

柔軟なインタフェースの実現に向けた人間行動の計測と理解

岡 兼司[†] 佐藤 洋一[‡]

[†] 松下電器産業株式会社 [‡] 東京大学生産技術研究所

概要： 環境の側が人間の行動を計測・理解した上で自動的に最適なサービスを提供するようなインタフェースは「柔軟なインタフェース」と呼ばれ、マン・マシン・インタフェースの中でも今後の重要な方向性の一つとなっている。我々は、柔軟なインタフェースを実現する上で最も重要な技術の一つとして、人間の動作を実時間で安定に計測・理解する技術について研究を行ってきた。特に、人間の意図が頻繁に表出する器官として手指と頭部に注目し、それらの動作を計測するための技術を提案した。本稿では、各技術の詳細について説明するとともに、幾つかの評価実験を通して提案手法の安定性・実時間性の高さを証明する。さらに、各計測技術を実際のインタフェースに適用した例についても紹介する。

Vision-based Tracking and Understanding of Human Body Movements for Human Computer Interface

Kenji OKA[†] Yoichi SATO[‡]

[†] Matsushita Electric Industrial Co. Ltd. [‡] The University of Tokyo

Abstract : “Ubiquitous computing environment” requires an innovative type of Human-Computer Interface (HCI) where ubiquitous computers automatically measure and understand users’ motion and they provide attentive services to the users depending on the measuring results. One of the key components for realizing such interfaces is to measure and understand human motion reliably in real time. In our work, we have proposed reliable methods for tracking each part of human body movements based on computer vision without using any invasive devices and recognizing the tracking results as various types of gestures. We have demonstrated the performance of our tracking methods via some experiments.

1 はじめに

昨今のコンピュータの急激な普及の中で、Graphical User Interface (GUI) は非常に大きな役割を果たしてきた。GUI が提供するインタフェースは、ユーザが常にコンピュータに注意を向けているという前提のもとで、非常に使い易く直観的なデザインがなされている。

その一方で、近年普及しつつあるユビキタスコンピューティング環境の下において、GUI は必ずしも適切なインタフェースとは言えないのが実情である。ユビキタスコンピューティング環境下では、GUI の前提とは逆に、ユーザがコンピュータの存在を意識せずに済むことが求められる。今後、ユビキタスコンピューティング環境の益々の普及が予想されていることから、そのような環境下で効率よく機能するインタフェースの開発が急務とされている。

この問題を解決しうるインタフェースとして注目されているのが、柔軟なインタフェースである。柔

軟なインタフェースとは、コンピュータの側が実世界における人間の行動を自動的に計測・理解し、その結果にもとづいてコンピュータがユーザに最適なサービスを提供するという形態のインタフェースである。ここでは、ユーザはコンピュータの存在を意識しなくて済むため、このインタフェースはユビキタスコンピューティング環境への応用に非常に適したインタフェースとして考えられている。同様の概念に沿ったインタフェースとして、Perceptual User Interface (PUI) [26] や Attentive User Interface (AUI) [29] などとも挙げられる。

柔軟なインタフェースを実現するために最も重要となる要素技術の一つとして、人間の行動を計測・理解するための技術が挙げられる。そこで本研究では、人間の行う多種多様な動作を計測・理解するための技術について研究を行うこととする。特に、人間が頻繁に動作させ、かつ人間の意図が頻繁に表出する器官として、手指と頭部の動作計測を行うもの

とする。

本研究では、ユーザに何もデバイスを装着しない非接触型の手法として、コンピュータビジョンにもとづく動作計測技術について研究を進める。そして、単純に計測可能というだけではなく、安定性、計測精度、実時間性などの面で十分にインタフェースに応用可能な計測技術の開発を目指す。加えて、以上の計測結果をインタフェースに応用した例についても幾つか示す。

以下、本稿は次のような構成となっている。第2節では、手や指先位置を計測し、その動作を認識するための研究成果について報告する。次いで第3節で、頭部の三次元的な姿勢を実時間で推定するための研究について説明する。さらに、第4節で人物動作の計測結果をインタフェースに応用した研究例を報告する。最後に、第5節で本研究の結論について述べる。

2 手と指先位置の計測と動作認識

人間が実世界の中での作業を行うにあたって、最もよく動作させる器官といえば手や指先である。人間の手や指先の動作を計測することにより得られる情報は、柔軟なインタフェースを構築するにあたって非常に大きな意味を持つと考えられる。

従来からの研究においても、コンピュータビジョンにより手や指先の動作を計測するための手法は提案されてきた。しかしながら、インタフェースへの応用に必要な条件である実時間性や計測精度、安定性を同時に満たすような手法はほとんど存在しなかった。

そこで本研究では、実時間性と精度の面でインタフェースへの応用に適した手指動作計測技術について研究を進める。特に、複数の指先動作を実時間で安定して計測・理解するための手法の開発を目指す。

2.1 先行研究の問題点と本研究のアプローチ

本節では、本研究の先行研究である指先追跡手法について説明する。ここで述べる先行研究の手法、および新たに提案する手法は、机型インタフェース EnhancedDesk (図1)での作業をターゲットとして開発したものである。

従来の手指追跡手法では、手領域を抽出するために肌色抽出や背景差分などの手法が多く用いられてきたが [19, 16]、拡張机型インタフェース上では、プロジェクタによる投影の影響によりユーザの手の色や背景が動的に変化するため、これらの従来手法では安定に手指を追跡することが困難であった。このような問題に対して、我々の先行研究において、中赤外領域光を撮影する赤外線カメラを利用することにより、動的に光源環境が変化する場合や背景が複雑な場合にも、マーカなどのデバイスを用いることなく、複数の指先位置を計測するための手法を開発してきた [21]。また、この計測結果を利用することにより、机上に投影されたアプリケーションをユーザ自身の手によって直接的に操作すること、すなわち、ダイレクトマニピュレーションにもとづくイン

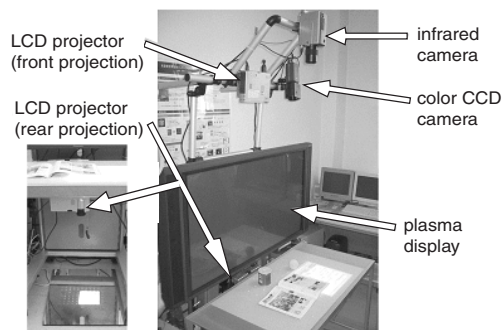


図1: 机型インタフェース EnhancedDesk の概観

タラクションを可能としてきた。

しかしながら、この先行研究には幾つかの問題点が残されていた。最初の問題点として、先行研究の指先位置計測手法では個々の入力画像フレームで独立に指先位置を特定していただけであり、個々の指先についての追跡を行っていなかったことが挙げられる。このために、ユーザの時系列の動作に関する情報が欠如していることが問題となってきた。また、過去の計測結果を考慮せずに個々の画像フレームで独立に指先位置計測を行うことは、指先位置計測の安定性の欠如にも大きく影響してきた。

さらに先行研究では、認識可能なインタラクションの種類が大幅に制限されてしまうという問題も存在した。一般に、手指の追跡結果をインタラクションに利用する際には、大きく分けて二種類のインタラクションモードが存在する。その一つがシンボリックジェスチャである [31, 25, 13]。ここでは、ある一定時間における一連の動作により定義される操作をシンボリックジェスチャとして考える。一方で、もう一つのインタラクションモードがダイレクトマニピュレーションであり、個々の入力画像フレームで計測された情報を直接的にインタラクションに利用するモードとして定義される [5, 22, 27]。先行研究では、個々の入力画像フレームから得られる情報をもとにダイレクトマニピュレーションによるインタラクションを可能としていた一方で、一連の手指動作に関する情報を獲得できなかったためにシンボリックジェスチャを認識することが困難であった。手の動作にもとづく多様なインタラクションを実現するためには、シンボリックジェスチャも含めた二種類のインタラクションモードを連携して利用することが不可欠である。

そこで本稿では、以下の2点について提案する。

複数の指先動作の計測 まず、個々の入力画像フレーム中で検出された指先位置に対して予測フィルタを適用することにより、次の画像フレームにおける指先位置を予測する。次いで、これらの予測位置と次フレームで実際に検出された指先位置との最適な組み合わせを決定す

ることにより、隣り合う画像フレーム間で指先同士の対応付けを行う。

二種類のモードの連携 まず、ユーザの親指の動作に着目してダイレクトマニピュレーションとシンボリックジェスチャの判別を行う。その後、ダイレクトマニピュレーションに対しては指先位置を直接的に利用した処理を適用するのに対し、シンボリックジェスチャに対してはHidden Markov Model (HMM) [20] にもとづく認識処理を行い、その結果をインタラクションに利用する。

以降、2.2 節で指先動作の計測手法について、2.3 節でインタラクションモードの連携手法について説明する。

2.2 複数の指先動作の実時間計測

2.2.1 個々の画像フレームにおける指先、手のひら中心および指の付け根の検出

本節では、我々の先行研究において開発された個々の入力画像フレームから実時間で複数の指先を検出するための手法について簡単に説明する [21]。

従来手法で利用されてきた肌色抽出や背景差分にもとづく手領域の抽出手法には光源環境や背景の状況に影響されやすいといった問題がある。そこで本手法では、このような問題点を回避するために、中赤外領域光を撮影する赤外線カメラによってユーザの手の熱放射を計測する。ここで得られる赤外線カメラ画像では、手に対応する領域で輝度が非常に高くなる。そのため、この画像に対して適当な閾値の二値化処理を適用することにより、背景が複雑な場合や光源環境の変化が起こる環境下においても、ユーザの手に対応する領域を画像中で安定に抽出することが可能となる。

手領域を抽出した後、手領域に応じて設定された探索領域内において指先を検出する。本手法では適当な大きさの円形テンプレートを用いて正規化相関にもとづくパターンマッチングを行い、複数の指先候補点を検出する。さらに、多重マッチングや指関節周辺での誤検出を除去することにより、複数の指先位置を特定する。

その後、手のひらの中心位置を計測するための処理を行う。本手法では、抽出された手領域に対して、領域が十分に小さくなるまで収縮処理を繰り返し適用し、残った領域の重心を手のひら中心として決定する。

以上の処理によって複数の指先および手のひらの中心を検出するが、2.3.1 節における親指判別を行うためには指の付け根の位置を特定する必要がある。ここではまず、領域収縮処理を利用して手領域を手のひら領域と指領域に分割する。その後、得られた指領域について、先に検出された指先位置を考慮することにより、指先に対応する端点とは反対側の端点を指の付け根の位置として決定する。

