

対数極座標空間の高次局所自己相関特徴を用いたポインティングデバイスの提案

小渡 悟

沖縄大学人文学部

概要

ポインティングデバイスは、Graphical User Interfaceを構成する上で不可欠な要素であり、現在は操作が直感的で分かりやすいということでマウスが広く利用されている。しかし、マウスを操作するにはユーザが直接マウスに触れる必要があり、利用する場所が限定される。一般にポインティングデバイスは、ポインティング機構とスイッチ機構からなりたつことから、これらの機構を非接触型のデバイスで実現すれば、利用場所の制限から解放される。本研究では非接触型デバイスである単眼カメラを用い、画像からユーザの手の位置と形状を推定することで、ユーザに対して拘束感や不自然な使用感を与えないポインティングデバイスの構築を検討した。本システムでは、取り込み画像を直交座標系から対数極座標系に変換することで、画像のデータ量を減らし計算コストを削減することで、通常のカメラ以外の専用ハードウェアを用いずにリアルタイム性を実現し、対数極座標空間の高次局所自己相関特徴を用いることで背景の変化や手の回転に対しての頑健性を実現した。また、より快適な計算機とのインタラクションを実現するために、ポインティングによる直接操作型の機能に加え、手の動作軌跡からのジェスチャを識別する機能を組み込んだ。評価実験にて、通常の実験とデジタルビデオカメラを用いて構築したシステムにて、1フレームあたり処理時間は平均約30msと実用的な速度で安定して検出することができ、また手領域の追跡、抽出されたフレーム区間に対してシンボリックサインの推定が実現した。

Pointing Device Using Higher-Order Local Autocorrelation Feature in Log Polar Coordinates System

ODO Satoru

Department of Welfare and Culture, Okinawa University

Abstract

A pointing device is indispensable element in composing graphical user interface. Now, mouse is widely used as a pointing device because its usage is intuitive and easy to understand. For the user to use the mouse device, the user must be able to touch the equipment directly with the provided mouse. Therefore, the area of the operation which the mouse equipment could be freely used is limited. In general, a mouse device is comprised of pointing and switch systems. If it is possible to replace the system on a nontouch device, there would be no need to consider the kind of limitation which a mouse device has. In this paper, we propose a technique to estimate the shape and position of the human hand with from a monocular camera for recognition of the human gestures. In the system, the sequential images are recorded with a monocular camera, and then transformed into the log-polar mapped images by log-polar mapping. The higher order local autocorrelation feature changes when the target is rotated. We introduced therefore a rotational invariance by carrying out the log-polar mapping before extracting the higher order local autocorrelation feature. Our system using the log-polar mapping reduces the data size of the images without spacial accuracy of the target.

1 はじめに

マウス、トラックボール、タブレットなどのポインティングデバイスは、Graphical User Interface（以下、GUI）を構成する上で不可欠な要素であり、その中でも現在は操作が直感的で分かりやすいということでマウスが広く利用されている。しかし、人間から計算機への意思伝達を自然に行なえるマン・マシンインターフェースという観点からみると、マウスをはじめとする一般的なポインティングデバイスは、操作を行なうにはユーザがデバイスに直接触れる必要があり、また、ユーザの手が自然に届く範囲に存在することを前提としているなどの問題点が浮かび上がる。

一方、コンピュータの利用分野は、仮想環境、モバイル環境、ユビキタス環境などと実生活全般に大きく拡大しつつあり、また、計算機の急速な普及に伴いユーザがより良い操作性や自然な操作感覚をえられるインターフェースが求められている。これらの環境で用いられるポインティングデバイスは、ユーザを空間内に固定するものであってはならない。また、モバイル環境においては、計算機の小型化により機器の表面積が減少するため、従来のユーザの指先やスタイラスペンなどが機器に接することによるポインティングデバイスを実装することは困難になってきている。

一般にポインティングデバイスは、ポインティング機構とスイッチ機構からなりたつことから、これらの機構をユーザの身体動作から推定できれば、上記の制限から解放されるものと思われる。ユーザの身体動作からポインティングデバイスを実現する手法としては、腕の筋収縮信号を用いたEMG制御型デバイス [1]、両足で板を傾斜させて操作する両足操作型デバイス [2]、3系統のセンサーを組み合わせた「Ubi-Finger」 [3] などが提案されている。その他にも身振り手ぶりなどのジェスチャ識別までを含めると、データグローブ、磁気式モーションキャプチャ、加速度センサ、関節角度検出センサといったセンサを体に装着することで身体動作を計測することでジェスチャ識別を行なう各種の手法が提案されている [4, 5]。これらはセンサを直接体に装着することと専用の機材を用いることで処理の高速性、安定性、計測精度を実現できるが、特殊な機材を装着することになるためユーザに対して拘束感や不自然な使用感を与えてしまい、快適とは言いがたい。

