人物の位置、姿勢を詳細に推定することが重要となる．

一般に、画像から人物位置、姿勢を詳細に推定するタスクは計算コストが高く、ウェアラブル PC 単体によって実時間で処理するのは困難である．また、入力画像をサーバに転送し、サーバ上で人物位置、姿勢を推定する場合、無線 LAN を経由したクライアント/サーバ間の通信に遅延が生じるため、推定結果をアクティブカメラへ高速にフィードバックすることが難しくなる．また、無線 LAN では通信障害や基地局の切り換えによって、一時的に極端に転送レートが低下したり通信不能となる可能性があり、そのような場合にはアクティブカメラへのフィードバックができなくなり、対象人物を見失ってしまう危険性がある．

本稿では、これらの問題を解決する手法として、分散モンテカルロ追跡法を提案する．分散モンテカルロ追跡法は Isard ら [4] によって提案された ConDensation アルゴリズムを、クライアント/サーバ型の分散環境へと拡張した手法であり、ウェアラブルシステムにおいてアクティブカメラへの高速なフィードバックと、人物位置、姿勢の詳細な推定を同時に実現する．

3.1 ConDensation アルゴリズム

文献 [4] で述べられている ConDensation アルゴリズムについて簡単に説明する．ある時刻 t における追跡対象の状態を \mathbf{X}_t 、画像から得られる観測結果を \mathbf{Z}_t とおく．また、時刻 t までに得られる観測結果を

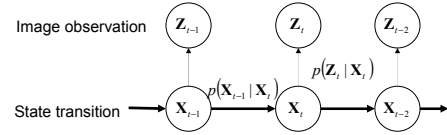


図 2: ConDensation アルゴリズム

$\mathbf{Z}_t = (\mathbf{Z}_1, \dots, \mathbf{Z}_t)$ とする．このとき、対象追跡問題は時刻 t までの観測結果 \mathbf{Z}_t が得られたときに状態 \mathbf{X}_t の確率密度 $p(\mathbf{X}_t|\mathbf{Z}_t)$ を推定する問題と考えることができる．

ここで、追跡対象の状態遷移について図 2 に示すようなマルコフ過程が成り立つと仮定したとき、時刻 $t-1$ における追跡対象の確率密度 $p(\mathbf{X}_{t-1}|\mathbf{Z}_{t-1})$ と、時刻 $t-1$ から t までの間の状態推移確率 $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ が与えられると、時刻 t における事前確率 $p(\mathbf{X}_t|\mathbf{Z}_{t-1})$ を次式のように求めることができる．

$$p(\mathbf{X}_t|\mathbf{Z}_{t-1}) = \int p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Z}_{t-1})d\mathbf{X}_{t-1} \quad (1)$$

このとき、時刻 t における確率密度 $p(\mathbf{X}_t|\mathbf{Z}_t)$ は、ベイズの法則に従って次のように求めることができる．

$$p(\mathbf{X}_t|\mathbf{Z}_t) \propto p(\mathbf{Z}_t|\mathbf{X}_t)p(\mathbf{X}_t|\mathbf{Z}_{t-1}) \quad (2)$$

すなわち、各時刻において尤度 $p(\mathbf{Z}_t|\mathbf{X}_t)$ を画像から推定することで、状態 \mathbf{X}_t の確率密度 $p(\mathbf{X}_t|\mathbf{Z}_t)$ を得ることができる．

全ての \mathbf{X}_t に対して尤度を計算することは現実的ではないので、確率密度 $p(\mathbf{X}_t|\mathbf{Z}_t)$ をランダムサンプリングによって推定する．ある時刻 t における確率密度 $p(\mathbf{X}_t|\mathbf{Z}_t)$ を状態 \mathbf{X}_t のサンプルの集合 $\{\mathbf{s}_t^{(1)}, \dots, \mathbf{s}_t^{(N)}\}$ と各サンプルの評価値 $\{\pi_t^{(1)}, \dots, \pi_t^{(N)}\}$ の組によって離散的に表現する．ここで、あるサンプル $\mathbf{s}_t^{(i)}$ の評価値 $\pi_t^{(i)}$ は $\pi_t^{(i)} \propto p(\mathbf{Z}_t|\mathbf{X}_t = \mathbf{s}_t^{(i)})$ となるように画像から推定する．