なお、以上で述べたような二値化閾値や円形テンプレートの大きさなどはユーザによってある程度異

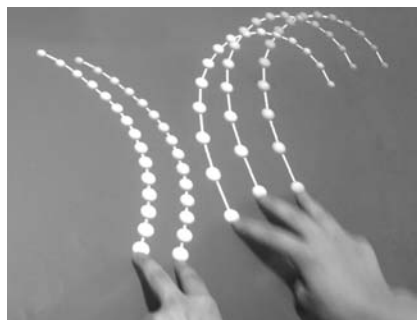


図 2: 指先軌跡の計測

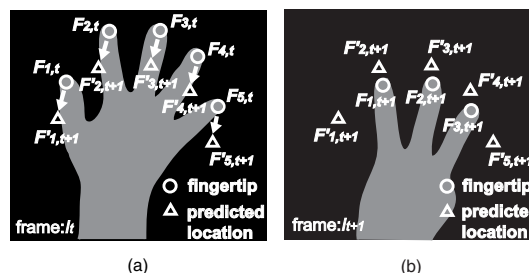


図 3: 指先対応付け手法

なる。そのため、これらのパラメータは初期化時に取り込んだ赤外線カメラ画像にもとづいてユーザ毎に設定する。これにより、ユーザに依存しない安定したシステムが実現できる。

2.2.2 画像フレーム間での指先同士の対応付け

本計測手法では、2.2.1 節の手法を用いて検出された複数の指先について、それらを隣り合う画像フレーム間で対応付けることにより、背景や光源環境の動的な変化が起こる場合にも、複数の指先の動作を実時間で安定に計測することを可能とする (図 2)。

ここでは、 t 番目の画像フレーム I_t で n_t 個の指先が検出された場合を想定し、これらの指先位置を $F_{i,t}$ ($i = 1, 2, \dots, n_t$) とする (図 3 (a))。まず、次の画像フレーム I_{t+1} での指先位置 $F'_{i,t+1}$ を予測フィルタを用いて予測する。次いで、画像フレーム I_{t+1} で実際に検出された n_{t+1} 個の指先位置 $F_{j,t+1}$ ($j = 1, 2, \dots, n_{t+1}$) と予測位置 $F'_{i,t+1}$ との比較を行う (図 3 (b))。これらの検出位置と予測位置との間で最適な組み合わせを求めることにより、隣り合う画像フレーム間での指先同士の対応付けが実現する。

以下、指先位置予測手法と対応付け手法について詳細を説明する。

指先位置の予測

本節では、ある画像フレームで検出された指先位

置をもに、予測フィルタを用いて次フレームにおける指先位置を予測する手法について説明する。本手法では、予測フィルタとして特徴点追跡などにしばしば利用されるカルマンフィルタを適用する。なお、本節で説明する処理を各指先に対して独立に適用する。

まず、本手法では各画像フレームに対して各指先の位置と速度についての推定を行うため、状態ベクトル x_t を次のように定義する。

$$x_t = (x_t, y_t, \dot{x}_t, \dot{y}_t)^T \quad (1)$$

ここで、 $(x_t, y_t, \dot{x}_t, \dot{y}_t)$ は、 t 番目の画像フレームにおける指先位置 (x_t, y_t) 、および指先の速度 (\dot{x}_t, \dot{y}_t) を表す。

次に、観測ベクトル y_t を t 番目のフレームで検出された指先位置として定義する。このとき、状態ベクトル x_t と観測ベクトル y_t を次式により関連付ける。

$$x_{t+1} = Fx_t + Gw_t \quad (2)$$

$$y_t = Hx_t + v_t \quad (3)$$

この式における F は状態遷移行列、 G は駆動行列、 H は観測行列を表し、また、 w_t は状態ベクトル x_t の速度成分に加わるシステム雑音、 v_t は実際の位置と検出された位置との誤差に相当する観測雑音を示す。

さらに、本手法では画像フレーム間隔 ΔT が非常に短いため、連続する画像フレーム間での各指先の移動を等速直線運動として近似する。そこで、 F, G, H を次のように定義する。

$$F = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (5)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (6)$$

ここで観測行列 H は作業空間である机上で定義される世界座標系と画像座標系との変換を表現する行列であり、カメラモデルとして透視投影を考えた場合、一般には射影変換で記述される。一方で、本稿では議論を簡潔にするために、一般性を失うことなく、状態ベクトル x_t の (x, y) 座標と観測ベクトル y_t の (x, y) 座標が一致するとしている。

また、システム雑音 w_t と観測雑音 v_t については、二次元零ベクトルを平均ベクトルとして持つ一定のガウス雑音として仮定する。それゆえ、 w_t および v_t の共分散行列はそれぞれ $\sigma_w^2 I_{2 \times 2}$ 、 $\sigma_v^2 I_{2 \times 2}$ となる。ここで、 $I_{2 \times 2}$ は 2×2 単位行列である。

以上の定義を用いて、カルマンフィルタを次式の

ように定式化する。

$$K_t = \tilde{P}_t H^T (I_{2 \times 2} + H \tilde{P}_t H^T)^{-1} \quad (7)$$

$$\tilde{x}_{t+1} = F \{ \tilde{x}_t + K_t (y_t - H \tilde{x}_t) \} \quad (8)$$

$$\tilde{P}_{t+1} = F (\tilde{P}_t - K_t H \tilde{P}_t) F^T + \frac{\sigma_w^2}{\sigma_v^2} \Lambda \quad (9)$$

ここで、 $\tilde{x}_t = \hat{x}_{t|t-1}$ であり、この $\hat{x}_{t|t-1}$ は y_0, \dots, y_{t-1} から推定される x_t の推定値を表す。また、 $\tilde{P}_t = \hat{\Sigma}_{t|t-1} / \sigma_v^2$ であり、 $\hat{\Sigma}_{t|t-1}$ は $\hat{x}_{t|t-1}$ の推定誤差の共分散行列を表す。さらに、 K_t はカルマンゲインであり、 $\Lambda = GG^T$ である。

以上の式を計算することにより、次の画像フレーム、すなわち $t+1$ 番目の画像フレームにおける予測位置を \tilde{x}_{t+1} の中の成分 (x_{t+1}, y_{t+1}) として得る。

なお、2 フレーム以上後における指先位置、すなわち m ($m > 1$) フレーム後での予測が必要な場合には次式により計算を行う。

$$\hat{x}_{t+m|t} = F^m \{ \tilde{x}_t + K_t (y_t - H \tilde{x}_t) \} \quad (10)$$

$$\hat{P}_{t+m|t} = F^m (\tilde{P}_t - K_t H \tilde{P}_t) (F^T)^m + \frac{\sigma_w^2}{\sigma_v^2} \sum_{l=0}^{m-1} F^l \Lambda (F^T)^l \quad (11)$$

ここで、 $\hat{x}_{t+m|t}$ は y_0, \dots, y_t から推定される x_{t+m} の推定値である。また、 $\hat{P}_{t+m|t} = \hat{\Sigma}_{t+m|t} / \sigma_v^2$ であり、 $\hat{\Sigma}_{t+m|t}$ は $\hat{x}_{t+m|t}$ の推定誤差の共分散行列を表す。

予測位置と実際の検出位置との対応付け

本節では、実際に検出された指先位置と式 (8) や (10) によって予測された指先位置とを対応付けるための手法について説明する。

ここでは、検出された指先位置と予測位置との間の複数の組み合わせについて距離の二乗和を計算し、その和が最小となる組み合わせを最適な組み合わせとして決定する。

ここで計算コストの削減を図るために、手のひらの中心のまわりにおける指先の順番を考慮することにより、計算すべき組み合わせの数を減らす (図 4 (a))。すなわち、指の交差などによる入力画像中の指先の順番変化は起こらないと仮定し、図 4 (a) の場合では、3 通りの組み合わせ (i) $\bigcirc 1-\Delta 1$ & $\bigcirc 2-\Delta 2$, ii) $\bigcirc 1-\Delta 1$ & $\bigcirc 2-\Delta 3$, iii) $\bigcirc 1-\Delta 2$ & $\bigcirc 2-\Delta 3$) についてのみ計算を行うこととする。これにより、考慮すべき組み合わせの数は最大で ${}_5P_5$ 通りから ${}_5C_5$ 通りに削減される。

次に、入力画像中で幾つかの指先を検出できなかった場合の対応について説明する。このような状況の一例を図 4 (b) に示す。この図では、手の高速移動時における手領域のぶれなどの影響により、親指と小指についての検出に失敗している。本手法では、検出に失敗した指先の予測位置に関する情報を次のようにして利用することにより、指先追跡を続行するかどうかについて判断する。

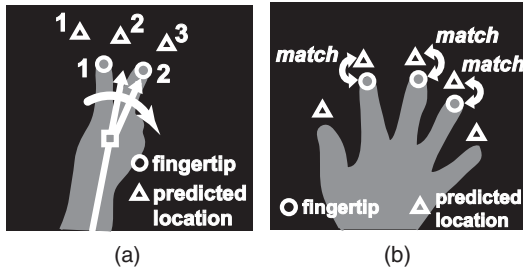


図 4: 指先の検出位置と予測位置との対応付け

ある予測位置に対応する指先位置が見つからなかった場合、その予測位置に対応する式(9)中の共分散行列 P_{t+1} の $(1, 1)$ 成分を調べる。この成分は指先位置予測についての不確定性を表していることから、この成分がある閾値よりも小さい場合にはこの予測位置を信頼し、この予測位置に対応する指先は何らかの原因により偶然検出されなかったにすぎないとみなす。そこでこの場合には、予測位置を実際の指先位置とみなして指先追跡を続行することとする。一方で、この $(1, 1)$ 成分が閾値よりも大きい場合には、予測された位置は信用できないと判断し、その指先に関する追跡を終了する。現在の実装では、不確定性に関する閾値として経験的に定められた一定の値を使用している。

また、予測位置の数よりも検出された指先位置の数の方が多き場合、どの予測位置にも対応付けられなかった指先位置について追跡を開始する。このとき、その指先位置から計算される予測位置の不確定性が閾値よりも小さくなるまでは新たな指先の軌跡としてみなさないこととする。

2.2.3 評価実験：指先動作の計測

本研究の複数指先追跡手法の信頼性について評価実験を行った。特に、隣り合うフレーム間での指先同士の対応を考慮することにより、指先追跡の信頼性がどの程度向上するかについて比較実験を行った。

本実験には7名の被験者が参加した。実験環境としては、Intel Pentium III 500MHz を搭載した Linux PC を使用し、画像処理ボードには日立 IP5005 を、赤外線カメラにはニコン LAIRD-S270 を使用した。