これに対して非接触型の手法としては、超音波の位相差を用いた位置計測手法 [6]、複数のカメラを用いる手法 [7] などがあるが、特殊な機材を使用することになるため、一般のユーザが利用するのは難しい。通常のカメラを用いて追跡領域の低次のモーメント特徴を用いる手法 [8] も提案されているが、低次のモーメントは対象の大きさや各方向の広がりなどの情報程度しか持たないため、対象が回転すると広がりの方が変わり誤認識を起こす、細かな形状が扱えないなどの欠点を有する。

そのため、一般のユーザが快適に利用することを考慮すると、このときのポインティングデバイスは、非接触でリアルタイム処理が可能で、取り付け箇所や空間を自由に設定できること、また、十分に小型で軽量で安価であることが必要である。

本研究では非接触型デバイスである単眼カメラを用い、画像からユーザの手の位置と形状を推定することで、ユーザに対して拘束感や不自然な使用感を与えないポインティングデバイス

の構築を検討した。

本システムでは、取り込み画像を直交座標系から対数極座標系に変換することで、画像のデータ量を減らし計算コストを削減することで、通常のカメラ以外の専用ハードウェアを用いずにリアルタイム性を実現し、対数極座標空間の高次局所自己相関特徴を用いることで背景の変化や手の回転に対しての頑健性を実現した。また、より快適な計算機とのインタラクションを実現するために、ポインティングによる直接操作型の機能に加え、手の動作軌跡からのジェスチャを識別する機能を組み込んだ。

ここで、計算機の入力デバイスとしての手指動作によるマウス様な機能の実現のためにジェスチャ認識を用いることを考慮すると、任意の動きからジェスチャ認識を行なうよりも、ユーザが入力操作を意図したときにジェスチャを認識したほうが誤動作を招かなくて都合がよい。そこで本システムでは、入力キーとなる手形状をカメラの前で行なうことで、マウス様な機能のオン／オフを行なうことができるようにした。

2 システムの構成

2.1 ジェスチャ認識アルゴリズム

固定単眼カメラで撮影された時系列画像の各フレームの画像を、対数極座標変換 (Log Polar Mapping; 以下, LPM と略す) [9] によって対数極座標画像に変換する。

この LPM による処理の利点は、少ない画素数で高解像度、広域の作業範囲を得るだけでなく、スケーリング不変性、変換中心に対する回転不変性を持っており、また、画像のデータ量を減らし画像処理の計算時間を減少させることも期待できる。しかし、欠点として非均一なサンプリングなため平行移動に対して画像の形が大きく変わってしまうことから、動的視覚処理には不向きであるとされてきた [10]。

この欠点を解決するために、対数極座標上で連続した画像からオプティカルフローを求め、その運動方向が 2π で変化することを利用して平行移動を求める手法 [11] や、直交座標系における 2 次元フーリエ変換と同等の結果を対数極座標系から算出できる指数チャープ変換 [12] を用いることで、通常の 2 次元フーリエ変換と同じように平行移動に対して不変なパラメータを抽出する手法 [13] が提案されている。しかし、前者は通常の動きベクトルの算出が、画像が変形を伴わずに動くことを前提としている。そのため対数極座標上の画像が物体の平行移動により大きく歪むことから、対象物体が大きく移動すると動きベクトルが正しく算出できず、平行移動のパラメータをうまく算出できないという欠点がある。また、後者は通常の変換より処理時間がかかるため、実時間の処理には不向きである。

そこで本システムでは、LPM 画像から時間差分、空間差分、肌色情報を基に輪郭画像を生成し、この輪郭画像の重心をもとに手領域の位置を推定する。また、手領域の過去の位置情報から移動先を推定する。そして、肌色情報を用いて抽出した手領域から高次局所自己相関特徴を算出し、学習ベクトル量子化を用いたニューラルネットワークにて手形状の識別を行なう。この機能を用いてポインティング操作を実現する。さらに、手領域の位置の軌跡を用いジェスチャ

