

複数指先軌跡の実時間計測とHCIへの応用

岡 兼司[†] 小林貴訓[‡] 佐藤洋一[†] 小池英樹[‡]

[†] 東京大学 生産技術研究所
〒 106-8558 東京都港区六本木 7-22-1
E-mail : {oka, ysato}@cvl.iis.u-tokyo.ac.jp

[‡] 電気通信大学大学院 情報システム学研究所
〒 182-8585 東京都調布市調布ヶ丘 1-5-1
E-mail : {yosinori, koike}@vogue.is.uec.ac.jp

あらまし キーボードやマウスなどの従来型インターフェースよりも快適な次世代インターフェースとして、実世界指向インターフェースに期待が寄せられている。この実現にはユーザの行動を理解することが必要となり、その中でユーザの指先位置を認識する技術は重要な役割を果たす。従来の指先位置認識では、多くの仮定や制約が存在していたため認識精度が不十分であり、実時間処理が困難であることも問題であった。本報告では、ユーザの複数の指先位置を安定に実時間で追跡する手法について紹介する。赤外線画像の各フレームで指先位置を検出するのに加え、フレーム間で複数指先位置の対応関係を考慮することにより、安定かつ高速に複数の指先位置の軌跡を計測する。本稿では、提案手法の内容および追跡性能に関する実験結果を報告し、HCIへの応用について述べる。

キーワード: HCI, Perceptual User Interface, 実時間画像処理, 指先軌跡検出, カルマンフィルタ

Real-Time Tracking of Multiple Fingertips and Its Application for HCI

Kenji Oka[†] Yoshinori Kobayashi[‡] Yoichi Sato[†] Hideki Koike[‡]

[†] Institute of Industrial Science,
The University of Tokyo
7-22-1 Roppongi Minato-ku Tokyo
106-8558, Japan
E-mail : {oka, ysato}@cvl.iis.u-tokyo.ac.jp

[‡] Graduate School of Information Systems,
The University of Electro-Communications
1-5-1 Chofugaoka Chofu Tokyo 182-8585, Japan
E-mail : {yosinori, koike}@vogue.is.uec.ac.jp

Abstract In this work, we introduce a fast and robust method for tracking positions of the center and the fingertips of a user's hand. In particular, our method makes use of infrared camera images for reliable detection of a user's hand even in complex backgrounds, and uses a template matching technique for detecting fingertips in each image frame. Then we consider correspondences of detected fingertips between successive image frames. This contributes to significantly better performance for detecting fingertips even when a hand is moving fast. By using the proposed method, we can measure loci of multiple fingertips successfully in real-time. In this paper, we describe the details of our proposed method, and report the result of the experiment that we conducted for evaluating the method's tracking performance.

Keywords : HCI, Perceptual User Interface, real-time image processing, finger tracking, Kalman filter

*現在、三菱電機株式会社所属

1 はじめに

現在の技術進歩とともに、近年、ユーザとコンピュータとの新たなインターフェースが要求されるようになった。従来のコンピュータを利用した作業ではユーザとコンピュータは対面しているのが普通であったため、インターフェース開発にはユーザがコンピュータを常に意識しているということが前提になっていた。しかし現在では、社会の至るところにコンピュータが存在し、ユーザの実世界の作業を補助することが役割となりつつあるため、ユーザにコンピュータの存在を意識させないインターフェースが求められている。このように、ユーザがコンピュータとのインタラクションにとらわれず、実世界とのインタラクションに集中できるように設計されたインターフェースは実世界指向インターフェースと呼ばれ、近年盛んに研究が行われている。

実世界指向インターフェースを実現するには、ユーザの動作を様々な視点から認識することが必要となるが、その1つに指先の動きを追跡することが挙げられる。人間は実物体を操作するときには手を使うことが多いが、特に細かい作業を行うときには指による操作を行うため、指先位置を正確に認識することは人間の動作を理解する上で重要な役割を果たすことになる。

以上の見地からこれまでも指先位置を認識する研究は行われてきたが、従来の指先位置認識には様々な問題があった。具体的には、手の向きが常に特定の向きにあることを仮定したり、指の本数は常に1本にしておくという制約を設けていたため、人間が自然な振る舞いを行う場合には十分な認識性能を得られないことが問題であった。また、一連の処理をソフトウェアで行っていたため、インターフェースとして利用する上で重要な実時間処理が困難であるという問題も挙げられていた。