各試行において、被験者は図4(a)のように伸ばした指の本数を一定に保ったまま机上で自由に手を動かした。最初に、被験者は1本だけ指を伸ばして30秒間手を動かした。次いで、伸ばした指の本数を2本、3本、4本、5本として同様に動かした。各試行はそれぞれ30秒間行ったため、各試行につき約900フレームのデータが得られた。

本実験では、隣り合う画像フレーム間で検出された指先同士の対応関係を考慮する場合としない場合の指先追跡性能について比較実験を行った。このとき、公平な比較を行うために、カメラ出力を一旦ビデオに録画し、そのビデオに対して各追跡手法を適

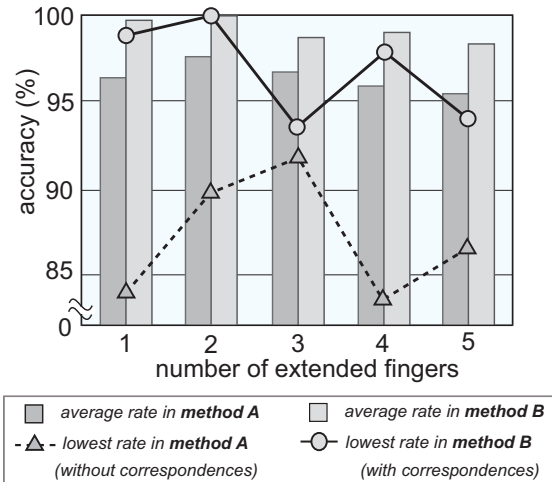


図 5: 指先追跡手法の評価実験

用した。

実験結果を図5に示す。手法Aは指先同士の対応を考慮しない手法を表し、手法Bは対応関係を考慮する手法を表している。また、図中の棒グラフは追跡された指先の数についての正解率における被験者7名の平均値を示し、折れ線グラフは正解率についての被験者7名中の最低値を示す。

図5を見ると、フレーム間で指先同士の対応を考慮することによって、指先追跡性能が大きく向上していることがわかる。さらに、折れ線グラフにあるように正解率の最低値についても大幅な向上が見られる。このことから、本追跡手法はユーザにかかわらず非常に信頼性が高く、実用的なアプリケーションへの応用に適していることがわかる。

2.3 指先動作にもとづくインタラクション

本節では、計測された指先動作にもとづいて、ダイレクトマニピュレーションとシンボリックジェスチャを連携して利用するための機構について述べる。

本研究で提案する機構は拡張机型インタフェースにおける様々なアプリケーションでの利用が可能であることを前提としているが、ここでは具体例として、第4節で示すような描画システムで利用されるダイレクトマニピュレーションとシンボリックジェスチャ、さらにはそれらを組み合わせるための機構について述べる。

まず、ダイレクトマニピュレーションとしては、物体をつまんで移動させるような動作を行うことによって、仮想物体をマニピュレーションするインタラクションが中心となる。このとき、個々の入力画像フレームにおいて計測された指先位置を利用して、マニピュレーションのモードや移動先の位置などの決定を行う。一方、シンボリックジェスチャとしては、人差し指1本で図6に示すような軌跡を描くという一連の動作を行うと、その軌跡の意味を認識器によ

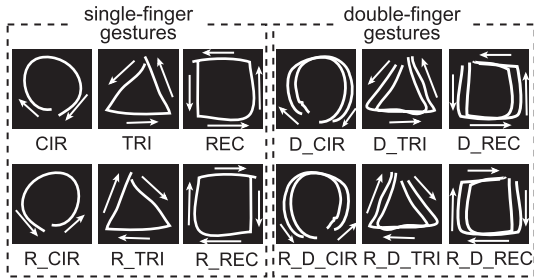


図 6: シンボリックジェスチャの例

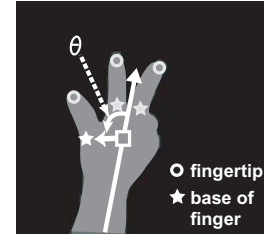


図 8: 角度 θ の定義

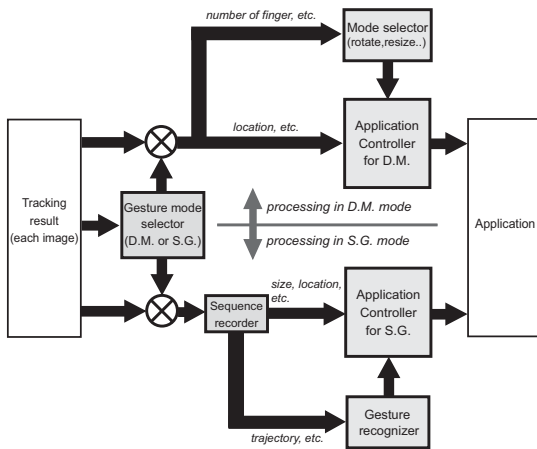


図 7: 手指によるインタラクションのための処理の流れ

り認識した上で、描いた位置に認識結果を表示するようなインタラクションを行う。また、2本指で同様に軌跡を描くと選択範囲を指定できるような操作なども考えられる。

このような手指による様々なインタラクションを実現する機構として、本研究では図7に示す機構を利用する。まず、指先動作の情報から手の動作がダイレクトマニピュレーションであるか、シンボリックジェスチャであるかを判別する。次いで、ダイレクトマニピュレーションとして判別された動作については、指先同士の間隔や伸ばした指の本数にもとづいて「回転」や「移動」などの操作モードを選択してから、指先の位置情報をもとに操作モードの変量を制御する。一方、シンボリックジェスチャとして判別された動作については、HMMにもとづいて構築された認識器に追跡結果のシーケンスを入力してジェスチャの種類を認識し、さらに軌跡情報からジェスチャの位置や大きさなどの情報を付加するといった処理を行う。

このとき、ダイレクトマニピュレーションとシンボリックジェスチャの判別については、机上での細か

なダイレクトマニピュレーションでは人差し指に加えて親指が非常に多く用いられるという観察にもとづいて、親指の動作に注目してこれら二種類のモードを切り替えることとする。すなわち、ユーザが親指を折り曲げることにより親指が追跡されていないときのジェスチャをシンボリックジェスチャとして、親指を伸ばして親指が追跡されているときのジェスチャをダイレクトマニピュレーションとしてインタラクションモードの切り替えを行う。なお、本研究における指の判別（ある追跡中の指が親指であるか人差し指であるか、など）においては、五指をすべて判別するのではなく、親指かそれ以外の指かという判別を行うこととする。これは、親指かそれ以外の指かの判別だけでもインタラクションモードの切り替えに関して大きな支障はないという判断に加えて、親指はその他の指と比較して手の構造上判別が容易であるという理由による。

以下、この機構を実現するために必要な要素技術として、2.3.1節では追跡中の複数の指先の中から親指を判別するための手法について、2.3.2節ではシンボリックジェスチャの種類認識手法について説明する。

2.3.1 親指の判別

本節では親指とそれ以外の指先を区別するための手法について説明する。本手法では、図8に示すように、指の向き（手のひらの中心から指の付け根へと向かう向き）と腕の向き（腕の慣性主軸の向き）との間の角度 θ を求め、この角度 θ をもに親指であるかどうかを判別する。ここで指の付け根を利用する理由は、指を動かしたときでも、指先と比較して指の付け根はほとんど動かないためである。

まず初期化の段階で、親指の標準角度 θ_T と人差し指の標準角度 θ_F ($\theta_T > \theta_F$) を求めておく。ここで標準角度とは、親指らしさや人差し指らしさを表すための基準となる角度である。これらは、初期化時に取り込んだ手の画像に対して2.2.1節で述べた手法を適用することにより親指と人差し指の付け根位置を検出した後で、これらの付け根に対する角度 θ を計算することによって求められる。

その後、個々の入力画像フレームにおいて、2.2.1節で述べた手法により検出された指の付け根に対する角度 θ を計算する。ここで、ある指の軌跡の開始点

表 1: 親指判別の評価実験

タスク	(a)	(b)	(c)
平均値 (%)	98.3	99.4	98.2
標準偏差 (%)	4.6	0.8	3.6

から k 番目のフレームにおける角度 θ を θ_k とする。また、現在のフレームがその指の軌跡の開始点から N 番目のフレームにあたるとする。このとき、親指らしさを表すスコア s_T は次式により計算される。

$$s'_T(k) = \begin{cases} 1.0 & \text{if } \theta_k > \theta_T \\ \frac{\theta_k - \theta_F}{\theta_T - \theta_F} & \text{if } \theta_F \leq \theta_k \leq \theta_T \\ 0.0 & \text{if } \theta_k < \theta_F \end{cases} \quad (12)$$

$$s_T = \frac{\sum_{k=1}^N s'_T(k)}{N} \quad (13)$$

こうして得られた s_T が 0.5 以上のとき、その指を親指としてみなす。

本研究の親指判別手法では、手領域を含めた腕の慣性主軸方向を考慮した角度を利用していることや親指の付け根と人差し指の付け根の位置が大きく離れていることに加えて、過去の履歴を考慮したスコアの計算を行っているため、不自然な手の姿勢を長時間持続させない限り、安定に親指の判別を行うことが可能である。

そこで、本節で提案する親指の判別性能についての評価実験を行った。実験環境は 2.2.3 節の実験と同様である。本実験では、机上の作業を想定した 3 種類のタスクについて実験を行った。まず、タスク (a) が人差し指だけを利用した描画作業、そして、タスク (b) が親指と人差し指によって物体をつまむ作業である。ここで、タスク (a) はシンボリックジェスチャによるインタラク션을想定した作業であり、一方、タスク (b) ではダイレクトマニピュレーションによる操作を想定している。また、タスク (a) に対する実験の補足として、親指だけを利用した描画作業であるタスク (c) に対する実験も行った。

各タスクにおける親指判別の正解率を表 1 に示す。この表によると、本実験ではいずれのタスクにおいても非常に高い正解率で親指の判別が行われた。このことから、本手法では信頼性の高い親指判別が実現されていることがわかる。

2.3.2 シンボリックジェスチャの認識

他のシンボリックジェスチャ認識 [31, 25, 13] と同様に、本手法においても HMM を利用してシンボリックジェスチャの種類を認識を行う。認識器への入力については、検出された指先の本数と指先の動きの向きという二種類のデータ系列を使用する。これにより、指先の動作と伸ばした指の本数の両方に依存したシンボリックジェスチャ認識が可能となる。