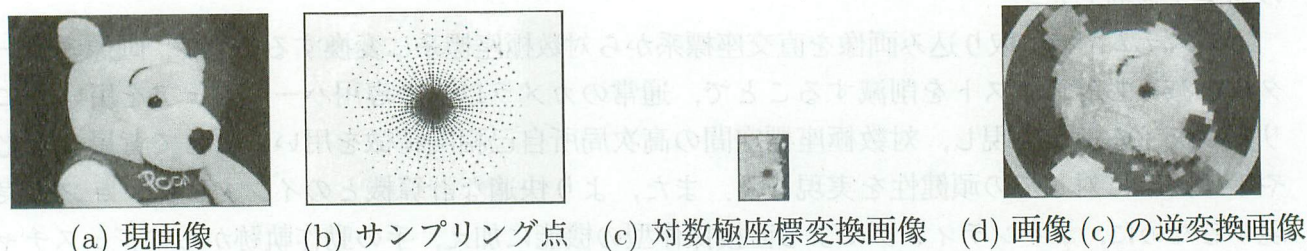


図 1: 対数極座標変換の例

認識を行うことで、手を用いたポインティング操作とジェスチャ操作という二種類の操作を連携して利用することを実現する。ポインティング操作とジェスチャ操作の切り替え(モードセレクト)は、あらかじめ操作の切り替え用の手形状を決めておき、その手形状を提示することで行う。

これより、平行移動量をチャープ変換などの複雑な計算をせずに算出することで処理の高速化が図れ、また、詳細な形状に関する情報が捨象された重心を位置の推定に用いることで、対象物体の急激な変化による追跡精度の悪化の可能性を低減させる。また、肌色領域から抽出した色情報と背景差分を用いることで、背景領域の類似した色情報をもつ物体を除去し、対象物体を抽出することができる。

2.2 対数極座標変換画像の生成

直交画像での座標 $I(x, y)$ を複素平面 Z とし、そのときの複素平面上の点を $z = x + iy$ と表現する。つぎに LPM の座標 $L(p, q)$ も同様に複素平面 W とし、そのときの複素平面上の点を $w = p + iq$ と表現する。このとき、LPM は式 (1) となる。

$$w = \log(z + \alpha) \quad (1)$$

ここで、 α は原点が特異点となるのをさけるための並進量である。

LPM により、原画像 I は角度方向、半径方向の各成分に分解される。半径方向のサンプリングは対数的なものとなるため、周辺部の情報は中心部に比べると大まかに捉えられている。そのため、中心部で高解像度を保ちつつ、周辺部にいくにつれ解像度を下げていくため、空間の状況を大まかに捉えデータ量を大幅に削減し、また処理時間を減少させることができる。LPM の例として、図 1(a) の画像を図 1(b) に示す点においてリサンプリングを行なった結果を図 1(c) に示す。またその逆変換を図 1(d) に示す。

入力画像から LPM を行う際、各画素が LPM 画像のどの画素に対応しているかを計算する必要がある。処理の高速化のためには LPM をハードウェアで実装する方法 [14, 15] もあるが、本システムではユーザに余計な負担を掛けることを避けるために、LPM はソフトウェアの処理で実装した。本システムでは、入力画像の画像サイズは一定であることから変換処理の扱いを容易にするために、あらかじめ前処理の段階で各画素の対応点を求め、ルックアップテーブルを作成する。このようにすることで変換処理の扱いが容易になり、計算時間の短縮をはかる。

2.3 手領域の位置推定

人物の手領域を抽出し追跡を行うために、時間差分、空間差分、肌色情報を用いる。背景と移動領域の分離は、時系列画像中の移動領域は観測された輝度値の時間的变化に対応するため、時間差分を用いることで容易に行える。しかし、時間差分の結果は、背景と移動領域の輝度値の差に依存してしまうため、照明の変化に影響を受けやすい。そのため、画像中の輝度値の変化が大きい部分は、移動領域のエッジ付近にあることを考慮し、照明の変化に影響されがたい空間差分により得られるエッジ情報を併用する。また、ここでは手領域の抽出が目的であるため、その大きな特徴である肌色情報を用いる。

以下、入力画像より手指領域の抽出方法について述べる。入力画像より作成した時刻 t での LPM 画像を $L(p, q, t)$ とし、この $L(p, q, t)$ の時間差分により得られる画像 $L_1(p, q, t)$ 、空間差分により得られる画像を $L_2(p, q, t)$ 、肌色領域 $L_3(p, q, t)$ を次のように計算する。