ConDensation アルゴリズムは、次のような手順によって、直前の時刻の確率密度に従ったサンプル集合の生成と、各サンプルの評価値の推定を繰り返すことで追跡を行う．

1. フレーム $t-1$ におけるサンプル集合 $\{\mathbf{s}_{t-1}^{(1)}, \dots, \mathbf{s}_{t-1}^{(N)}\}$ から、各サンプルの評価値 $\{\pi_{t-1}^{(1)}, \dots, \pi_{t-1}^{(N)}\}$ の比に従って N 個のサンプルを選択する．これを $\{\mathbf{s}'_{t-1}{}^{(1)}, \dots, \mathbf{s}'_{t-1}{}^{(N)}\}$ と置く．
2. 選択された N 個のサンプルそれぞれについて、推移確率 $p(\mathbf{X}_t|\mathbf{X}_{t-1} = \mathbf{s}'_{t-1}{}^{(i)})$ に従って次の状態を選択し、時刻 t におけるサンプル $\mathbf{s}_t^{(i)}$ とする．

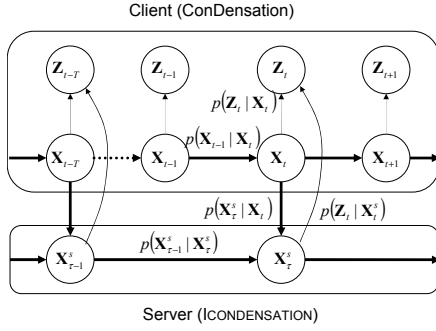


図 3: 分散モンテカルロ追跡法

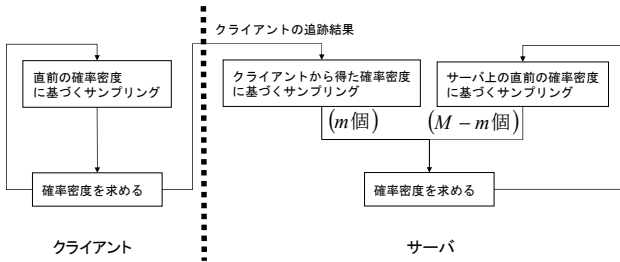


図 4: 分散モンテカルロ追跡法の概要

3. 各サンプル $s_t^{(i)}$ の評価値 $\pi_t^{(i)}$ を推定する .
4. $t = t + 1$ として 1. へ

ただし、初期状態における事前確率 $p(X_0)$ 、及びフレーム間の推移確率 $p(X_t|X_{t-1})$ はあらかじめ与えられているものとする .

3.2 分散モンテカルロ追跡法のアルゴリズム

ConDensation アルゴリズムでは、常に複数の解の候補を保持しながら追跡することでノイズや環境変動に対してロバストな追跡を実現することが可能であるが、ランダムサンプリングによって追跡対象の確率密度を求めため、追跡対象の状態を詳細に推定するためにはサンプリング回数を増す必要があり、計算コストが高くなってしまふ .そこで、分散モンテカルロ追跡法では、クライアントとサーバに分散して ConDensation アルゴリズムを実行することで、クライアント上では必要最小限のサンプリング回数による、粗く高速な追跡、サーバ上ではサーバではクライアントの結果を基にしてより多くのサンプリング回数によって詳細な推定を実現する .

図 3 に、分散モンテカルロ追跡法の状態遷移を示す .分散モンテカルロ追跡法では、クライアントとサーバでそれぞれ独立に ConDensation アルゴリズムを実行する .このとき、クライアント上での推定結果をサーバに反映させるために、サーバ上では ICONDENSATION[3] で用

いられている重点サンプリング (Importance Sampling) を導入する .

ICONDENSATION は、ConDensation アルゴリズムとは異なる推定方法によって得られた確率密度に従ったサンプリングを追加することによって、異なる知識からの推定結果を統合する方法である .ここでは、クライアントで推定された確率密度に従ったサンプリングを追加することで、サンプリング回数やフレームレートが異なる 2 つの ConDensation アルゴリズムを統合する .

図 4 に分散モンテカルロ追跡法の概要を示す .サーバ上では、まずクライアント上で推定された最新の確率密度に従って m 個のサンプルを生成する .同時に、サーバ上で直前のフレームで推定された確率密度に従って $M - m$ 個のサンプルを生成する .こうして得られた計 M 個のサンプル集合に対して、各サンプルの評価値を計算する .