ここで、指先の動きの向きについては次のように求める。まず、追跡中の各指先についての動きのベク

表 2: シンボリックジェスチャ認識の評価実験

ジェスチャタイプ	1 本指	2 本指
平均値 (%)	99.2	97.5
標準偏差 (%)	0.5	1.8

トルを求め、それらの平均ベクトルを計算する。次いで、角度の一周を 16 分割したときに、この平均ベクトルの向きに対応する角度がどこに含まれるかに応じて、コード 1 からコード 16 までのコードを割り当てる。さらに、16 種類のコード以外に指先がほとんど静止した状態を表すコードを用意し、計 17 種類のコードを指先の向きを表す入力として使用する。このときに指先の動きの平均ベクトルを用いた理由は、親指以外の指でシンボリックジェスチャを行う場合、伸ばした指の指先が一つのシンボリックジェスチャの中で同様の動きを取ることが多いためである。

そこで、本節で紹介したシンボリックジェスチャ認識の性能について評価実験を行った。認識するシンボリックジェスチャとして、図 6 に示すような指先軌跡による 12 種類のシンボリックジェスチャを用意した。

1 名によって行われたシンボリックジェスチャを各シンボリックジェスチャに対して 80 試行分用意し、それらを用いて HMM の学習を行った。認識実験には学習データを採取した 1 名を含まない 6 名が参加し、各シンボリックジェスチャに対して 20 回の試行を行った。このとき、シンボリックジェスチャの順番や大きさ、描く位置については各被験者の任意で決定した。得られた実験結果は表 2 の通りである。表には 1 本指のシンボリックジェスチャと 2 本指のシンボリックジェスチャについての認識の正解率を示している。この結果から、1 本指のシンボリックジェスチャの認識率は 100% に近く、非常に頑健であることがわかる。また、2 本指のシンボリックジェスチャ認識についても高い認識率を示している。

3 頭部の三次元的な姿勢の計測

本節では、頭部姿勢を計測するための手法について説明する。まずは、具体的な手法の説明に先立って、本研究で頭部姿勢に注目する理由を述べる。

第 2 節で手と指先の計測技術およびジェスチャ認識技術に関する研究について述べた。この研究により、ユーザの手指の動作を高い信頼性のもとで計測することが可能となるなど、大きな成果が得られた。この成果は、柔軟なインタフェースを構築する上で重要な意味を持つものである。

その一方で、本研究における新たな問題も明らかになってきた。それは、計測された手指動作が意識的に行われたものであるかどうかを判断するのが困難であるという問題である。この問題が原因で、ユーザが意図しない偶発的な手動作についてもジェスチャとして認識してしまうという状況が頻繁に発生し、手

動作の安定した認識・理解に向けての大きな障害となっている。これは、手のジェスチャを認識するために手の動作から得られる情報だけしか利用しておらず、ユーザの意識的な動作を検出するために十分な情報が得られていないためである。

この問題に対して、従来研究の中には手指の動作計測と音声認識を併用するマルチモーダルなアプローチを取り入れることにより、ユーザの意図に関する十分な情報を得ることを図った研究例が幾つか見られる [2, 18]。しかしながら、これらのユーザの音声を利用する従来研究では動作を行うごとにユーザの発声が要求されることや、事務所や教室といった静粛な場所での使用には適していないなどの問題があり、必ずしも自然なインタフェースを実現しているとは言えない。

一方、ユーザの音声以外で、マン・マシン・インタフェースのためにしばしば利用される情報はユーザの視線方向や顔の向きである。これらの情報はユーザの意図と密接な関係があることが知られており、また、音声による入力作業と比較してユーザにとって負担がかからない自然な入力となり得る。このため、視線情報や顔の向きの情報を利用した意識的動作の検出は、自然なマン・マシン・インタフェースを実現する上で非常に有効であると考えられる。

以上のように、ユーザの視線方向や顔の向きは、ユーザの行動理解のための重要な手がかりとなる。このうち、視線方向を高精度に計測しようとする場合、特殊な装着型デバイスや目の周辺の高解像度画像が必要とされ、一般的なインタフェースへの応用は困難である。

そこで本研究では、ユーザの顔の向きの計測を目的とし、頭部姿勢を三次元的に実時間で推定することを目指す。

3.1 従来手法の問題点と本研究のアプローチ

頭部姿勢推定手法の分類方法の一つとして、確率論にもとづく時系列フィルタを利用する手法とそれを利用しない手法とに分類する方法がある [7, 14, 8, 28]。一般に、時系列フィルタを利用する手法の場合、推定の不確実性を確率的に考慮することにより、入力画像中の雑音などが推定性能に及ぼす悪影響を大幅に抑制することが可能となっている。

時系列フィルタのうちコンピュータビジョンでもっとも一般的に利用される手法は、カルマンフィルタや拡張カルマンフィルタである。そして、それらを頭部姿勢推定に利用した手法も数多く報告されている [1, 11, 17]。これらの手法において、拡張カルマンフィルタなどのフィルタリング手法は頭部姿勢の追跡性能を向上させる効果が確認されている。しかしながら、これらのフィルタリング手法は基本的に状態量の確率密度関数が単峰性のガウス分布となることを仮定するために、背景などの影響による誤検出や顔の部分的な遮蔽への対処が困難であるという問題も抱えている。

一方で、パーティクルフィルタ（コンピュータビジョンでは Condensation アルゴリズムと呼ばれるこ

ともある [10]）の場合、状態の確率密度関数に単峰性やガウス性という仮定は存在しない。そのため、パーティクルフィルタでは任意の確率密度関数を取り扱うことができ、誤検出や対象の遮蔽が発生するような場合にも、安定して頭部姿勢を推定することが可能となる。最近では、パーティクルフィルタを頭部姿勢の推定に応用した研究についても幾つか発表されている [23, 3, 6]。

その一方で、インタフェースへの応用を前提とした頭部姿勢推定技術であることを考えると、遮蔽などへの対処以外にも重要な条件が幾つか挙げられる。まず、実際のユーザの動きを考慮すると、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が重要となる。それと同時に、ユーザが実世界のある点に注目しているなどの理由で頭部が静止しているときには、十分に高い精度で頭部姿勢を推定することが重要となる。さらに、任意のユーザが使用する環境を想定して、初期化部も含めて完全に自動化されたシステムへ容易に拡張可能であることが要求される。しかしながら、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、これらの要求を十分に満たしているとは言い難い。

これらの問題に対し、本研究における貢献は大きく分けて以下の2点である。

適応的拡散制御の導入 パーティクルフィルタの頭部動作モデルにおける仮説の拡散を適応的に制御することにより、突発的な動作への追従性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実現する。

頭部変形モデルの自動構築 特別な事前準備なしに頭部剛体モデルを高速に自動構築し、そのモデルを用いて頭部姿勢推定を開始する。さらに、頭部姿勢推定と並列して顔の変形に関する情報を蓄積し、その分析結果として頭部変形モデルを自動的に構築する。

本研究で提案する頭部姿勢推定システムの流れは図9に示す通りである。すなわち、ユーザの頭部剛体モデルや変形モデルを構築するための *Model construction step* と、その頭部モデルと入力画像フレームから頭部姿勢と顔変形を逐次的に推定するための *Pose estimation step* により構成されている。以降、3.2節では *Pose estimation step* について説明し、3.3節では *Model construction step* について説明する。

3.2 頭部姿勢推定システム

本節では、頭部モデルと左右2台のカメラ¹からの入力画像を用いて、三次元的な頭部姿勢を実時間で推定するための手法を説明する。

3.2.1 頭部モデル

本手法で用いる頭部モデルは K 個の特徴点を持っており、各々の特徴点は2つの要素により構成され

¹ 本稿ではカメラを2台用いる場合について説明するが、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。

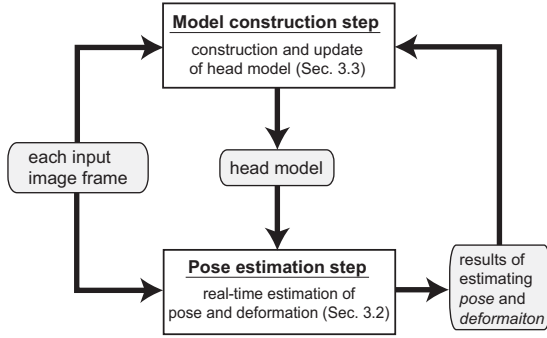


図 9: 頭部姿勢推定システムの流れ

ている。1つはフレーム t における特徴点の三次元座標であり、これはユーザ頭部に固定されたモデル座標系内での座標として表現される。そして、もう1つは特徴点のテンプレート画像であり、左右カメラに対して1枚ずつ、計2枚保有している。ここでは、 M_t を K 個の特徴点の三次元座標で構成された $3K$ 次元形状ベクトルとする。また、 T_L は左カメラに対する K 個のテンプレートの集合を表すものとし、同様に T_R を右カメラに対するテンプレート集合とする。本稿で使用するモデルは、 $K = 10$ で固定されており、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する。

ここで、頭部モデルの形状ベクトル M_t を次式により定式化する。

$$M_t = \bar{M} + \mathcal{M}a_t \quad (14)$$

このとき、 \bar{M} は平均形状ベクトル、 \mathcal{M} は B 個の基底形状ベクトルを並べた $3K \times B$ の基底形状行列、 a_t は \mathcal{M} の B 次元係数ベクトルである。すなわち、形状ベクトル M_t は \mathcal{M} の各列に対応する基底形状ベクトルの線形和により表現され、 a_t の各要素が顔変形のパラメータとなる [8, 15, 6]。このように M_t を表現した上で、 B の大きさを制限することにより、少数のパラメータで顔形状の変化を表現することを可能とするとともに、頭部変形モデルが不自然に変形するのを抑制する効果も与える。本稿では、 B は 5 に設定している。なお、 \bar{M} と \mathcal{M} の計算方法については 3.3.3 節で説明する。

3.2.2 頭部姿勢推定の概要

頭部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、三次元的な頭部姿勢と顔変形を推定する。すなわち、各入力画像フレーム t に対して $(6+B)$ 次元状態ベクトル $x_t = (p_t^T, a_t^T)^T$ を推定する。このとき、 p_t は世界座標系からモデル座標系への並進と回転を表す 6 次元頭部姿勢ベクトルである。姿勢推定にあたっては、3.2.1 節で述べた頭部変形モデルとパーティクルフィルタを利用する。

パーティクルフィルタ [10] では、状態ベクトルの確率密度関数を離散的な重み付きの仮説群により表現す

Estimation of Head Pose and Deformation

1. generate new samples $\{s_t^{(i)}\}$ from $\{s_{t-1}^{(i)}; \pi_{t-1}^{(i)}\}$
2. determine weights $\{\pi_t^{(i)}\}$
 - a. calculate a score $c_t^{(i)}$ using $\mathcal{N}_h(s_t^{(i)})$
 - b. calculate weight $\pi_t^{(i)}$ from the score
3. apply resampling to sample set $\{s_t^{(i)}; \pi_t^{(i)}\}$
4. aggregate samples to have a result x_t

