

## 実時間手領域追跡によるユーザインタフェースの実装

木俣 孝一<sup>†</sup> 若間 俊旭<sup>††</sup> 岡田 至弘<sup>††</sup>

<sup>†</sup> 龍谷大学大学院理工学研究科電子情報学専攻

<sup>††</sup> 龍谷大学工学部情報メディア学科

滋賀県大津市瀬田大江町横谷 1-5

E-mail: {k.kimata, t.wakama}@afc.ryukoku.ac.jp  
okada@rins.ryukoku.ac.jp

あらまし 本研究では、複数対象の手と顔の位置関係を重回帰分析によってモデル化する手法を提案する。これにより、Mean Shift や CAMSHIFT を用いた追跡アルゴリズムの課題である類似した色の対象の重なりや一時的な遮蔽による追跡失敗を回復・修正できることを示す。また、手のジェスチャを用いたインタフェースの例として、手の動きに合わせて表示された画像の操作が可能なバーチャルウォールへ実装について述べる。

## Implementation of User Interface by Real-time Hand Tracking

Kouichi KIMATA<sup>†</sup>, Toshiaki WAKAMA<sup>††</sup>, and Yoshihiro OKADA<sup>††</sup>

<sup>†</sup> Division of Electronics and Infomatics, Graduate School of Science and Technology, Ryukoku University

<sup>††</sup> Department of Media Infomatics, Faculty of Science and Technology, Ryukoku University  
1-5 Yokotani, Seta Oe-cho, Otsu, Shiga, 520-2194 Japan

E-mail: {k.kimata, t.wakama}@afc.ryukoku.ac.jp  
okada@rins.ryukoku.ac.jp

**Abstract** This paper describes the way that the position of the hand and the face is modeled by Multiple Regression Analysis. Mean Shift Algorithm and CAMSHIFT Algorithm can't track the object that occluded by the other objects. But, this way modify tracking point that loses sight of the object. In addition, we implemented the Virtual Wall as a example of interface. It operates a image by user's gestures.

### 1. はじめに

近年、マウスやキーボードなどとは異なる、直感的な操作ができるインタフェースの開発が進められている。特に、頭部方向と手のジェスチャ、視線方向といった非言語行動に対応したインタフェースの開発が注目されている。ユーザのジェスチャを認識する手法には、

- データグローブなどのセンサをユーザに装着させる方法
- 画像処理技術による頭部追跡や手位置判別などにもとづく方法

がある。画像処理技術にもとづく方法は、データグローブなどを用いる方法に比べて、ユーザの負担が少ないという有利さがある一方、

- 安定した追跡の実現
- 実時間での処理の実現

という問題がある。このため、計算量が少なく、急激な動きの変化に対応可能な追跡手法として以下のアルゴリズムにもとづく手法が提案されている。

- Mean Shift Algorithm [1] [2]
- Continuously Adaptive Mean Shift Algorithm (連続適応的平均値アルゴリズム:以下 CAMSHIFT) [3] [4]

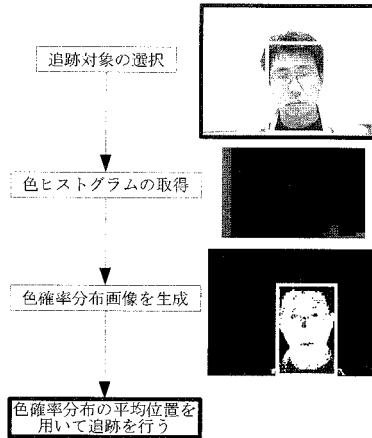


図1 CAMSHIFT の処理の流れ

CAMSHIFT は PUI(Perceptual User Interface) を目的として作られ、Mean Shift Algorithm を動画像に適用した追跡アルゴリズムである。Gary らによって、これらの手法による追跡は計算量が少なく、急激な動きの変化に強いことが報告されている。しかし、どちらのアルゴリズムにもとづく方法も特徴量として色を用いているので、

- 類似した色の対象が重なる場合の追跡に失敗する
- 一時的な遮蔽についても追跡に失敗する
- 重なりや遮蔽のために追跡に失敗した場合、その回復・修正ができない

といった問題がある。さらに、CAMSHIFT では追跡対象の指定をユーザが手動で行わなければならないという問題もある。

そこで本研究では、複数対象の手と顔の位置関係を重回帰分析によってモデル化する手法を提案する。これにより、Mean Shift Algorithm や CAMSHIFT で問題となっていた類似した色の対象の重なりや一時的な遮蔽による追跡失敗を回復・修正できることを示す。