そこで本報告では、ユーザの複数の指先位置を安定に実時間で追跡するための手法について紹介する。まず、ユーザの腕を含む手領域を赤外線カメラで撮影することにより、背景の影響を受けにくい赤外線画像を得る。次に、得られた赤外線画像に対して正規化相関に基づくテンプレートマッチングを行うことにより指先位置を検出する。同時に、収縮処理を適用することにより手のひらの中心位置を検出する。

さらに、各フレームで検出された指先位置について、フレーム間の指先位置の対応関係を求める。具体的には、各画像フレームの指先位置を検出する際に直後のフレームでの各指先の予測位置をカルマンフィ

ルタにより計算し、直後のフレームで実際に検出された指先位置との対応関係を求めることにより実現する。これにより、安定かつ高速に複数の指先位置の軌跡を計測することが可能となる。その結果として、各フレームで指先の対応関係を考慮せずに指先位置を検出する場合よりも頑健な指先位置の検出が可能となる。

本報告の構成は次のようになっている。2章において関連研究について説明し、本研究の意義を明らかにする。次いで、3章において赤外線画像の各フレームから指先を認識するまでの手法について述べ、4章において各フレームで認識された指先位置をフレーム間で対応付ける手法を説明する。さらに、5章において指先検出性能の評価実験を行った結果を示し、6章において本研究をHCIに応用する方針について述べる。最後に、7章において本報告のまとめを示す。

2 関連研究

これまでもユーザの手や指先を追跡する手法はいくつか提案されてきた。ここで、それらの手法の代表的なものを紹介し、各手法の限界についても述べる。

まず、ユーザにグローブ状のデバイスを装着させることによって手の形状や位置を認識する研究について紹介する。これらの研究では、関節角や手の位置などをグローブに付属されたセンサにより計測している。このようなデバイスを使用した研究としては、VPL ReserchのData Glove [1]が有名である。この研究では光ファイバや磁力センサを利用して各変量を計測している。

グローブなどのデバイスを使用した場合にはユーザの手の形状や位置を正確かつ高速に追跡することが可能である。しかしながら、デバイスに付属したケーブルなどがユーザに拘束感を与えることが考えられ、特にユーザの自然な使用感が要求されるHCIの分野での応用には適していない。

そのため、ユーザの手や指先を追跡するために画像処理技術を利用した研究が行われてきた。このような研究の典型的な手法の1つに、ユーザの手や指先にマーカーを装着する手法がある。これにより、指先位置などを画像処理により容易に検出することが可能となる。このような研究として、指先に装着したカラーマーカーにより指先位置を検出する手法[2]やカラーマーカーの装着されたグローブを使用することにより手の姿勢や向きを推定する手法[3]が挙げられる。

しかし、マーカーをユーザに装着する手法では、当然ながらユーザに対してマーカーの装着を強制することとなり、デバイスを使用する手法の場合と同様に、ユーザの自然な使用感を実現する上では問題がある。

もう1つの手法として、ユーザの肌の色を抽出したり、背景画像との差分を取ることでユーザの手領域を特定する手法が研究されている。しかし肌色抽出を行う場合、人間の肌の色が必ずしも均一ではないなどの理由により、手領域を明確に抽出できないことが多い。さらに、光源環境が一定でない状況下で使用する場合、手の表面の色は一定とならないので肌色抽出による手領域の抽出は非常に困難となる。また、背景差分を取る手法についても、背景が複雑である場合や背景に肌色に近い色が存在する状況下では安定に手領域を抽出することは困難である。

肌色抽出などによってユーザの手領域を抽出した画像から指先位置や手の位置および向きを推定する手法はいくつか提案されている [4][5][6]。これらの手法では、抽出された手領域の輪郭の形状を基にして指先位置を検出したり、手領域の輪郭を複数の線で近似することによって指の側面の位置を検出するという処理が用いられている。

また、手の動作解析を手形状の3次元モデルを用いることにより行う手法も報告されている [7][8][9]。これらの手法では、1台もしくは複数のカメラによってユーザの手を撮影し、3次元モデルへのマッチングを通して手の姿勢推定を行っている。特に [8]、[9] ではカルマンフィルタを利用して手の3次元位置や関節角を推定している。