図 10: 頭部姿勢と顔変形の推定の流れ

る。本手法で用いる仮説群 $\{(s_t^{(i)}; \pi_t^{(i)})\} (i = 1 \dots N)$ は、 $(6+B)$ 次元状態空間内の N 個の仮説 $s_t^{(i)}$ と、各仮説に対応する重み $\pi_t^{(i)}$ で構成される。これにより、この仮説群は任意の非ガウス性の確率密度関数を近似することが可能となる。

推定の流れを図 10 に示す。まず、直前のフレーム $t-1$ の仮説群 $\{(s_{t-1}^{(i)}; \pi_{t-1}^{(i)})\}$ と式 (15) の表す動作モデルを用いて新たな仮説を N 個生成する。

$$s_t^{(i)} = s_{t-1}^{(i)} + \tau v_{t-1} + \omega \quad (15)$$

この動作モデルでは隣り合うフレーム間での頭部姿勢の変化が等速直線運動に従うことを仮定している。 $s_{t-1}^{(i)}$ は $\{(s_{t-1}^{(i)}; \pi_{t-1}^{(i)})\}$ の中から選択された仮説であり、 τ はフレーム間の時間間隔、 v_{t-1} は直前のフレーム $t-1$ で計算されている姿勢速度、 ω はシステム雑音である。

このとき、 ω は平均が 0 である $(6+B)$ 次元ガウス雑音で、共分散行列のうち頭部姿勢に関する要素、すなわち左上 6×6 の要素は姿勢の速度に応じて適応的に制御される。その詳細と効果については 3.2.3 節で述べる。一方、共分散行列の残りの部分、すなわち顔変形に関する部分は対角行列であり、対角成分を B 次元ベクトル $\beta = \kappa \mu$ とする。この B 次元ベクトル μ は、顔変形パラメータである a_t の各要素の標準偏差を並べたベクトルであり、その計算方法については 3.3.3 節で述べる。なお、 κ は経験的に 0.5 に設定している。

次に、新たな仮説 $s_t^{(i)}$ に対応する重みをカメラ h からの入力画像フレーム I_t^h をもとに計算する。仮説 $s_t^{(i)}$ が与えられたとき、正規化相関にもとづく関数 $\mathcal{N}_h(s_t^{(i)})$ を次の手順に沿って適用する。ここではまず、 $s_t^{(i)}$ の形状成分にあたる $a_t^{(i)}$ と式 (14) を用いて頭部モデルの形状を変形し、次いで、 $s_t^{(i)}$ の姿勢成分である $p_t^{(i)}$ によって変形後のモデルを移動する。その後、この頭部モデル上の K 個の三次元特徴点を画像フレーム I_t^h に投影し、その投影点周辺の画像とテンプレート集合 T_h の中の対応するテンプレートとの間のマッチングスコアを正規化相関により計算する。このマッチングスコアを K 個の投影点について計算し、それらの和を $\mathcal{N}_h(s_t^{(i)})$ の出力値とする。

以上のように定義した $\mathcal{N}_h(s_t^{(i)})$ をすべての画像平面 h に対して適用し、式 (16) から合計スコア $c_t^{(i)}$ を

計算する。次いで、式 (17) に示すようなガウス関数を適用し、重み $\pi_t^{(i)}$ を計算する。最後に、 $\pi_t^{(i)}$ の合計が 1 になるように $\pi_t^{(i)}$ を正規化する。

$$c_t^{(i)} = \sum_{h \in \{L, R\}} \mathcal{N}_h(s_t^{(i)}) \quad (16)$$

$$\pi_t^{(i)} \propto \exp\left(-\frac{(2K - c_t^{(i)})^2}{2\sigma^2} - \frac{1}{2} \sum_{b=1}^B \left(\frac{a_{t,b}^{(i)}}{\mu_b}\right)^2\right) \quad (17)$$

ここで、 σ はガウス関数で用いる標準偏差であり、経験的に 3.0 に設定している。また、式 (17) では、 $a_t^{(i)}$ についての関数を乗じることにより、モデルが過度に変形しないように拘束を与えている。このとき、 $a_{t,b}^{(i)}$ は $a_t^{(i)}$ の b 番目の要素、 μ_b は μ の b 番目の要素である。

最後に、仮説群 $\{(s_t^{(i)}; \pi_t^{(i)})\}$ を用いて、現在の姿勢 p_t と顔変形 a_t を表す状態ベクトル x_t を計算する。ここでは、重みが最大となる仮説の近傍に属する仮説集合の加重平均により x_t を求める。

$$w_t^{(i)} = \begin{cases} 1 & \text{if } \|s_t^{(i)} - s_t^{(M)}\| < d \\ 0 & \text{else} \end{cases} \quad (18)$$

$$x_t = \frac{\sum_{i=1}^N s_t^{(i)} \pi_t^{(i)} w_t^{(i)}}{\sum_{i=1}^N \pi_t^{(i)} w_t^{(i)}} \quad (19)$$

このとき、 $s_t^{(M)}$ は最大の重みを持つ仮説である。現在のところ、 d の値は経験的に決定している。

また、次の画像フレームでの姿勢推定のために x_t の速度 v_t を計算しておく。

$$v_t = \frac{x_t - x_{t-1}}{\tau} \quad (20)$$

v_t の要素のうち、顔変形に対応する最後の B 個の要素は 0 に設定する。これは、顔変形のパラメータの変化は等速直線運動の仮定には十分に適合しないためである。

3.2.3 適応的拡散制御

本推定手法では、動作モデル (式 (15)) の中で仮説の拡散に相当するランダム雑音 ω を適応的に制御することにより、ユーザの突発的な動きに対する追従性と頭部静止時の精度との両方を改善することを目指す。

一般に、ユーザが突発的な動きをした場合、時系列フィルタで定義した動作モデルの仮定 (本手法では、隣り合う時刻間での等速度・等角速度運動) から大きく外れるという事態が頻繁に起こる。このように動作モデルの仮定から大きく外れるということが、追従性の低下や追跡失敗の大きな原因となっている。

この解決法として、動作モデルで加えるランダム雑音を大きく設定することにより、十分広範囲に仮説を拡散することが挙げられる。しかしながら、パー

ティクルフィルタの場合、あまり広範囲に仮説を拡散すると確率密度関数の近似精度が全体的に低下するために、結果として推定精度の劣化を引き起こすこととなる。

これに対して、本手法では動作モデルによる仮説の拡散を適応的に制御することにより問題の解決を図ることとする。すなわち、必要なときだけ広範囲に仮説を拡散するための機構を導入する。以下、適応的拡散制御の詳細について説明する。

適応的拡散制御の詳細

本手法の適応的拡散制御は、 ω (式 (15)) の取り得る範囲を姿勢運動の速度に応じて増減することにより実現する。

本手法では、 ω を平均が零ベクトルのガウス雑音とする。そして、 ω の共分散行列のうち、頭部姿勢に関係する要素、すなわち左上 6×6 の要素を適応的に制御することとする。この制御の方法は以下の通りである。

ω の共分散行列の左上 6×6 の要素を 6 次元対角行列とし、その対角成分を左上から順に $\beta_x^2, \beta_y^2, \beta_z^2, \beta_\phi^2, \beta_\theta^2, \beta_\psi^2$ とする。このとき、次の 6 次元ベクトル δ_t を定義する。

$$\delta_t = (\beta_x, \beta_y, \beta_z, \beta_\phi, \beta_\theta, \beta_\psi)^T \quad (21)$$

この 6 次元ベクトル δ_t を制御することにより仮説拡散の制御を行う。以降、この 6 次元ベクトルを拡散制御ベクトルと呼ぶこととする。

本手法では、動作モデルによる姿勢予測の不確実性が姿勢変化の大きさに比例するものと仮定する。この仮定により、次式に示すように、拡散制御ベクトル δ_t は各速度成分の絶対値に関して線形に増加するものとする。

$$\delta_t = \Gamma \dot{p}'_{t-1} + \gamma \quad (22)$$

ここで、 \dot{p}'_{t-1} は、 p_{t-1} の速度 \dot{p}_{t-1} に対して各要素の絶対値を取った 6 次元ベクトルである。

このとき、 6×6 行列 Γ と 6 次元ベクトル γ については、実際の追跡結果にもとづいて決定する。ここでは、式 (15) の中で ω が現フレームの推定姿勢と前フレームからの予測姿勢との誤差に相当することに注目し、この誤差を表現するために与えられるべき拡散制御ベクトル δ_t を考える。そして、この δ_t が式 (22) から得られるように Γ と γ を最適化する。

まず、速度の各成分の絶対値 \dot{p}'_{t-1} と与えられるべき拡散制御ベクトル δ_t を計算する。このとき、 δ_t は次式により計算する。

$$\delta_t = \{p_t - (p_{t-1} + \tau \dot{p}_{t-1})\} \times \alpha \quad (23)$$

これは、時刻 $t-1$ の推定結果から予測される時刻 t での予測姿勢と実際の時刻 t での推定結果との誤差に比例する拡散を与えることを意味する。本研究では経験的に $\alpha = 2$ で固定する。

以上のように計算された \dot{p}'_{t-1} と δ_t の組に対して、次式により Γ と γ を最適化する。

$$\min_{\Gamma, \gamma} \sum_t \|\delta_t - (\Gamma \dot{p}'_{t-1} + \gamma)\|^2 \quad (24)$$

実際には擬似逆行列を用いた行列計算により求める。

なお、この計算は推定姿勢 p_t を得るごとに逐次的に行うことが可能である。このため、各フレームで推定した姿勢を蓄積してパラメータを逐次更新することにより、ユーザの行動履歴に応じたパラメータを獲得することが可能となる。

3.2.4 評価実験：適応的拡散制御

本節では、本推定手法について評価実験を行い、その結果について述べる。特に、適応的拡散制御を適用した場合としなかった場合の性能を比較した。

本実験では、CPUとしてIntel Pentium4 3.0GHzを、OSとしてWindows XPを搭載した汎用PCを1台使用した。入力画像として、2台のIEEE1394カメラ(Point Grey Research社製Flea)により取り込まれた640×480画素の画像を使用した。また、画像テンプレートの大きさは16×16、パーティクルフィルタの仮説の総数は1000であった。このとき、本推定手法は毎秒30フレームで動作した。なお、本実験では蓄積データの違いによる性能差を排除するため、適応的拡散制御で使用する Γ として6×6単位行列を、 γ として要素がすべて1の6次元ベクトルを使用した。ここで設定した値は、3.2.3節の手法から得られる結果の典型的な値である。また、純粋に推定性能を比較するために、頭部モデルはあらかじめ準備しておいたモデルを用いた。