また、手のジェスチャを用いたインタフェースの例として、追跡対象を自動で検出し、手の動きに合わせて表示された画像の操作が可能なバーチャルウォールへ実装について述べる。

## 2. CAMSHIFT による追跡

CAMSHIFT は、Mean Shift Algorithm を動画像から得られる色分布に動的に適用するように拡張した追跡アルゴリズムである。CAMSHIFT での Mean Shift Algorithm は、指定した追跡対象のヒストグラムから作成した色確率分布画像に対して、空間モーメントを用いることで移動先を決定する。ここで、図1に CAMSHIFT の処理の流れを示す。

CAMSHIFT をジェスチャ認識に用いる場合、以下の課題がある。

- (1) 追跡対象の自動検出
- (2) 類似色相を持っている対象 (背景) との重なり/隣接による誤認識への対応
- (3) 急激な照明変化による誤認識への対応
- (4) 誤認識や遮蔽による追跡の失敗の回復・修正

同じあるいは類似する色相の対象 (例えば、顔と手) の色確率分布が近接すると、Mean Shift Algorithm による極大の位置を誤認識する恐れがある。また、急激な照明変化では追跡対象の色分布が大きく変化するため見失ってしまう。さらに、追跡失敗時に再度追跡を再開するための仕組みがないため、一度対象を見失うと同じ対象が追跡できなくなるといった問題がある。そこで、重回帰分析を用いたモデルを適用することで追跡回復を行う。

## 3. 重回帰分析による追跡回復

### 3.1 重回帰分析

重回帰分析とは回帰分析の一つである。回帰分析とは、ある変数 (説明変数) とその変数に対して求めたい結果 (目的変数) が学習データとして与えられたとき、説明変数から目的変数を予測するためのモデルとなる式を推定する統計手法である。回帰分析で、説明変数が1つの場合を単回帰分析、2つ以上の場合を重回帰分析という。重回帰分析は、説明変数  $(x_1, x_2, \dots, x_n)$  と目的変数  $y$  との関係を表す式として以下の式で表される。式 (1) は一般的に線形重回帰モデルと呼ばれる。

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n \quad (1)$$

$(b_0, b_1, \dots, b_n)$  は偏回帰係数という。偏回帰係数は、説明変数と目的変数に関する観測値の組である学習データ  $(y_N, x_{1N}, x_{2N}, \dots, x_{4N})_{N=1,2,3 \dots N}$  を式 (1) に与えてできる  $N$  組の式に最小二乗法を用いることで求められる。そして、偏回帰係数を式 (1) に代入することで求められる式を動きのモデルとして用いる。

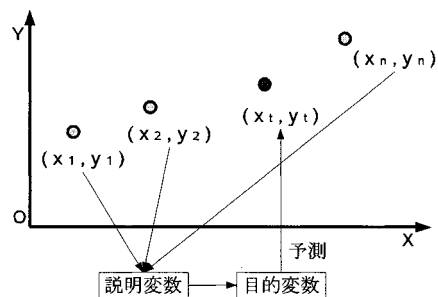


図2 重回帰分析

### 3.2 重回帰分析による動きのモデル生成

実測座標を比較するための予測座標は、図3のように各追跡対象の位置関係を重回帰分析を用いてモデル化したものを2つ使用して $x$ 座標、 $y$ 座標として求めることができる。

まず、位置関係によって追跡対象の動きをモデル化する方法について述べる。ここでの重回帰分析における目的変数は、動きを予測したい追跡対象の重心の座標の1つ( $x$ 座標または $y$ 座標)とし $y_1$ と表わす。そして、説明変数は追跡している対象以外の残った2つの重心座標で( $x_1, x_2, x_3, x_4$ )と表わす。説明変数と目的変数との関係は以下のような式で表わせる。