しかしながら、これらの手法ではオクルージョンや計算コストの問題が大ききな障害となる。また、3次元モデルの自由度が高いために、複数のカメラ画像を利用した場合であっても、2次元画像から手の状態を推定することは困難なことが多い。

手や指先を追跡、および解析する手法については、本章で紹介した以外にも、現在までに様々な手法が提案されている。[10]や[11]には本章で紹介できなかった手法についても数多く紹介されている。

3 赤外線画像からの複数指先認識

本章では、赤外線カメラで撮影された映像の各フレームから複数の指先を実時間で認識する手法について説明する [12][13]。

3.1 赤外線画像からの手領域抽出

腕を含むユーザの手領域の抽出には、2章で述べたような背景差分や肌色抽出が一般的に用いられている。しかし、これらの手法では光源環境や背景の状態によっては領域を抽出することが困難な場合が多い。そこで、本システムでは赤外線カメラを使用して人の体温の近傍 ($30^{\circ}\text{C} \sim 34^{\circ}\text{C}$) の明度が高くなるように撮影し (図 1(a))、適当な閾値で2値化処理を行うことにより領域を抽出する (図 1(b))。この手法により、背景や照明の変化に影響されず体温に近い領域だけを安定に抽出することが可能となる。画像は 256×220 画素の解像度で取得し、2値化した画像から慣性主軸を計算し、腕全体の傾きを考慮して、先端から 80×80 画素の範囲を手領域として定義する (図 1(c))。画像処理を行う範囲をここで定義した手領域だけに限定することにより、画像処理の高速化を図る。

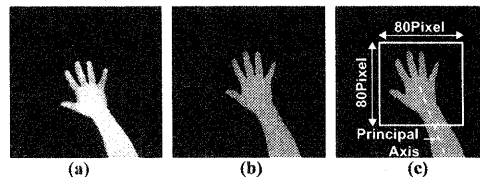


図 1: 手領域の抽出

3.2 指先位置の認識

指先位置を認識する手法について説明する。一般に指先形状の輪郭が円に近いことに基づき、円形テンプレート (図 2(a)) を用いて手領域周辺で正規化相関に基づくパターンマッチングを行うことにより指先位置を検出する。ユーザの指先の大きさはユーザによって異なるため、円形テンプレートは直径 $5 \sim 9$ 画素の範囲で変更可能とした。カメラとユーザの手の距離はほぼ一定でカメラで撮影した際の手の大きさが極端に変わらないことから、ユーザが同じであれば 15×15 画素の一定の大きさのテンプレートで指先位置を検出することが可能である。

パターンマッチングの処理後、誤検出された指先位置を除去する (図 2(b))。誤検出の例として、1個の指先に対して周辺の数多くの点でマッチングに成功する場合が挙げられる。この問題への対処法として、近接した場所で相関値の高い点が多く検出された場合、最も相関値の高い点を残し、残りを指先候補点から除外することにする。

それ以外の誤検出の例として、指の関節が検出され

る場合や孤立点が検出される場合がある。これらの場合に対処するために、テンプレートの周囲の点に対応するピクセルを調べ、指先がテンプレートを突き抜けていたり、周囲に指らしきピクセルが存在しない場合にも指先候補点から除外するという手法をとる。今回のシステムでは、テンプレートの周囲の点として8点(矩形の4頂点および各辺の2等分点)を調べることとした。

最終的に指先候補点を相関値の高い順に最大5つまで採用し、さらに相関値が閾値より低いものを除外して、それらをユーザの指先点として認識する(図2(c))。

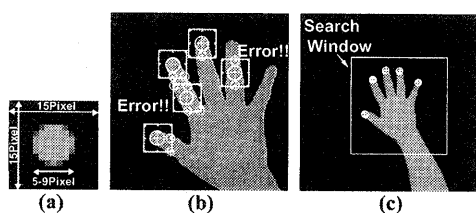


図 2: 指先位置の認識

3.3 手のひら位置の検出

手のひら位置の検出は、腕全体の傾きと手領域の大きさから手首位置を仮定し、手首から先の部分に対して収縮処理を適用することで行う(図3(a))。指が削られ一定の画素数以下になるまで収縮処理を繰り返す(図3(b))、残った領域の重心をとることで手のひらの中心を検出した(図3(c))。