入力画像列としては2種類用意した。画像列1では、ユーザは比較的ゆっくり動作し、時折静止している。一方、画像列2では、図11に示すように、ユーザは比較的高速に動作している。このとき、いずれの画像列にも、手などによる意図的な遮蔽や照明・背景変化などは含まれていない。いずれの画像列も20秒(600フレーム)分のデータを含んでいる。これと同時に、磁気センサであるPolhemus社のFASTRAKを頭部に装着して頭部姿勢を計測し、その計測値と本手法による推定値とを比較した。

本実験の適応的拡散制御を適用しない手法では、3.2.3節の拡散制御ベクトル δ_t を固定するという方法をとった。本稿では、 $(5, 5, 5, 5, 5, 5)^T$ に固定した場合の推定結果を掲載する。これは、予備実験の結果、精度と追従性について総合的に良好な結果を得たと判断されたものである。

図11は画像列2に対する推定結果である。濃い太線(with)が提案手法である適応的拡散制御を適用した場合の推定値、薄い太線(gr.tr.)が磁気センサによる計測値である。また、細線(w/o)は適応的拡散制御を適用しなかった場合の推定値である。この図が示すように、適応的拡散制御を適用しない場合では頭部を追従することはできたが、推定値があまり安定せずに細かなブレが生じた。これに対し、適応的制御を適用した手法ではユーザの動作の激しさにかかわらず安定した頭部姿勢推定を実現することができた。図12の上段は画像列2の推定結果画像の一例である。

また、表3は画像列1(比較的低速な動作)に対する推定値の平均二乗誤差の平方根、表4は画像列

	x	y	z	roll	yaw	pitch
W/O Adap.Ctrl	3.24	2.33	3.87	0.40	3.67	2.34
With Adap.Ctrl	1.25	1.91	2.71	0.25	1.55	1.70

表 3: 画像列 1 (低速) に対する推定値の平均二乗誤差の平方根 (x, y, z[mm] and roll, yaw, pitch[degree])

	x	y	z	roll	yaw	pitch
W/O Adap.Ctrl	3.70	4.03	5.11	1.01	3.95	2.89
With Adap.Ctrl	3.42	3.34	4.92	0.87	2.86	2.34

表 4: 画像列 2 (高速) に対する推定値の平均二乗誤差の平方根 (x, y, z[mm] and roll, yaw, pitch[degree])

2(図11の高速な動作)に対する推定値の平均二乗誤差の平方根である。この表からも、適応的拡散制御が推定精度の向上に有効に作用したことがわかる。

図12に本システムによる追跡結果の画像を示す。上段は画像列2に含まれる画像の推定結果である。一方、下段の推定時には、これまでの実験で使用した画像列とは異なる画像列を用いており、手による意図的な遮蔽や照明変化を含む画像列を使用した。

このように、様々な頭部姿勢や環境下において、安定した頭部姿勢推定を実現している。本システムによる追跡の様子を動画で記録したものはウェブ²上にて閲覧できる。

3.3 頭部変形モデルの自動構築

本節では、ユーザ頭部の変形モデルを獲得するための手法について説明する。なお、本節は図9のModel construction stepでの処理に対応する。

従来の頭部姿勢推定に関する研究においても、顔形状の変化を表現することが可能なモデル、すなわち頭部変形モデルを利用した手法は幾つか提案されており、顔形状の変化を伴う場合にも安定した推定を実現している[11, 8, 9, 6, 15]。

しかしながら、これらの手法は、姿勢推定の前にあらかじめ頭部変形モデルを準備しておく必要があるという問題を抱えている。すなわち、頭部変形モデルの構築作業を完了するまでは頭部姿勢を推定することが不可能である。また、推定結果を用いて逐次的に頭部モデルを更新するような機構も備えていない。

これに対し本研究では、頭部変形モデルの構築と頭部姿勢の実時間推定を同時に実行するための手法を提案する。さらに、頭部姿勢および顔変形の推定性能を逐次的に向上させるための機構も実現する。

本モデル構築手法は、図13に示すように、初期

² <http://www.hci.iis.u-tokyo.ac.jp/~oka/>

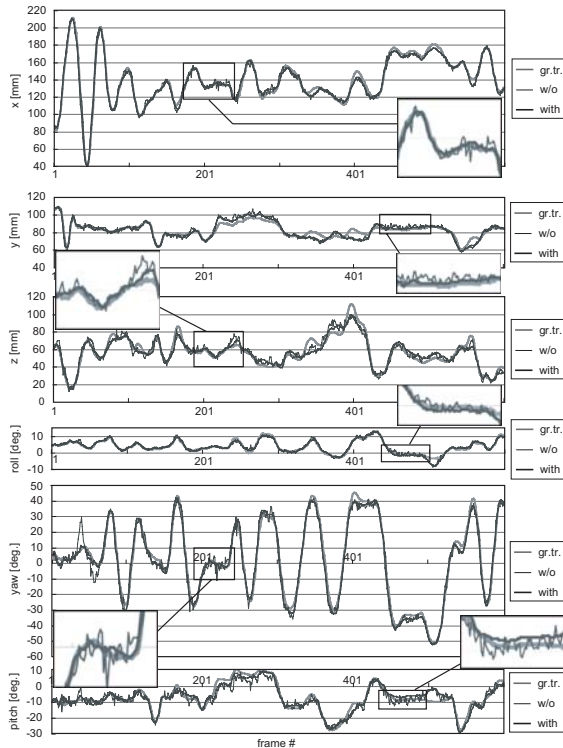


図 11: 画像列 2 (高速) の推定結果 (濃い太線: 適応的拡散制御ありの場合, 細線: 適応的拡散制御なしの場合, 薄い太線: 磁気センサによる計測値.)

化部 (*Initialization step*), および逐次更新部 (*Pose estimation step*, *Feature-point recalculation step*, *Deformation analysis step*) により構成されている。

まず, 初期化部 (*Initialization step*) では暫定的な頭部モデルとして頭部剛体モデルを自動的かつ高速に構築する (3.3.1 節)。初期化が完了すると, 3つの段階から成る逐次更新部に移行する。最初の段階 (*Pose estimation step*) では, 3.2 節で述べたように, 入力画像から実時間で頭部姿勢 p_t と顔変形 a_t を推定する。第 2 段階 (*Feature-point recalculation step*) では, 第 1 段階で推定された p_t と a_t を利用して, 正確な特徴点位置を再計算する (3.3.2 節)。最後に, 第 3 段階 (*Deformation analysis step*) で, 式 (14) の平均形状ベクトル \bar{M} と基底形状行列 \mathcal{M} を主成分分析により計算し, a_t の各要素をパラメータとする頭部変形モデルを構築する (3.3.3 節)。その後, 新たに構築した頭部変形モデルを次フレームでの頭部姿勢推定, すなわち, *Pose estimation step* で使用する。これにより, 頭部変形モデルを随時更新するとともに, 頭部姿勢と顔変形の推定性能を逐次的に向上させるような機構を実現する。

3.3.1 頭部モデルの初期化

初期化部では, 暫定的な頭部モデルとして, ユーザ頭部の三次元的な剛体モデルを自動的かつ高速に



図 12: 頭部姿勢推定結果画像の一例

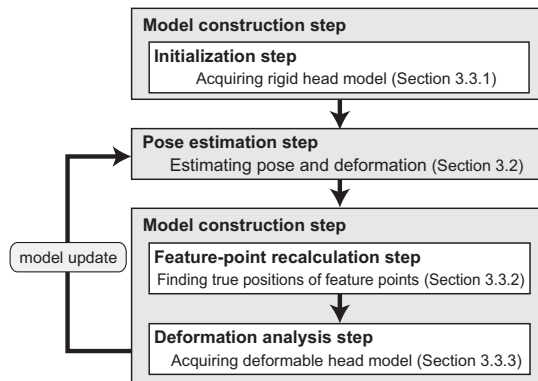


図 13: 頭部変形モデル構築の流れ

構築する。ここでは, オムロン社の OKAO ビジョンライブラリを利用する [12]。OKAO ビジョンライブラリは入力画像から顔と 6 個の顔特徴点 (両目の両端と口の両端) を検出することが可能である。残りの $(K - 6)$ 個の特徴点については, 事前知識にもとづく位置関係や Shi らの手法 [24] を用いて検出する。

まず, 以上の処理を左カメラからの入力画像に適用して K 個の特徴点を検出する。次に, これらの特徴点に対応する点を右カメラからの入力画像中で探索するために, エピポーラ線上でのテンプレートマッチングにもとづく探索を行う。こうして得られた K 組の二次元的な特徴点位置座標から, K 個の特徴点の三次元位置座標を計算する。そして, 頭部に固定されたモデル座標系を定義するとともに, モデル座標系内での顔の三次元形状 M を決定する。最後に, この M とテンプレート集合 T_L, T_R を頭部モデルとして登録する。

なお, ここで得られる剛体モデルでは顔変形を表す a_t を推定するのは不可能である。そこで, この初期化直後で剛体モデルしか存在しない場合には, a_t

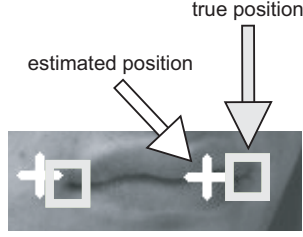


図 14: 特徴点の推定位置と正確な位置

を 0 に設定する．

3.3.2 頭部姿勢の推定結果を用いた特徴点位置の再計算

本節では，図 13 における *Feature-point recalculation step* について説明する．

正確な頭部変形モデルを構築するためには，各特徴点の正確な三次元位置を獲得しなければならない．このとき，式 (14) と *Pose estimation step* で推定された a_t を用いて計算される形状 M_t の各特徴点位置は，良い近似にはなりうるが，必ずしも正確な位置と一致するとは限らない (図 14) ．

そこで本手法では，*Pose estimation step* (3.2 節) で推定した頭部姿勢 p_t と顔形状 M_t を参照しながら，改めて各特徴点の正確な位置を計算する．ここで， M'_t をモデル座標系における K 個の特徴点の正確な三次元位置座標を表す $3K$ 次元ベクトルとし，以下， M'_t を真形状と呼ぶ．

本手法では， M'_t についての誤差関数 E_t を定義し， E_t を最小化することによって M'_t を決定する．

$$E_t = E_t^I + \epsilon E_t^M \quad (25)$$

ここで， E_t^I は特徴点近傍の見えに関する誤差を表し， E_t^M は推定形状 M_t を用いた幾何学的な拘束を与える．このとき， ϵ は定数であり，本稿では経験的に 2000 に固定している．なお，最小化の方法は Gokturk らの手法 [8] と同様であるため，計算の詳細についてはそちらを参照されたい．以下， E_t^I と E_t^M について具体的に説明する．