$$y_1 = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 \quad (2)$$

$$y_2 = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 \quad (3)$$

そして、説明変数を式(2)、(3)に代入し目的変数を得る。以下に具体的な動きモデル生成の処理手順を述べる。

**Step1** 学習データ ( $y_N, x_{1N}, x_{2N}, x_{3N}, x_{4N}$ ) $N=1,2,3 \dots N$ を事前に取得する。

**Step2** step1で得た学習データを式(2)、(3)に代入し、最小二乗法を用いて偏重回帰係数 ( $b_0, b_1, b_2, b_3, b_4$ )を求める。

**Step3** step2で得られた偏重回帰係数を式(2)、(3)に代入し、追跡対象間の位置関係をモデル化した式を作成する。

**Step4** 説明変数をモデル化した式に代入し、目的変数を求める。

step1は事前に行い、step2~4は追跡中に処理を行う。

以上の処理で、説明変数から目的変数を求めることができる。そして、座標を求めるには式(3)についても同様に、step1~step4の処理を行う。これで、2つの動きモデルができ、2つの目的変数からなる予測座標を求めることができる。図3に具体例を示す。図3では、追跡対象は左手の重心座標であるので、予測座標である2つの目的変数は、左手の $x$ 座標と $y$ 座標である。また、目的変数を求めるための説明変数は、顔と右手の重心座標である。そして、それぞれの目的変数に対する動きモデルはstep1~step4の処理手順で生成された、左手の重心の $x_1$ 座標を求める動きモデルと左手の重心の $y_1$ 座標を求める動きモデルである。そして、左の重心座標の予測座標を求めるためには、説明変数である顔と右手の重心座標を2つの動きモデルに代入することで、求めることができる。

ここに示す、図3の処理を、顔や右手の場合も同様に行うことで、それぞれの追跡対象に対してモデルを生成することが出来る。

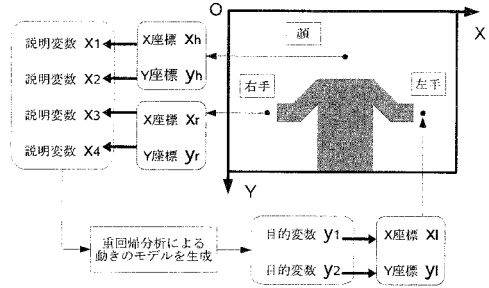


図3 重回帰分析によるモデル

### 3.3 追跡失敗時の回復・修正

得られた予測座標と実測座標を比較して、回復・修正を行う。以下に具体的な処理手順を述べる。

**Step1** 予測座標と実測座標の距離を求める。

**Step2** 予測座標と実測座標の距離が閾値以上であれば、予測座標を探索ウィンドウの中心に置く。

**Step3** 探索ウィンドウサイズを初期位置の探索ウィンドウサイズより少し大きく設定する。

**Step4** step3の探索ウィンドウに Mean Shift Algorithm を適用させ、収束した位置に探索ウィンドウの中心を移動させる。

step1~step4を追跡中に行う。

## 4. 実験

ここでは、追跡対象の1つが類似色の領域と重なることで起こる追跡失敗に対して、重回帰分析を用いた動きのモデルによって、追跡の回復・修正が行われるかを確認し、考察を行う。実験に使用した機器の性能を以下に記す。

[計算機のスペック]

- CPU—Pentium4 3.2G[Hz]
- メモリ—1G[Byte]

[カメラのスペック]

- カメラ解像度—720 × 480[pixel]
- フレームレート—29[frame/s]

実験に用いる動画像として、正面を向いた人物が両手を挙げた状態から下ろす動きを撮影したものを用いた。(図4)追跡対象は右手・左手・顔の3点で、背景にある類似色の領域を通過する際に、類似色の影響を受けて追跡対象が背景に移ってしまう場合を追跡の失敗として想定し、実験を行った。今回は右手が類似色の領域に重なるものとして、あらかじめ重回帰分析を用いて、左手・顔の座標から右手の動きのモデルを求めておき、それを適用した。そして、CAMSHIFTによる対象物体の追跡において、動きのモデルを適用した場合とそうでない場合で対象の追跡にどのような影響が出ているかを考察した。



図4 実験環境

## 5. 結果と考察

まず、CAMSHIFT のみで対象の追跡を行った場合の様子を図 5(a) に示す。背景にある肌色の類似色の領域で、追跡対象が右手から背景に移ってしまっているのが確認できる。次に、動きのモデルを適用した場合の様子を図 5(b) に示す。背景にある肌色の類似色の領域で、追跡対象が背景に移ってしまうことなく、右手の追跡が続けられているのが確認できた。

しかし、求めた動きのモデルを、実際に使用した動画像と同じ動きを2倍に早めた動画像に対して適用すると、追跡対象は背景の類似色の領域に移ってしまうことが分かった。これは、重回帰分析のモデルにおいて、追跡対象の座標の情報を利用して追跡対象の移動速度の影響を受けてしまうことが原因と考えられる。また、今回用いた動きのモデルは線形重回帰モデルによって生成されているため、単純な動きには対応できるが、複雑な動きになると線形重回帰モデルでは表すことが出来なといった問題がある。

これらの解決方法として、説明変数に座標以外のパラメータ(追跡対象の移動方向・移動量など)を用いたり、複雑な動きを単純な動きの組み合わせで表すなどといった方法が考えられる。