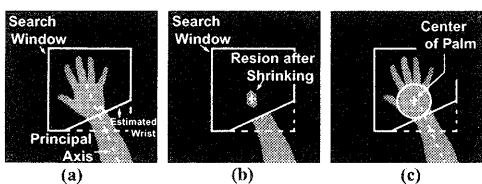


図 3: 手のひら中心の検出

4 フレーム間での対応付けによる複数指先の軌跡の計測

4.1 軌跡処理への拡張

前節では、赤外線画像の各フレーム毎に指先位置を認識する手法について述べた。しかし、この手法には安定性の面でいくつかの問題がある。以下に問題点

を示す。

- (1) 手を高速に動かしたときなどにしばしばテンプレートマッチングの失敗が起り、その間は指先位置の決定が不可能となる。
- (2) テンプレートマッチングの相関値に対して閾値による指先の判定をしているが、この閾値よりも相関値が低い指先も存在する。
- (3) 指を折り曲げた状態でも、先端部分が指先として認識されることがある(図4)。

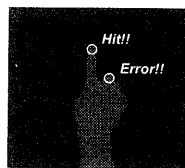
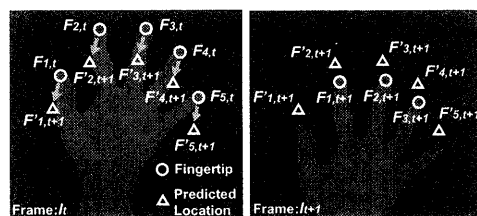


図 4: 指先誤認識の例

また、個々のフレームで複数の指先位置を検出したとしても、あるフレームで検出された各指先を前後のフレームで検出された指先と対応付けなければ、各指先が描く軌跡を計測することはできない。この点は指先によるジェスチャ認識などに応用する上で問題となる。

本報告では、各フレームで検出された指先をその前後のフレームでの指先位置と対応付けることにより軌跡計測を行う手法について提案する。以下に提案手法の概略を述べる(図5)。



(a): フレーム I_t (b): フレーム I_{t+1}

図 5: 軌跡計測手法

まず、時刻 t の画像フレーム I_t (図5(a)) で n_t 個の指先位置 $F_{i,t} (i = 1, 2, \dots, n_t)$ を検出したとする。その際、直後のフレーム I_{t+1} での各指先の n_t 個の予測位置 $F'_{i,t+1}$ を求めておく。次に、フレーム I_{t+1} (図5(b)) で実際に検出された n_{t+1} 個の指先位置 $F_{j,t+1} (j = 1, 2, \dots, n_{t+1})$ を求めておいた予測位置 $F'_{i,t+1}$ と比較し、それらの最適組み合わせを求め

る。こうして決定された組み合わせを対応する指先とみなし、さらに対応の取れた指先位置を追跡することにより指先の軌跡を計測することが可能となる。

軌跡計測が可能になる以外に、先の問題点が以下のように解決する。

- (1) フレーム I_t で指先 $F_{i,t}$ についてマッチングに失敗しても、予測位置 $F'_{i,t}$ で代用することが可能である。
- (2) 軌跡の開始時刻から数フレーム間は指先として認識しないことにより、先の問題点 (3) のような雑音への対処が可能となる。
- (3) 雑音への対処によって相関値の閾値を引き下げることが可能となり、指先を見落とす頻度を抑制できる。

本手法で使用する指先位置予測や指先対応付けのアルゴリズムについては、次節以降で詳細に説明する。

4.2 カルマンフィルタによる指先位置の予測

フレーム間で指先同士の対応付けを行う上で、指先予測位置の計算が必要となる。そこで、あるフレームで検出された指先位置を基に、その直後のフレームでの指先位置を予測する手法について説明する。予測にはカルマンフィルタ (Kalman Filter) を使用する [14]。

なお、カルマンフィルタを利用した例として、[8]、[9] では手の 3 次元姿勢推定、関節角推定などを行っている。本研究では、複数の指先の軌跡追跡を安定に行うために利用する。

ここでは画像中における指先の位置と速度を考慮することとし、状態ベクトル \mathbf{x}_t を以下のように定義する。

$$\mathbf{x}_t = (x(t), y(t), v_x(t), v_y(t))^T \quad (1)$$

ここで、 $x(t), y(t), v_x(t), v_y(t)$ は時刻 t での指先位置 $(x(t), y(t))$ 、および指先の速度 $(v_x(t), v_y(t))$ を示す。