まず， E_t^I を定義する前に， E_t^I で使用する関数 \mathcal{P}_h を定義する．これは，頭部姿勢 p_t に応じた並進や回転を真形状 M'_t に適用したのち， M'_t の各特徴点を画像平面 h に投影する関数である．

$$m_{h,t} = \mathcal{P}_h(p_t, M'_t) \quad (26)$$

ここで， $m_{h,t}$ は K 個の投影点の二次元座標で形成される $2K$ 次元ベクトルである．

また， K 次元輝度ベクトル $I_t^h(m_{h,t})$ についても定義する．このとき， $I_t^h(m_{h,t})$ の k 番目の要素は，入力画像 I_t^h における $m_{h,t}$ の k 番目の二次元座標での輝度を表す．

これらの定義を用いて， E_t^I を次のように定義する．

$$E_t^I = \sum_{\substack{\text{ROI} \\ h \in \{L, R\}}} \left\{ \rho \|I_t^h(m_{h,t}) - I_{t-1}^h(m_{h,t-1})\|^2 + \|I_t^h(m_{h,t}) - I_1^h(m_{h,1})\|^2 \right\} \quad (27)$$

ここで，式 (27) の第 1 項は，現在の画像 I_t^h 中の K 個の注目領域 (Regions Of Interest ; ROIs) と直前の画像 I_{t-1}^h 中の注目領域との見えの誤差を表す標準的な誤差関数である．一方，第 2 項は現在の画像 I_t^h 中の注目領域と追跡開始時の画像 I_1^h 中の注目領域との見えの誤差を表す．この項は，Gokturk らの手法 [8] でも使用されており，特徴点のドリフトを回避する効果が報告されている．また， ρ は第 1 項と第 2 項の間の比率を表す．現在は経験的に， ρ を 4 に，注目領域の大きさを 16×16 画素に設定している．

次に， E_t のもう一方の項 E_t^M について定義する．これは，推定形状 M_t の各点の近傍領域で真形状 M'_t の各点を探索するための項である．

$$E_t^M = \|M'_t - M_t\|^2 \quad (28)$$

この項の導入により，ユーザの頭部姿勢が大きく変化した場合であっても，特徴点を見失うことなく追跡できるようになる．また，頭部変形モデルの更新により M_t が正確な形状に近づくにつれて，より安定して M'_t を計算することが可能となる．

以上の定義により得られる E_t (式 (25)) を最小化することにより，多くの場合は真形状 M'_t を正しく決定することができる．その一方で， M'_t の中に剛体運動に対応する成分が含まれる場合も存在する³．この成分を除去せずに頭部変形モデル構築し，その頭部モデルを用いて頭部姿勢を推定した場合，頭部剛体運動と顔変形を適切に分離することが不可能になる．そこで， M'_t に含まれる剛体運動成分を除去するために，次のような処理を用いる [15]．まず， M'_1 から M'_{t-1} までの平均形状 \bar{M}' を計算する．その後， M'_t と \bar{M}' の対応する点同士の距離の二乗和が最小になるように， M'_t に三次元的な並進と回転を適用し，その結果を改めて M'_t として定義する．以上の方法により， M'_t に剛体運動成分が含まれる場合には，その大部分を除去することが可能である．しかしながら， M'_t が正しく計算されている場合には悪影響を及ぼす可能性もある．それゆえ本手法では，この処理が必要とされる場合，ここでは M'_t と M_t との距離が一定の閾値を超えた場合のみ，以上の処理を適用するものとする．

3.3.3 主成分分析による頭部変形モデルの構築

本節では，図 13 の *Deformation analysis step* とし，3.3.2 節で獲得した真形状 M'_t をもとに平均形状ベクトル \bar{M} と基底形状行列 \mathcal{M} を主成分分析を用いて計算する．特に本手法では，個々の入力データに対して高速かつ増分的に基底行列を計算するこ

³ その多くは，*Pose estimation step* における頭部姿勢 p_t の推定誤差が原因である．

とが可能な主成分分析手法, Candid Covariance-free Incremental PCA (CCIPCA) [30] を使用する.

本手法で正確な頭部変形モデルを構築するためには, できるだけ信頼性の高いデータを使用する必要がある. そこで, 推定姿勢 p_t から判断して, ユーザがカメラの方を向いている場合の M'_t だけを分析対象として使用する. この場合には, ユーザが横を向いている場合と比較して, 特徴点追跡に成功している可能性が高いためである. このような M'_t に対して, まず最初に, このフレームまでに得られた M'_t の平均ベクトル \bar{M} を計算する. 次に, M'_t と \bar{M} との間の距離を計算し, その距離が一定以上であるという条件を満たす場合にのみ, M'_t を CCIPCA への入力とする. このような条件は, M'_t が変形に関する情報を有しているかどうかを判断するために必要となる.

以上で適用した CCIPCA の結果, 複数の基底ベクトルが得られる. これらの基底ベクトルのうち最初の B 個のベクトルを横に並べたものを $3K \times B$ 基底行列 \mathcal{M} として定義する. このとき, CCIPCA により得られた固有値の平方根を大きい順に B 個並べた B 次元ベクトル μ も定義する. μ の各要素は, 顔変形パラメータである a_t の各要素の標準偏差に対応する. それゆえ本手法では, μ を式 (15) におけるシステム雑音の分散や式 (17) における各仮説の重みを決定するために使用する.

3.3.4 評価実験: 頭部変形モデル

提案手法の性能を評価するために実験を行った. 本実験の設定は 3.2.4 節の評価実験と同様である. このとき, 本システムは毎秒 30 フレームで動作した.

本実験で用意した画像列では, ユーザが頭部を動かしながら顔の変形を時折行っている. この画像列は 60 秒 (1800 フレーム) 分のデータで構成されている. このうち最初の 120 フレームでは, 初期化時に自動的に獲得される頭部剛体モデルを用いて頭部姿勢を推定した. これは, 姿勢推定の開始直後は, CCIPCA によって計算される基底行列 \mathcal{M} の信頼性が低いからである. その後の残り 1680 フレームでは, 逐次的に更新される頭部変形モデルにより頭部姿勢と顔変形を推定した. なお, この 1680 フレームのうちの最初の約 1100 フレームでは, ユーザは口の開閉やしがめ面といった顔変形を行いながら頭部運動を行っている. これに対し, 残りの約 600 フレームでは, ユーザは顔変形をほとんど行わずに頭部の剛体運動だけを行っている.

また, 上記の推定結果と比較する目的で, 同じ 1800 フレームに対して, 頭部剛体モデルを用いた頭部姿勢推定も行った. この頭部剛体モデルは, 3.3.1 節で述べた初期化部の処理により構築されたものであり, 変形モデルを用いる実験の最初の 120 フレームで使用するものと同じモデルである. このようにして得られた 2 種類の頭部姿勢推定結果を比較した.

図 15 に頭部剛体モデルと頭部変形モデルをそれぞれ用いた場合の推定結果画像を示す. また, それぞれの場合に対する推定結果の時系列データを図 16

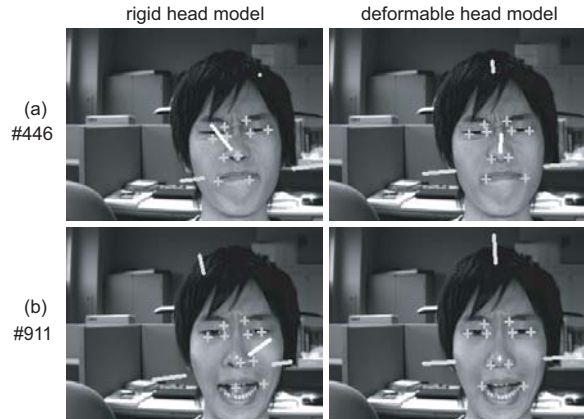


図 15: 頭部剛体 / 変形モデルを用いた頭部姿勢推定結果画像の一例

に示す. 図 16 において, 細い線は剛体モデルを用いた場合の推定結果であり, 太い線は変形モデルを用いた場合の推定結果である. 図 16 を見ると, 第 121 フレームから第 1200 フレーム付近までは両方の推定結果に明確な相違が見られる. 図 15 を参照すると, 頭部変形モデルを用いた方がより正しく推定できていることが明らかである. すなわち, 頭部変形モデルは, 剛体モデルとは異なり, 顔形状の変化に対して適切に対処可能であることが示された. 一方, 残りの約 600 フレームでは両方の推定結果がほぼ一致している. このことから, 頭部変形モデルが不要な顔変形を伴うことなく, 正しく頭部剛体運動を推定できていることがわかる.

4 計測技術のインタフェースへの応用

第 2 節と第 3 節で, 人間動作の計測技術に関する本研究の取り組みについて説明してきた. これらの中でも説明したように, 各計測技術は「インタフェースへの応用」を前提として開発してきたものである. そこで本節では, 本研究で開発した動作計測技術をインタフェースに適用した例を紹介し, 本研究の動作計測手法が実際にインタフェースに応用可能であることを示す.