## 6. バーチャルウォール

### 6.1 バーチャルウォールの構成

今回実装したバーチャルウォールは、壁面に表示された画像をジェスチャによって操作するものである。バーチャルウォールの構成を図 6、処理の流れを図 7 に示す。背面投射型スクリーンに、プロジェクタで画像を投影し、スクリーン上の手の動きに合わせて、画像を操作する。手の動きはスクリーンの裏側に映るシルエットを、カメラを用いて取得し、追跡を行う。

大まかな処理の流れとして、次のステップに分けられる。

#### Step1 追跡対象の抽出と追跡

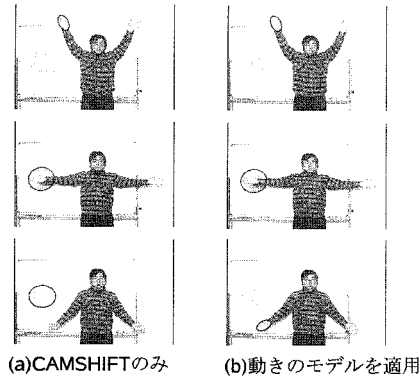


図5 追跡結果

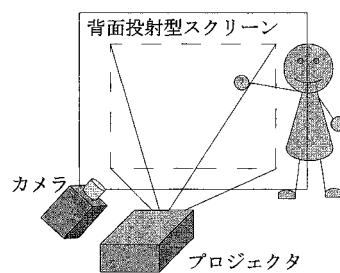


図6 バーチャルウォールの構成

#### Step2 対象の移動方向の計測

#### Step3 ジェスチャの分類

#### Step4 ジェスチャによる画像操作

それぞれのステップについて以下に述べる。

### 6.2 追跡対象の抽出と追跡

ジェスチャにおける追跡対象として、手の領域抽出を行う。バーチャルウォールでは、背面投射型スクリーンに触れている部分のみが裏側に映り込むため、背景差分を用いることでユーザが触れている領域、つまり手領域を含む領域が得られる。次に、背景差分から得られる領域に対して、HSV 表色系を用いて肌色に近い部分を抽出し、肌色の部分の領域が閾値以上の大きさの領域を手の領域とする。最後に、抽出した手の領域の外接矩形を求めることで、自動的に CAMSHIFT における追跡範囲の指定を行い、追跡を始める。

### 6.3 追跡対象の移動方向の計測

ジェスチャの分類を行うために、追跡対象である手の移動方向を求める。CAMSHIFT によって得られた手の重心座標について、現在のフレームと前のフレームを比較して、手が動いている方向と移動距離を求め、ジェスチャの分類のためのパラメータに使う。

1 フレーム前の手の重心座標を  $(x_t, y_t)$ 、現在のフレームの手の重心座標を  $(x_{t+1}, y_{t+1})$  として手の動いた方向

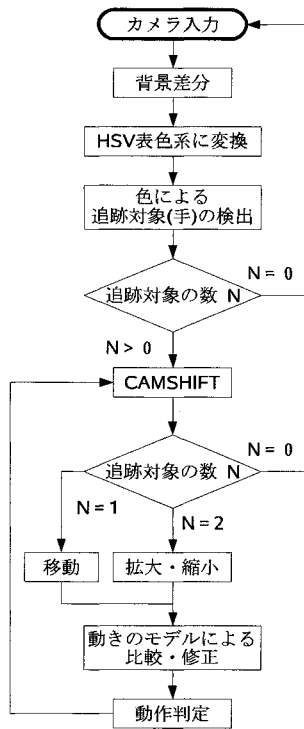


図7 動作判定が行われるまでの流れ

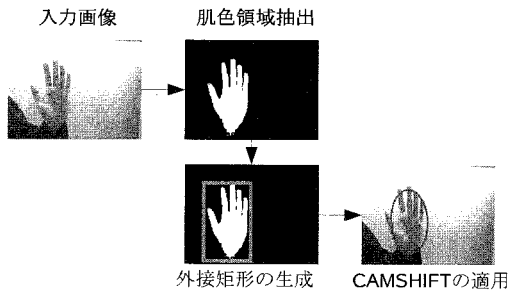


図8 追跡対象の抽出

を  $\theta$ (式(4))、移動距離を  $d$ (式(5))として求める。この2つのパラメータをジェスチャの分類に用いる。

$$\theta = \tan^{-1} \frac{(y_t - y_{t+1})}{(x_t - x_{t+1})} \times \frac{180}{\pi} \quad (4)$$

$$d = \sqrt{(x_t - x_{t+1})^2 + (y_t - y_{t+1})^2} \quad (5)$$

#### 6.4 ジェスチャの分類

画像の閲覧において、一般的にディスプレイ上で行う画像の操作として、画像の移動・拡大・縮小の3つの操作に注目する。これら3つの操作に対して、次に示す両手を用いたジェスチャを対応付ける。ここで、インタフェースに用いるジェスチャとして図10に3つの動きを示す。