時間差分画像 $L_1(p, q, t)$ は、式 (2) に示すように連続した 2 フレーム間の差分をとることにより得られる。これより時刻 t の静止領域と運動領域の分割が行われる。

$$L_1(p, q, t) = \begin{cases} 1 & , |L(p, q, t) - L(p, q, t-1)| \geq th_t \\ 0 & , otherwise \end{cases} \quad (2)$$

ただし、 th_t は輝度値が変化したかを決定する閾値である。

空間差分画像 $L_2(p, q)$ は、時刻 t における画像に対して、式 (3) に示す 3×3 の Sobel フィルタを施すことで、画像中のエッジを抽出することで得る。

$$L_2(p, q, t) = \begin{cases} 1 & , \sqrt{L_{HS}(p, q, t)^2 + L_{VS}(p, q, t)^2} \geq th_s \\ 0 & , otherwise \end{cases} \quad (3)$$

ただし、 th_s はエッジであるかを決定する閾値である。また $L_2(p, q, t)$ 、 $L_{HS}(p, q, t)$ 、 $L_{VS}(p, q, t)$ は、それぞれ時刻 t における座標 (p, q) の空間差分値、 p 方向 Sobel オペレータによる値、 q 方向 Sobel オペレータによる値を表している。

入力画像は RGB 表色系で表現されているが、この RGB 表色系では、各値間の相関が高いため、明るさの変化に大きく影響を受けるという問題がある。そのため本稿では、明るさの変動に影響を受けず、RGB それぞれの値が 1 対 1 で対応づけられる $L^*u^*v^*$ 表色系 (CIE1976 表色系) に変換し、その中の u 、 v 平面を用い、肌色領域の平均値 $M(\bar{u}, \bar{v})$ 、分散共分散行列 C を求める。これより、肌色領域 $L_3(p, q, t)$ は、式 (4) に示すようになる。背景領域の壁やダンボールなどの肌色領域が存在する場合にはその領域も抽出されてしまうが、前述の差分画像との論理積をとることで背景領域として静止している壁やダンボールの領域を除去することができる。

$$L_3(p, q, t) = \begin{cases} 1 & , (L_c - M)^T C^{-1} (L_c - M) \geq th_c \\ 0 & , otherwise \end{cases} \quad (4)$$

ただし、 th_c は肌色であるかを決定する閾値である。

このように式 (2) から式 (4) より得られた情報から式 (5) に示す輪郭画像を得る。

$$L_d(p, q, t) = \begin{cases} 1 & , \sum_{i=1}^3 L_i(p, q, t) = 3 \\ 0 & , \text{otherwise} \end{cases} \quad (5)$$

この得られた輪郭画像 L_d から重心を算出することにより、時刻 t における入力画像 I 中の手の位置 $p_{xy}(t)$ が算出される。

2.4 手領域の運動推定

手領域の重心位置を追跡する方法について述べる。時刻 t における手領域の重心位置を $p_{xy}(t)$ 、手領域の移動速度を $v(t)$ 、加速度を $a(t)$ 、時刻 $t-1$ 時による時刻 t での手領域の推定重心位置を $\hat{p}_{xy}(t)$ とする。ただし、時刻 $t=0$ では手領域の重心位置、推定重心位置を取り込み画像の中心点とし、速度、加速度はともに 0 と仮定する。

時刻 t における手領域の推定重心位置 $\hat{p}_{xy}(t)$ は、式 (6) として求まる。この推定重心位置 $\hat{p}_{xy}(t)$ は、直交画像から LPM 画像に変換する場合の対数極座標空間の中心点となる。ただし、式中の Δt はフレーム間隔を表す。

$$\hat{p}_{xy}(t) = p_{xy}(t-1) + v(t-1) \cdot \Delta t \quad (6)$$

ただし、実際の手領域の重心位置 $p_{xy}(t)$ と推定重心位置 $\hat{p}_{xy}(t)$ が一致しない場合は、時刻 $t-1$ から時刻 t までの間に、式 (7) に示す加速度が生じたものとし、移動速度 $v(t)$ は式 (8) とする。

$$a(t-1) = \frac{2}{(\Delta t)^2} (\hat{p}_{xy}(t) - p_{xy}(t)) \quad (7)$$

$$v(t) = v(t-1) + a(t-1) \cdot \Delta t \quad (8)$$

2.5 手形状の推定

2.5.1 手領域の抽出

LPM 画像 L からの手領域の抽出として、式 (4) による肌色情報による領域選択後、肌色領域のラベリングを行い、その中の最大領域を手領域とする。しかし、図 2 に示すように、図 2(a) から肌色領域を抽出する際に、照明と手の位置関係により、図 2(b) のように影が生じ手領域の一部が欠けてしまう場合がある。そのため本システムでは、手指全体の形状が含まれるよう肌色の最大領域を抽出後、LPM 画像の半径方向を外周向けに走査し、図 2(c) に示すように肌色領域がすべて含まれる範囲までを切り出すことにした。これより、若干の背景のノイズが含まれるが、手領域の取りこぼしを回避することができる。図 2(c) の画像にエッジ強調を施し、高次局所自己相関特徴の抽出を行う。