サーバ上での直前の追跡結果と、クライアントからの追跡結果のどちらをより信頼するかによって、それぞれのサンプル数を決定することができる .対象の動きが急な場合や通信障害などによってフレームレートが低い場合など、サーバ上での直前の結果をあまり信頼できないときには、 m を大きくすることで、クライアントからの最新の情報を重視して追跡することができる .また、サーバの直前の追跡結果が信頼できる場合には m を小さくすることで、サーバで得られたより詳細な結果を重視して追跡することができる .

分散モンテカルロ追跡法は単なる並列化とは異なり、クライアント単体でも追跡処理が完結するため、無線 LAN を経由した通信に遅延が生じる場合や、一時的に通信障害が生じる場合においても、クライアント上ではそのような遅延や通信障害の影響を受けることなくアクティブカメラへのフィードバックを行うことができる .

4 分散モンテカルロ追跡法による人物追跡

4.1 色情報と輪郭情報を用いた評価

実際に本研究で用いた人物追跡処理について述べる .本研究では、人物頭部形状を楕円によって近似し、人物頭部の状態を楕円の 5 つのパラメータによって次式のように表現する .

$$\mathbf{X} = (x, y, s, r, \theta) \quad (3)$$

ただし、 x, y は楕円の中心座標、 s はスケール、 r は偏平率、 θ は傾きとする .

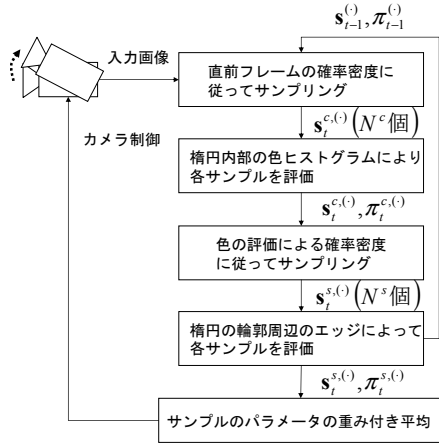


図 5: 人物追跡処理の概要

人物頭部の状態の評価は色情報と輪郭情報に基づいて行う。このとき、色情報の評価のためのサンプリングと、輪郭情報の評価のためのサンプリングを交互に行うことで、色情報による評価と輪郭情報による評価を統合する。

図 5 に実際の追跡手順を示す。まず、直前フレームの確率密度 $\{(s_{t-1}^{(c)}, \pi_{t-1}^{(c)})\}$ と推移確率 $p(\mathbf{X}_t | \mathbf{X}_{t-1})$ に従って、 N^c 個のサンプル $\{s_t^{c,(1)}, \dots, s_t^{c,(N^c)}\}$ を生成する。そして、各サンプルについて楕円内部の色ヒストグラムとあらかじめ学習した人物頭部の色ヒストグラムとのヒストグラムインタセクションを計算し、これを各サンプルの評価値 $\{\pi_t^{c,(1)}, \dots, \pi_t^{c,(N^c)}\}$ とする。

次に、確率密度 $\{(s_t^{c,(i)}, \pi_t^{c,(i)})\}$ に従って、新たに N^s 個のサンプル $\{s_t^{s,(1)}, \dots, s_t^{s,(N^s)}\}$ を生成する。そして、各サンプルについて楕円の輪郭上でエッジ検出を行い、評価値 $\{\pi_t^{s,(1)}, \dots, \pi_t^{s,(N^s)}\}$ を求める。ただし、実際には人物頭部形状が楕円によって近似されており、またランダムサンプリングによって状態を離散的に観測しているため、楕円の輪郭と人物頭部の輪郭は完全には一致しない。そこで、各観測点上で楕円の法線方向のエッジを検出し、観測点とエッジ点との距離に応じて減衰するような評価関数によって、輪郭周辺のエッジを含めて評価する [4]。

最終的に得られたサンプル集合と評価値の組 $\{(s_t^{s,(i)}, \pi_t^{s,(i)})\}$ を時刻 t における確率密度 $\{(s_t^{(c)}, \pi_t^{(c)})\}$ とする。ここで、各サンプルのパラメータを評価値で重み付けして平均することで、各フレームにおける人物の状態の推定値を得ることができる。クライアント上では、この推定値に従って、追跡対象人物を常に画像中に捉えるようにアクティブカメラを制御する。