次に、 \mathbf{y}_t を時刻 t での指先位置の 2 次元観測量ベクトルとし、基本システムとして次式を定義する。

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{G}\mathbf{w}_t \quad (2)$$

$$\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \quad (3)$$

ここで、 \mathbf{F} は状態遷移行列、 \mathbf{G} は駆動行列、 \mathbf{H} は観測行列であり、それぞれ次の行列を使用する。ただし状

態遷移行列 \mathbf{F} では、隣り合うフレーム間で近似的に等速直線運動を行うことを仮定している。

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

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

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

また、 \mathbf{w}_t はシステム雑音であり、ここでは等速直線運動を仮定した上での速度に加わる雑音を示す。同様に \mathbf{v}_t は観測雑音であり、本システムでは実際の指先位置とマッチングによって得られる指先位置との間の誤差に当たる。それぞれの共分散行列は $\sigma_w^2 \mathbf{I}_{2 \times 2}$ 、 $\sigma_v^2 \mathbf{I}_{2 \times 2}$ 、平均値ベクトルはいずれも 2 次元零ベクトルとする。なお、 $\mathbf{I}_{2 \times 2}$ は 2×2 単位行列を示す。

以上の定義を用いて次式のカルマンフィルタを逐次実行する。

$$\mathbf{K}_t = \tilde{\mathbf{P}}_t \mathbf{H}^T (\mathbf{I}_{2 \times 2} + \mathbf{H} \tilde{\mathbf{P}}_t \mathbf{H}^T)^{-1} \quad (7)$$

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{F} \{ \tilde{\mathbf{x}}_t + \mathbf{K}_t (\mathbf{y}_t - \mathbf{H} \tilde{\mathbf{x}}_t) \} \quad (8)$$

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

ここで、 $\tilde{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1}$ であり、 $\hat{\mathbf{x}}_{t|t-1}$ は $\mathbf{y}_0, \dots, \mathbf{y}_{t-1}$ が与えられたときの \mathbf{x}_t の推定量を示す。また、 $\tilde{\mathbf{P}}_t = \hat{\mathbf{P}}_{t|t-1}$ であり、 $\hat{\mathbf{P}}_{t|t-1} = \hat{\Sigma}_{t|t-1} / \sigma_v^2$ 、 $\hat{\Sigma}_{t|t-1}$ は $\hat{\mathbf{x}}_{t|t-1}$ の推定誤差の共分散行列である。さらに、 \mathbf{K}_t はカルマンゲインであり、また、 $\Lambda = \mathbf{G}\mathbf{G}^T$ である。

こうして得られた $\tilde{\mathbf{x}}_{t+1}$ の指先位置を直後のフレームにおける予測位置として利用する。

また、任意のフレーム数 m フレーム後の指先予測位置を計算する場合は、以下の式を利用する。

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

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

これらの式 (8)、(10) により予測される指先位置をどのように利用するかについては次節にて詳しく説明する。

4.3 予測位置を利用した指先同士の対応付け

前節で得られた予測位置を基にして、フレーム間で指先同士を対応付ける手法について説明する。

4.1 節で述べたように、フレーム間で指先同士を対応付ける操作とは、テンプレートマッチングによって得られる指先位置（以下、指先位置）と予測位置との最適な組み合わせを求める操作である。考えられるすべての組み合わせについて計算を行う場合、最大で120通り（ $= {}_5P_3$ ：指先位置が5個、予測位置が5個のとき）の組み合わせについて考慮しなければならない。そのため、計算量の削減を図るには組み合わせの数を減らす必要がある。そこで、指の交差などの指の順序が入れ替わる動作は起こらないと仮定することにより、考慮すべき組み合わせの数を減らすことにする。以下に具体的な操作を述べる。

まず、手のひらを中心にして時計回りの順に指先位置を並べる処理を行う（図6）。この処理は、手のひらの中心を始点として各指先位置を終点とするベクトルを $f_{i,t}$ 、腕の主軸の向きを表すベクトルを a_t とし、 a_t を基準としたときに $f_{i,t}$ のなす角度 θ ($-180^\circ < \theta < 180^\circ$) が大きい順番に指先を並べることにより実現する。