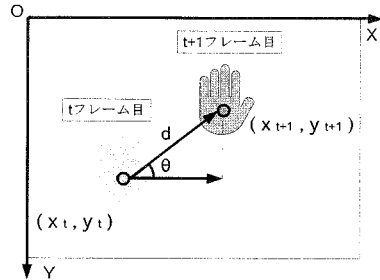


図9 追跡対象のパラメータ取得方法

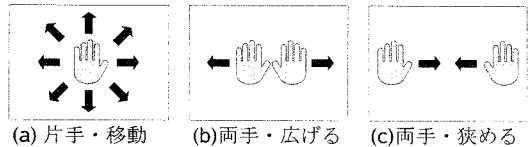


図10 ジェスチャの分類

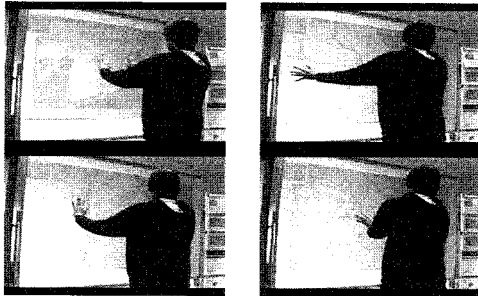
図10を参考に、追跡している手の数によって分類を行う。まず、追跡している手が無いときは、ジェスチャによる入力が行われていないといえる。次に、追跡している手が1つのときは、追跡対象の重心座標を指示点として、指示点が移動している状態とする。最後に、追跡している手が2つのときは、2つの追跡対象の間の距離を用いることで分類を行う。2つの追跡対象の間の距離が増加したとき、両手を広げる動作をしているとする。2つの追跡対象の間の距離が減少したとき、両手を狭める動作をしているとする。

#### 6.5 ジェスチャによる画像操作

前述した、3つのジェスチャと画像操作との対応付けを行う。まず、図10(a)の移動の状態の場合、表示された画像を追跡対象の移動方向へ移動する。次に、図10(b)の両手を広げる状態の場合、表示領域の中心を支点にして、画像を拡大する。最後に、図10(c)の両手を狭める状態の場合、表示領域の中心を支点にして、画像を縮小する。実際にジェスチャによって操作している様子を図11に示す。

### 7. まとめ

今回は、複数対象の手と顔の位置関係を重回帰分析によってモデル化する手法を提案し、従来手法で課題となっていた、類似した色の対象の重なりや一時的な遮蔽による追跡失敗を回復・修正できることを示した。また、手のジェスチャを用いたインタフェースの例として、追跡対象を自動で検出し、手の動きに合わせて表示された画像の操作が可能なバーチャルウォールへ実装について述べた。今後は、より複雑な動きに対応可能な動きのモデ



(a)画像の拡大

(b)画像の縮小

図 11 バーチャルウォールの実装

ルとバーチャルウォールに表示される情報に応じたユーザインタフェースについて検討していく。

#### 文 献

- [1] Dorin Comaniciu, Visvanathan Ramesh, Peter Meer: "Real-Time Tracking of Non-Rigid Object using Mean Shift", IEEE CVPR2000, pp.142-149, 2000.
- [2] 川中 大樹, 中島 平, 岡谷 貴之, 出口 光一郎: "平均値シフト法を用いた複数物体追跡の研究", 情報処理学会研究報告, Vol.2003-CVIM, No.41, pp.9-14, May 2003.
- [3] Gary R. Bradski: "Computer Vision Face Tracking For Use in a Perceptual User Interface," Intel Technology Journal Q2 1998.
- [4] Gary R. Bradski: "Real Time Face and Object Tracking as a Component of a Perceptual User Interface," 4th IEEE Workshop on Applications of Computer Vision (WACV'98), pp.214-218, 1998.
- [5] 石井, 中西, 小池, 岡, 佐藤: "EnhancedMovie: 机型インタフェースを用いた動画編集システム", WISS2003 論文集, pp.41-46, 2003.
- [6] 岡 兼司, "拡張机型インタフェースのための複数指先の追跡とその応用", 情報処理学会論文誌 vol.44, pp.74-84, 2003.
- [7] 曆本 純一, "HoloWall, 壁面型インタフェースの新しい構成手法", コンピュータソフトウェア, ISSN 02896540, pp.535-541, 1998.