このように、色情報の評価のためのサンプリングと、輪郭情報の評価のためのサンプリングを交互に行うことで、最初に色情報による評価である程度候補を絞り込むことができ、一度のサンプリングで色情報と輪郭情報の両方を評価するのに比べて効率良くサンプルを評価することができる。また、それぞれの評価の間に一度確率密度の形で標準化しているため、評価値の間の関係を考慮する必要がないという利点もある。

4.2 モーションセンサの情報の統合

3.1 節で述べたように、ConDensation アルゴリズムではフレーム間での追跡対象人物の状態変化を状態推移確率 $p(\mathbf{X}_t | \mathbf{X}_{t-1})$ としてあらかじめ与えておく必要がある。しかし、ウェアラブルシステムにおいては、ユーザの移動にともなってカメラの位置、姿勢が変化するため、追跡対象人物の画像上でのみかけの変化が大きくなり、あらかじめ与えた状態遷移確率だけでフレーム間の状態変化を表現することは困難である。また、アクティブカメラ制御によって、制御パラメータの変化に応じたカメラの姿勢変化も生じる。そこで、アクティブカメラの制御パラメータの変化とモーションセンサの情報を ConDensation アルゴリズムに統合することにより、カメラの姿勢変化を考慮した人物追跡を実現する。

時刻 t におけるカメラパラメータを θ_t^c, ϕ_t^c 、モーションセンサから得られるカメラの姿勢を $\theta_t^m, \phi_t^m, \psi_t^m$ とする。このとき時刻 $t-1$ から時刻 t までのカメラの角度変化 $\dot{\theta}_t, \dot{\phi}_t, \dot{\psi}_t$ は次式によって与えられる。

$$\begin{aligned} \dot{\theta}_t &= (\theta_t^c - \theta_{t-1}^c) + (\theta_t^m - \theta_{t-1}^m) \\ \dot{\phi}_t &= (\phi_t^c - \phi_{t-1}^c) + (\phi_t^m - \phi_{t-1}^m) \\ \dot{\psi}_t &= \psi_t^m - \psi_{t-1}^m \end{aligned} \quad (4)$$

これらの角度変化を用いて楕円の中心位置、及び傾きを補正する。他のパラメータ、つまりスケール及び偏平率に関しては、カメラの角度変化に対する変化が微量であるのでここでは無視する。時刻 t におけるあるサンプルの中心位置を (x_t, y_t) 、傾きを θ_t とすると、カメラの角度変化を考慮して補正されたサンプル位置 (x_t^h, y_t^h) と傾き θ_t^h は次式によって求められる。

$$\begin{aligned} x_t^h &= f \tan(\arctan(\frac{x_t}{f}) + \dot{\theta}_t) \\ y_t^h &= f \tan(\arctan(\frac{y_t}{f}) + \dot{\theta}_t) \\ \theta_t^h &= \theta_t + \dot{\psi}_t \end{aligned} \quad (5)$$

ただし、 f は焦点距離とする。

このようなパラメータの補正によって、追跡対象の状態変化から、カメラの姿勢変動に起因する変化を取り

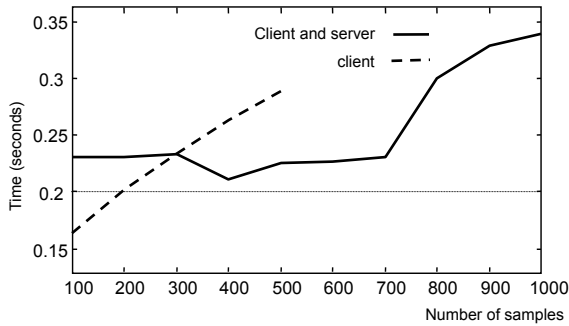


図 6: 1 フレームあたりの処理時間の比較

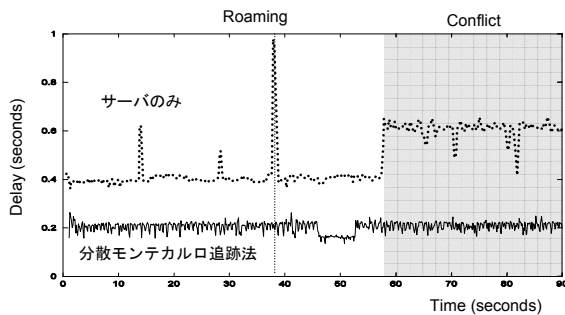


図 7: カメラ制御の遅延の比較

除くことができ、あとは対象人物の移動に起因する状態変化のみを考慮すればよい。そこで、対象人物の移動に起因する状態変化を状態推移確率 $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ として与え、補正されたパラメータと $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ を用いて次の時刻のサンプルを生成する。なお、実験では $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ をパラメータ毎に異なる標準偏差を持つ正規分布として与えた。