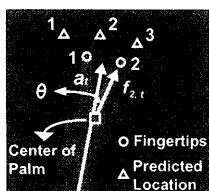
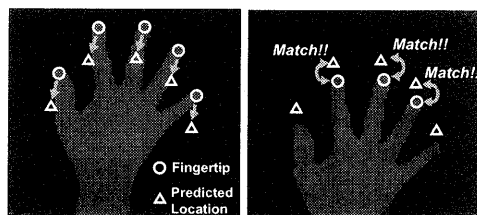


図6: 指の順番と組み合わせ

さらに、予測位置についても同様に並べ、組み合わせを計算するときには、これらの順番を基にして組み合わせが交差しないものだけ計算する。例として図6の場合には、i) $\bigcirc 1-\Delta 1$ & $\bigcirc 2-\Delta 2$, ii) $\bigcirc 1-\Delta 1$ & $\bigcirc 2-\Delta 3$, iii) $\bigcirc 1-\Delta 2$ & $\bigcirc 2-\Delta 3$ という3通りの組み合わせのみ考慮の対象とする。

ここで、指先位置と予測位置との対応付け操作の例として、指先位置3個、指先予測位置5個の場合について説明する図7。

あるフレームで得られた指先位置を図7(a)中の5個の○印とする。このとき、各指に対する次のフレームにおける予測位置（5個の△印）は式(8)によって計算される。次に、次のフレームで得られた指先位置を図7(b)中の3個の○印とする。これら3個の指先位置○と5個の予測位置△について対応する組み合



(a): 予測位置の導出 (b): 指先の対応付け

図7: 画像フレーム間での指先対応付け

わせを考える。この組み合わせを考えるとき、指先の順番を考慮しない場合には組み合わせの数は60通り（ $= {}_5P_3$ ）存在する。一方、本手法では指先の順番を考慮することにより指先位置と予測位置の組み合わせは10通り（ $= {}_5C_3$ ）まで削減される。この10通りの各組み合わせに対してそれぞれ指先位置と予測位置との間の距離の2乗和を計算し、その値が最小になる組み合わせを最適な組み合わせとして対応付けを行う。この処理により、人指し指、中指、薬指の予測位置が得られた指先位置と対応しているという結果が得られる（図7(b)）。

また、以上の処理で対応の取れなかった予測位置については次のように扱う。図7(b)の場合、親指と小指の対応が取れていない。このとき、親指と小指のテンプレートマッチングに失敗したとみなす。そこで、式(10)によって予測位置を計算し、得られた予測位置をそのフレームにおける指先と仮定する。さらに、式(11)によって予測位置の不確定性を計算し、得られた不確定性が閾値を超えた場合には予測位置は信頼できないとみなしてその指についての指先追跡を終了する。指先追跡を継続する場合は、再び式(10)を使用して次のフレームにおける予測位置を計算し、この予測位置と次のフレームで検出される指先位置との対応を考える。ここで対応が取れば通常の指先追跡を継続し、対応が取れなければ前フレームでの処理と同様に指先位置を仮定し、不確定性の判定を行う。

一方、指先位置5個、指先予測位置3個のように指先位置が予測位置を上回る場合には、先の例と同様にして3個の最適な組み合わせを求め、残った指先位置2個を新たに発生した指先候補点としてみなす。ただし、4.1 節でも述べたように、これらの指先候補点はすぐに真の指先としてみなすのではなく、この指先が数フレーム間連続して検出されたときに初めて真の指先としてみなすものとする。

本手法により、任意の個数の指先位置と予測位置を高速に対応付けることが可能となる。

5 実験結果

本手法により指先の軌跡を実時間で計測することが可能となった。さらに4.1節で説明したように、画像フレーム間の対応を考慮しなかった場合と比較して、個々のフレームで指先を検出する性能は向上していることが予想される。本節では個々のフレームでの指先検出性能について比較実験を行い、フレーム間で指先対応を取ることに伴う検出性能の向上を確認する。

実験方法を説明する。まず、被験者は手の甲を上方向に向けた状態を保って机上で手を自由に移動させる。この際、指の本数は一定の本数に保っておく。その場面を真上から赤外線カメラで撮影し、デジタルビデオカメラに録画する。最初に指の本数を1本にして30秒間指を移動させ、以後2本から5本までの場合につき各30秒間指を移動させる。本システムでは1秒間に処理するフレーム数は約30フレームであるため、1回の試行につき約900フレームの実験データを得た。