一つの例が, 図 17 に示す両手描画システムである [4]. これは図 1 の机型インタフェース EnhancedDesk のアプリケーションの一つである. この描画システムでは, 2.3.2 節のシンボリックジェスチャ認識を利用して, ユーザの右手の人差し指で描いた図形と同じ位置で同じ形と大きさを持つ電子図形を表示することができる. 例えば, ユーザが指先で机の上に四角形に似た軌跡を描くと, システムはこれが四角形であると認識し, さらにユーザの指先の軌跡から四角形的位置と大きさを計算し, 四角形が電子図形として机面上に表示される. また, ユーザが右手の親指と人差し指でつまむ動作を行った場合, 親指と人差し指の間隔に基づいて「つまむ」と「はなす」のモー

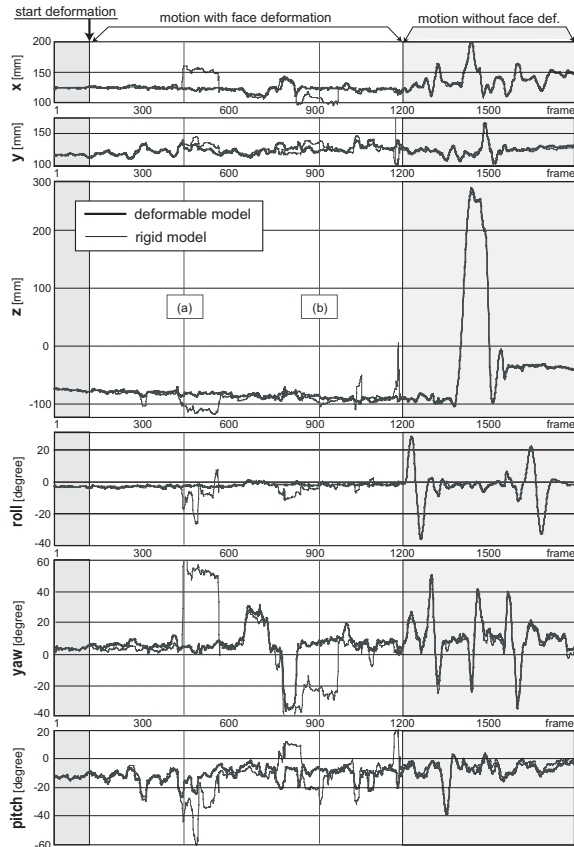


図 16: 頭部剛体 / 変形モデルを用いた頭部姿勢推定の結果

ドを判別し、電子図形のダイレクトマニピュレーションを行うことが可能である。

さらに、ユーザは自身の左手の位置に表示されているメニューの一つを指定してから、そのメニューの操作モードにしたがって右手で図形を描くことや操作対象の図形を指定するといった操作も可能となっている。このメニューはパイメニューと呼ばれる階層構造を持つ扇形のメニューであり、左手の追跡結果に基づいて常にユーザの左手の位置に表示される。このメニュー操作によって、コピーやペースト、色の変更といった操作を実現している。以上の機能を組み合わせて描画作業を行うことにより、GUIでの描画ツールと比較して高速で直観的に描画することが可能となる。

もう一例、2 台の EnhancedDesk における遠隔地間共同作業用アプリケーションを紹介する [32]。ここでは、共同作業において重要な要素とされるゲイズアウェアネス⁴を確保するために頭部姿勢情報を利用する。

この研究では、図 18 に示すようなシステムを使

⁴ 相手がどこを見ているのか、何に注目しているのかという視線情報を知覚すること。

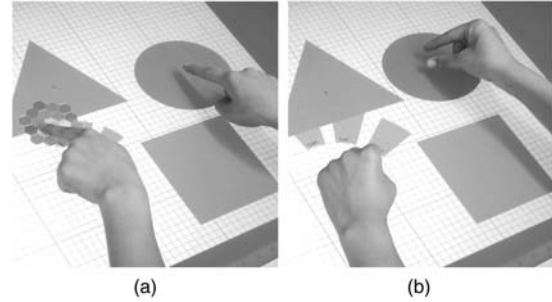


図 17: EnhancedDesk における両手描画システム

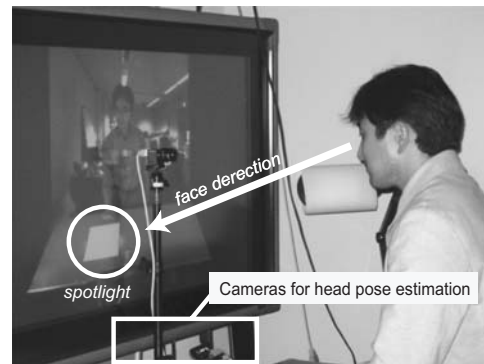


図 18: 2 台の EnhancedDesk における頭部姿勢情報の利用

用する。まず、ユーザ前方のプラズマディスプレイには相手側の作業環境（机）の様子を表示している。そして、ユーザの下方にはカメラが設置されており、ユーザの顔の向き（頭部姿勢）を実時間で計測している。ここで、ユーザがプラズマディスプレイに表示された相手側の机を見たとき、システムはユーザが机のどの位置を見ているのかを自動的に認識し、相手側の机上を実際にスポットする。これにより、自分が机上のどこに注目しているのかを相手に伝えることができ、ゲイズアウェアネスを擬似的に実現することが可能となる。このような機構により遠隔地における共同作業の効率の向上を目指している。

このように、本研究の人物動作計測技術は、実際のインタフェースに利用可能であることがわかる。

5 おわりに

本研究では「柔軟なインタフェース」の実現に向けて、実時間性や安定性などの面でインタフェース用途に適した人間動作計測技術について研究を行ってきた。その中で、指先位置や頭部姿勢の実時間計測技術を提案し、評価実験からその有効性を証明することができた。さらには、この計測技術をインタフェースに応用した例を紹介した上で、提案手法が実

際にインタフェースに適用可能であることを示した。

今後、本研究を発展させる上で、より安定したユーザ行動の理解に向けて、手指・頭部姿勢以外の計測についても検討を進める必要がある。例えば、全身の動作や人間の移動履歴なども利用することが考えられ、このとき、これらの動作を安定して計測するための技術が重要となってくる。

また、様々な器官・モードを計測したとして、それらを統合するためのフレームワークに関する問題が残されている。本稿には掲載できなかったが、本研究ではHMMを用いたフレームワークの検討も行った。しかしながら、精度や実時間性の面で十分な成果を得るには至らなかった。「人間行動の理解」を目的として考える場合、これらの点に関する検討・研究が非常に重要なものとなる。

謝辞

本研究の一部は、文部科学省科学研究費補助金特定領域(C)(2)「人間の意図・行動理解に基づく柔軟なヒューマン・マシン・インタラクションの実現」(課題番号13224051)の助成により行った。また、本研究の一部にはオムロン株式会社の顔検出・顔器官検出技術を利用した。

参考文献

- [1] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, "Visually controlled graphics," *IEEE Trans. PAMI*, vol.15, no.6, pp.602-605, 1993.
- [2] R. Bolt, "Put-that-there," *Computer Graphics*, Vol.14, pp.262-270, 1980.
- [3] B. Braathen, M.S. Bartlett, G. Littlewort, E. Smith, and J.R. Movellan, "An approach to automatic recognition of spontaneous facial actions," *Proc. FG 2002*, pp.360-365, 2002.
- [4] X. Chen, H. Koike, Y. Nakanishi, K. Oka, and Y. Sato, "Two-handed drawing on augmented desk system," *Proc. AVI 2002*, pp.219-222, May 2002.
- [5] J. Crowley, F. Berard, and J. Coutaz, "Finger tracking as an input device for augmented reality," *Proc. FG '95*, pp.195-200, 1995.
- [6] F. Dornaika and F. Davoine, "Head and facial animation tracking using appearance-adaptive models and particle filters," *Proc. CVPRW 2004*, vol.10, 2004.
- [7] A. Gee and R. Cipolla, "Fast visual tracking by temporal consensus," *Image and Vision Computing*, vol.14, pp.105-114, 1996.
- [8] S. Gokturk, J. Bouguet, and R. Grzeszczuk, "A data-driven model for monocular face tracking," *Proc. ICCV 2001*, pp.II-701-708, 2001.
- [9] K. Hiwada, A. Maki, and A. Nakashima, "Mimicking video: real-time morphable 3D model fitting," *Proc. Symposium on Virtual Reality Software and Technology*, pp.132-139, 2003.
- [10] M. Isard and A. Blake, "Condensation- conditional density propagation for visual tracking," *Int. J. Computer Vision*, vol.29, no.1, pp.5-28, 1998.
- [11] T. Jebara and A. Pentland, "Parametrized structure from motion for 3D adaptive feedback tracking of faces," *Proc. CVPR '97*, pp.144-150, 1997.
- [12] S. Lao, T. Kozuru, T. Okamoto, T. Yamashita, N. Tabata, and M. Kawade, "A fast 360-degree rotation invariant face detection system," *Demo session of ICCV 2003*, 2003.
- [13] J. Martin and J. Durand, "Automatic handwriting gestures recognition using hidden Markov models," *Proc. FG 2000*, pp.403-409, 2000.
- [14] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement," *Proc. FG 2000*, pp.499-504, 2000.
- [15] I. Matthews and S. Baker, "Active appearance models revisited," *Int. J. Computer Vision*, vol.60, no.2, pp.135-164, 2004.
- [16] S. Ong and S. Ranganath, "Automatic sign language analysis: a survey and the future beyond lexical meaning," *IEEE Trans. PAMI*, vol.27, no.6, pp.873-891, 2005.
- [17] K.R. Park, S.W. Nam, M.S. Lee, and J. Kim, "Gaze detection by estimating the depths and 3D motion of facial features in monocular images," *IEICE Trans. Fundamentals*, vol.E82-A, no.10, pp.2274-2284, 1999.
- [18] V. Pavlovic, R. Sharma, and T. Huang, "Gestural interface to a visual computing environment for molecular biologists," *Proc. FG '96*, 1996.
- [19] V. Pavlovic, R. Sharma, and T. Huang, "Visual interpretation of hand gestures for human-computer interaction: a review," *IEEE Trans. PAMI*, vol.19, no.7, pp.677-695, 1997.
- [20] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, pp.4-16, 1986.
- [21] Y. Sato, Y. Kobayashi, and H. Koike, "Fast tracking of hands and fingertips in infrared images for augmented desk interface," *Proc. FG 2000*, pp.462-467, 2000.
- [22] J. Segan and S. Kumar, "Shadow gestures: 3D hand pose estimation using a single camera," *Proc. CVPR '99*, pp.479-485, 1999.
- [23] J. Sherrah and S. Gong, "Fusion of perceptual cues for robust tracking of head pose and position," *Pattern Recognition*, vol.34, no.8, 2001.
- [24] J. Shi and C. Tomasi, "Good features to track," *Proc. CVPR '94*, pp.593-600, 1994.
- [25] T. Starner and A. Pentland, "Visual recognition of American sign language using hidden Markov models," *Proc. FG '95*, pp.189-194, 1995.
- [26] M. Turk and G. Robertson, "Perceptual user interfaces," *Comm. ACM*, vol.43, no.3, pp.33-34, 2000.
- [27] A. Utsumi and J. Ohya, "Multiple-hand-gesture tracking using multiple cameras," *Proc. CVPR '99*, pp.473-478, 1999.
- [28] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3D tracking using online and offline information," *IEEE Trans. PAMI*, vol.26, no.10, pp.1380-1384, 2004.
- [29] R. Vertegaal, "Attentive user interfaces," *Comm. ACM*, vol.46, no.3, pp.30-33, 2003.
- [30] J. Weng, Y. Zhang, and W. Hwang, "Candid Covariance-Free Incremental Principal Component Analysis," *IEEE Trans. PAMI*, vol.25, no.8, pp.1034-1040, 2003.
- [31] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," *Proc. CVPR '92*, pp.379-385, 1992.
- [32] 鬼頭哲郎, 小林貴訓, 岡兼司, 佐藤洋一, 中西泰人, 小池英樹, "注視点情報を用いた遠隔地間実世界型共同作業支援," *WISS 2004 予稿集*, pp.141-142, December 2004.