5 実験結果

提案手法を 2 節で述べた *VizWear-Active* の試作システム上に実装し実験を行った。なお、実験に用いた試作システムのスペックは、ウェアラブル PC が mobile-PentiumIII 500MHz を搭載したカード型 PC、サーバが dual Xeon 1.7GHz を搭載したデスクトップ PC である。

5.1 処理時間、及び遅延の評価

まず、クライアント、サーバそれぞれについて、サンプリング回数を変化させたときの 1 フレームあたりの処理時間を調べた結果を図 6 に示す。

破線のグラフは、クライアント上で色の評価のためのサンプリングの回数 N^c を 300 に固定し、形状の評価のためのサンプリングの回数 N^s を 100 から 500 まで変化させた場合のクライアント上での処理時間を表して

いる。また、実線のグラフは、クライアント上ではサンプリング回数を N^c, N^s をそれぞれ 300, 200 に固定し、サーバ上では $N^c = 500$ として、 N^s を 100 から 1000 まで変化させたときのクライアント、サーバの処理を含む処理時間を表す。

この結果より、クライアント上でサンプリング回数を変化させた場合は、サンプリングの回数にほぼ比例して処理時間が変化し、 $N^s = 200$ のときに約 0.2 秒となっていることがわかる。試作システムでは、1 フレームあたり 0.2 秒程度で処理できれば、アクティブカメラを制御しながら対象を見失うことなく追跡が可能であることが経験的にわかっているため、以降の実験ではクライアント上でのサンプリング回数を $N^c = 300, N^s = 200$ とした。

また、サーバ上では、サンプリング回数が 100 から 700 までの間は、0.2 秒を少し超える程度でほとんど変化せず、700 を超えるとサンプリング回数が増えるに従って処理時間も増えていることがわかる。これは、クライアント上での処理時間の 0.2 秒に無線 LAN による通信のオーバーヘッドが加わるためであり、これ以上サーバ上の処理を速くすることはできないが、クライアントに比べて大幅にサンプリング回数を増しても、ほぼ同じ時間で処理できていることがわかる。以降の実験ではサーバ上でのサンプリング回数を、全体の処理時間を変化させない最大のサンプリング回数である $N^c = 500, N^s = 700$ とした。

次に、画像をキャプチャしてからカメラ制御を行うまでの遅延時間を調べた結果を図 7 に示す。このとき、一般によく生じる通信障害を再現するために、実験開始から約 38 秒後に無線 LAN の基地局の切り換えを行い (Roaming)、約 48 秒以降には同じ無線チャンネルで別の通信を行った (Conflict)。

実線のグラフは分散モンテカルロ追跡法による遅延時間を示し、破線のグラフは入力画像をそのままサーバに転送し、サーバのみで追跡処理を行った結果をカメラ制御に用いた場合の遅延時間を示している。

この結果より、提案手法では遅延時間が約 0.2 秒であるのに対し、サーバの結果をカメラ制御に用いる場合では、無線 LAN の通信状態が安定しているときでも 0.4 秒程度の遅延が生じていることが確認できる。また、通信障害が生じたとき、サーバのみで追跡を行った結果を用いてカメラ制御を行う場合では、遅延がさらに悪化し、