次に、録画した赤外線映像を入力として、フレーム画像間の対応を取らない手法（以下、手法A）と対応を取る手法（以下、手法B）のシステムを実行する。その際、各フレームで検出された指先の本数を計測し、被験者の意図した指先の本数が正しく検出された割合（正解率）により指先検出性能を比較する。

被験者として男性7名が参加した。また、ハードウェアとして、CPU：Pentium III 500MHzを搭載したPC、画像処理ボードにはHITACHI IP5010、赤外線カメラにはNikonサーマルビジョンLAIRD-S270を使用した。なお、テンプレートマッチングに用いる円形テンプレートは各手法とも同じものを使用し、赤外線画像の2値化処理に用いる閾値も同じ値を使用した。

図8に実験結果を示す。図中の棒グラフは各手法における正解率の平均値を示したものであり、折れ線グラフは各手法における正解率の最低値を示したものである。

グラフより、すべての場合に関して、手法Bの正解率の平均値は手法Aの平均値を上回っていることが読み取れる。特に1本や2本のとき、手法Bの平均値はほぼ100%であり、手法Bは非常に頑健な検出が可能であることがわかる。また、手法Aの最低値に対する手法Bの最低値の向上は平均値の向上よりもさらに大きい。このことから、手法Bではユーザの手の動かし方や画像のノイズに影響されにくい検出がなされていることがわかる。

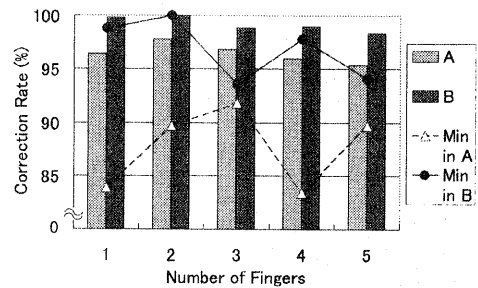


図8: 実験結果 (A: フレーム間対応なし, B: フレーム間対応あり)

以上の結果から、画像フレーム間で指先同士の対応を取った場合、対応を取らない場合よりも各フレームにおける指先検出性能は大幅に向上することが確認された。

6 HCIへの応用

本報告で紹介した指先軌跡計測手法をHCIに応用する方法について紹介する。

机上での作業とコンピュータでの作業の融合に注目した例としてDigital Desk[15]が挙げられる。Digital Deskは机上に投影されたコンピュータ画面を指先などで操作するもので、ユーザは机上に投影されたイラストを切り取ってコピーしたり、机上に投影された電卓で計算をすることができるものであった。

本研究の基礎である小林らのEnhanced Desk[16]では机上の紙書類に注目し、紙書類と電子情報の統合利用を試みている。しかし、手領域の認識に肌色抽出を行っていたため、ユーザの手の上に映像が投影されると手領域抽出がうまくいかないといった問題点があった。また、紙書類と電子情報の対応付けのためにあらかじめ紙書類に付与したバーコードを用いているが、バーコードがある程度以上の大きさで観察されることが必要なため、大型のバーコードを用いなければならなかった。

本研究で試作したEnhanced Desk(図9)は机の上方にプロジェクタを設置し、コンピュータ画面を鏡を介して机上に投影するものである。机上を観察するカメラは赤外線カメラおよびズーム機能付きパンチルトカメラの2台を使用し、机上全体の撮影と一部分を拡大した撮影が可能である。

現在のEnhanced Deskでは、手を直接用いることによるクリック動作やドラッグ操作が実現されている。また、書類等に添付した縮小された2次元バー

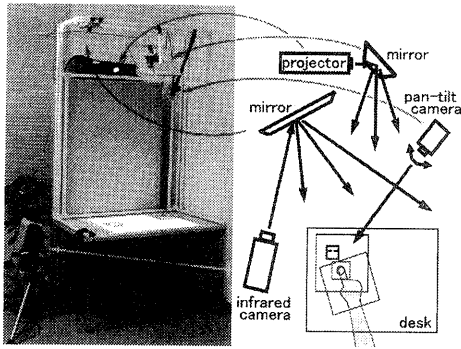


図 9: Enhanced Desk

コードを机上で指差すことで、関連したホームページを机上に投影することなどが可能となっている。

一方で、同じく指を使う作業でも、指先の複雑な動きを必要とする動作を認識することは困難であり、可能なインタラクションは限られているというのが現状である。

そこで、本研究で計測が可能になった指先軌跡にHMMを用いたジェスチャ認識など [17] を適用することにより、より複雑な指先動作を認識することが可能になると考えられ、インタラクションとして可能な例も幅広くなることが期待される。