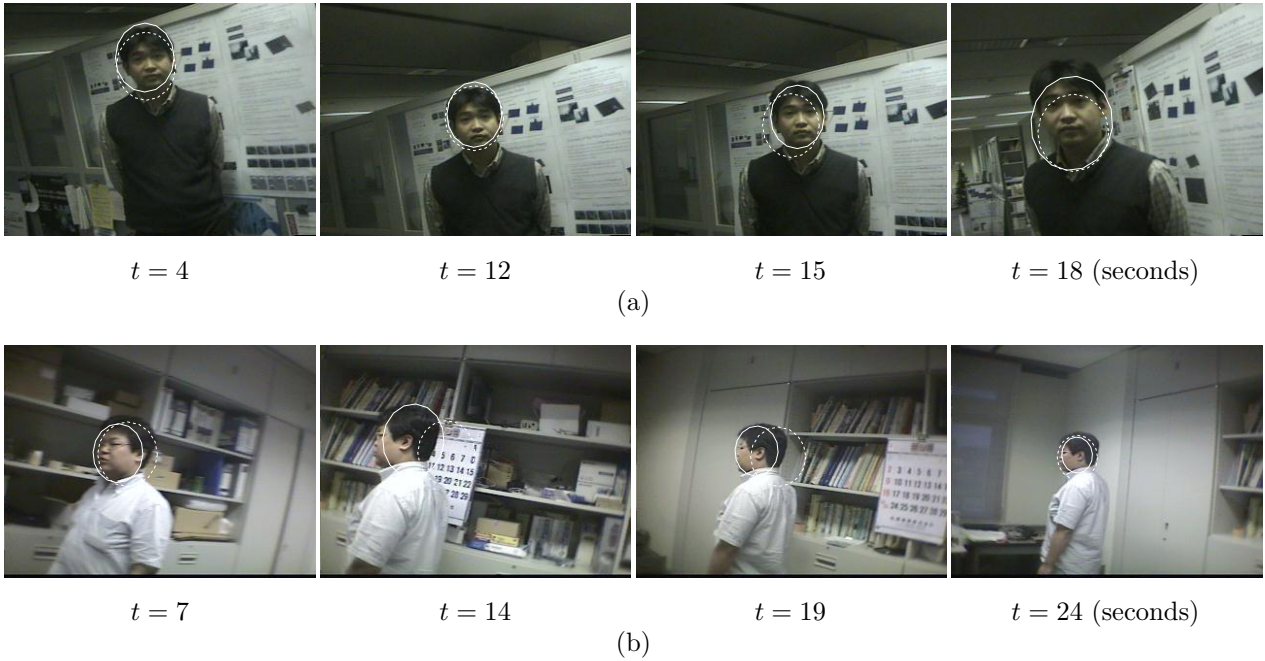


図 8: 人物追跡結果

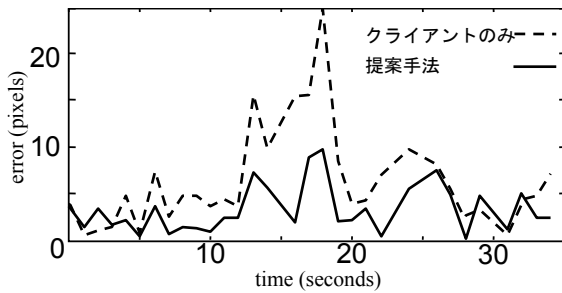


図 9: 推定誤差の比較



図 10: サンプルの分布

状況によっては 1 秒近い遅延が生じているのに対し、提案手法では、通信状態にもかかわらずほぼ一定の遅延でカメラ制御を行えていることがわかる。

これらの結果より、提案手法では通信状態にもかかわらず遅延の少ないカメラ制御が可能であり、また全体としてはクライアント単体より多くのサンプリング回数による追跡処理が可能であることが確認された。

5.2 人物位置、姿勢の推定精度の評価

人物追跡の各時刻において推定された楕円の中心位置と、手で与えた正解値との誤差を図 9 に示す。破線のグラフはクライアント単体による追跡結果の誤差を表し、実線のグラフは提案手法によってサーバ上で最終的に得られた結果の誤差を表している。この結果より、クライアント単体による追跡結果における誤差が、サーバ上での推定によって軽減され、高精度な追跡が実現できていることがわかる。

追跡結果の例を図 8 示す。図中の破線の楕円はクライアントのみによって推定された人物頭部を表し、実線の楕円はサーバ上で最終的に得られた追跡結果を表している。なお、(a) の結果は図 9 の誤差と対応している。この結果より、クライアントによる追跡結果には、位置だけでなく、スケールや傾きなどのずれが見られるが、提案手法によって、これらのずれが修正されていることがわかる。

また、ある時刻におけるサンプルの分布を図 10 に示す。図中の各点は各サンプルの楕円の中心位置を表している。また、左側の図はクライアント上でのサンプリングを行ったときの分布、右側の図は同じ画像に対するサーバ上でのサンプルの分布を表している。

クライアント上では、比較的広い範囲に粗くサンプルが分布し、サーバ上では狭い範囲でかつ密度が高くサンプルが分布していることがわかる。これは、クライアン

ト上ではフレーム間の推移確率 $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ をサンプルが広く分布するように設定して大まかな確率密度を求めたあとに、サーバ上ではその確率密度に従って今度は範囲を絞ってサンプリングを行っているためである。これによって、人物が速く移動する場合でもクライアントで対象人物を見逃すことなく追跡を行うことができ、また同時にサーバ上では詳細な姿勢を求めることが可能となる。

6 まとめ

本稿では、ウェアラブルシステムの限られたリソースで、安定かつ実時間の追跡を実現するための手法として分散モンテカルロ追跡法を提案した。本手法は Condensation アルゴリズムをクライアント/サーバによる分散処理に拡張した方法であり、クライアント上での大まかな追跡によるアクティブカメラへの遅延の少ないフィードバックと、サーバ上での高精度な人物位置、姿勢に推定を同時に実現することができる。また、ユーザの動作やカメラ制御によって入力画像中の人物位置が大きく変化する問題を解決するために、モーションセンサで計測される動作情報を追跡アルゴリズムに統合する方法を提案した。実験では、提案手法によってクライアント上では1フレーム約0.2秒で追跡することができ、また同時にサーバ上ではより正確な追跡結果が得られることが確認された。

本稿ではアクティブカメラの制御を伴う追跡処理において、アクティブカメラへの高速なフィードバックを得るためにクライアント/サーバでの分散処理を導入したが、手のジェスチャ認識に基づいたインターフェイス [10] など、システムとユーザとのインタラクションが生じる場合にも高速なフィードバックが要求される状況は多く存在し、そのようなアプリケーションへも同様のアプローチを適応することで、効率的なシステムを構築することが可能であると考えられる。

今後の課題として、サーバ上の詳細な追跡処理の結果をクライアントへフィードバックすることによって、クライアント上による追跡をより安定にすることが挙げられる。また、*VizWear-Active* の目的である状況理解に基づくユーザ支援のためには、本手法によって収集した人物情報を、顔認識やデータマイニングなどの技術を用いて整理し、適切な情報をユーザに提示することが必要である。

謝辞

本研究は、文科省科学技術振興調整費の支援による。また、本研究で使用したウェアラブルアクティブカメラは株式会社ベストテクノロジーの協力によって製作した。記して謝意を表す。

参考文献

- [1] <http://www.is.aist.go.jp/vizwear/>.
- [2] Jonny Farrington and Vanessa Oni. Visual augmented memory. In *4th International Symposium on Wearable Computers (ISWC2000)*, pp. 167–168, 2000.
- [3] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. 5th European Conference on Computer Vision (ECCV98)*, pp. 893–908, 1998.
- [4] Micheal Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, Vol. 29, No. 1, pp. 5–28, 1998.
- [5] Takekazu Kato, Takeshi Kurata, and Katsuhiko Sakaue. Face registration using wearable active vision systems for augmented memory. In *DICTA2002*, pp. 252–257, Jan 2002.
- [6] Takekazu Kato, Takeshi Kurata, and Katsuhiko Sakaue. VizWear-Active: Distributed monte carlo face tracking for wearable active cameras. In *ICPR2002*, aug 2002. (to appear).
- [7] Takeshi Kurata, Takashi Okuma, Masakatsu Kouroggi, Takekazu Kato, and Katsuhiko Sakaue. VizWear: Toward human-centered interaction through wearable vision and visualization. In *PCM2001*, pp. 40–47, 2001.
- [8] W.W. Mayol, B. Tordoff, and D.W. Murray. Wearable visual robots. In *4th International Symposium on Wearable Computers (ISWC2000)*, pp. 95–102, 2000.
- [9] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, W. R. Picard, and A. Pentland. Augmented reality through wearable computing. Technical Report 397, M.I.T Media Lab. Perceptual Computing Section, 1997.
- [10] 蔵田武志, 興梠正克, 加藤丈和, 大隈隆史, 坂上勝彦. ハンドマウスとその応用: 色情報と輪郭情報に基づく手の検出と追跡. 画像情報システム研究会, 情報メディア学会, pp. 47–52, Dec 2001.
- [11] 蔵田武志, 大隈隆史, 興梠正克, 加藤丈和, 坂上勝彦. VizWear: コンピュータビジョンとウェアラブルディスプレイによる人間中心インタラクション. 高臨場感ディスプレイフォーラム, 2001.