7 おわりに

本報告では、複数の指先位置とその軌跡を実時間で計測する手法の提案を行った。その結果、本システムによって実時間での安定した指先位置の検出が可能であることが確認された。さらに、各指先の軌跡を実時間で計測することも可能となった。また、この手法をHCIに応用する例について紹介した。

今後の課題として、6章で検討したように、指先軌跡の実時間計測を利用した指先によるジェスチャ認識の研究を進めることが挙げられる。さらに、指先によるジェスチャ認識を利用した有効なアプリケーション開発について検討を加える予定である。

参考文献

[1] T. G. Zimmermann, J. Lanier, C. Blanchard, S. Bryson, Y. Harvill, "A hand gesture interface device," *Proc. ACM Conf. Human Factors in Computing Systems and Graphics Interface*, pp. 189-192, 1987.

[2] R. Cipolla, Y. Okamoto, Y. Kuno, "Robust structure from motion using parallax," *Proc. IEEE International Conference on Computer Vision '93*, pp. 374-382, 1993.

[3] C. Maggioni, "A novel gestural input device for virtual reality," *Proc. 1993 IEEE Annual Virtual Reality International Symposium*, pp. 118-124, 1993.

[4] C. Maggioni, B. Kammerer, "GestureComputer - history, design and applications," *Computer Vision for Human-Machine Interaction (R. Cipolla and A. Pentland, eds.)*, pp. 23-51, Cambridge University Press, 1998.

[5] J. Segan, S. Kumar, "Shadow gestures: 3D hand pose estimation using a single camera," *Proc. 1999 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 479-485, 1999.

[6] M. Fukumoto, Y. Suenaga, K. Mase, "Finger-pointer: pointing interface by image processing," *Computer and Graphics*, Vol. 18, No. 5, pp. 633-642, 1994.

[7] J. Reh, T. Kanade, "Digiteyes: Vision-based hand tracking for human-computer interaction," *Proc. Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 16-22, Austin, Texas, 1994.

[8] N. Shimada, Y. Shirai, Y. Kuno, J. Miura, "Hand gesture estimation and model refinement using monocular camera - ambiguity limitation by inequality constraints," *Proc. 1998 IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 268-273, 1998.

[9] A. Utsumi, J. Ohya, "Multiple-hand-gesture tracking using multiple cameras," *Proc. IEEE Conference on Computer Vision and Pattern Recognition '99*, pp. 473-478, 1999.

[10] T. S. Huang, V. I. Pavlovic, "Hand gesture modeling, analysis, and synthesis," *Proc. 1995 IEEE International Workshop on Automatic Face and Gesture Recognition*, pp. 73-79, 1995.

[11] V. I. Pavlovic, R. Sharma, T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: a review," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 677-695, 1997.

[12] Y. Sato, Y. Kobayashi, H. Koike, "Fast tracking of hands and fingertips in infrared images for augmented desk interface," *Proc. 2000 IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 462-467, 2000.

[13] 小林貴訓, 佐藤洋一, 小池英樹, "Enhanced Desk のための赤外線画像を用いた実時間指先認識インターフェース," 日本ソフトウェア科学会 Workshop on Interactive Systems and Software (WISS'99), pp. 49-54, インタラクティブシステムとソフトウェア VII 近代科学社, 1999.

[14] 中野道雄, 西山 清, パソコンで解くカルマンフィルタ. 丸善株式会社, 1993.

[15] P. Wellner, "Interacting with paper on the DIGITAL DESK," *Communications of The ACM*, Vol. 36, No. 7, pp. 87-96, 1993.

[16] M. Kobayashi, H. Koike, "Enhanced Desk, integrating paper documents and digital documents," *Proc. 1998 Asia Pacific Computer Human Interaction*, pp. 167-174, 1998.

[17] T. Starner, A. Pentland, "Visual recognition of American sign language using Hidden Markov Models," *Proc. 1995 IEEE International Workshop on Automatic Face and Gesture Recognition*, pp. 189-194, 1995.