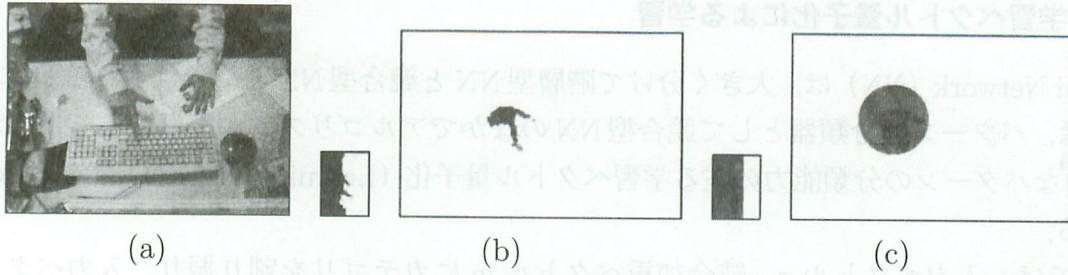


図 2: 手領域の抽出

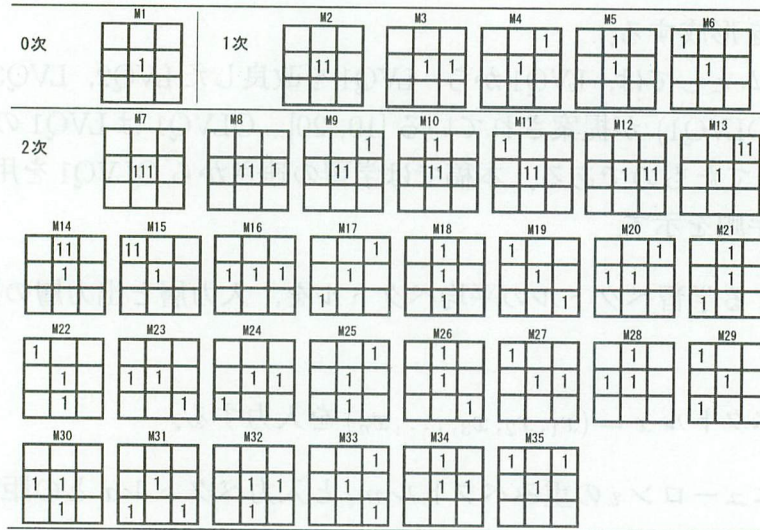


図 3: 高次局所自己相関特徴のための局所パターン

2.5.2 高次局所自己相関特徴の算出

高次局所自己相関特徴は、大津らにより提案された画像の認識や計測のために有効な画像特徴 [16] であり、式 (9) で定義される高次自己相関関数 [17] について、参照点とその近傍画素周りの計算をしたものである。

ここで $f(r)$ は画素位置 r の輝度値、 N は次数、 (a_1, a_2, \dots, a_N) は変位方向を表している。

$$x^N(a_1, a_2, \dots, a_N) = \int f(r)f(r+a_1)\dots f(r+a_N)dr \quad (9)$$

一般に自然画像を処理対象とする場合、近接する画素同士の相関が重要であると考えられるため、変位方向を参照点 r の周りの局所的な 3×3 画素の領域に限定し、高次自己相関係数の次数 N を 2 次までとした。これより平行移動に等価なものを除くと、特徴の数は図 3 に示す 35 個になる。図中の「1」は、局所パターンの対応する画素の位置にあたる。各特徴の計算は、局所パターンの対応する画素の値の積を全画素に対して足し合わせることで得られる。

高次局所自己相関特徴は平行移動パターンに対して不変であるという大きな利点があることから、LPM 画像から抽出することで、回転やスケーリングに対して不変な特徴となる。

2.5.3 学習ベクトル量子化による学習

Neural Network (NN) は、大きく分けて階層型 NN と競合型 NN とに分けられるが、本手法においては、パターンの分類器として競合型 NN のなかでアルゴリズムが簡単であるにもかかわらず、強力なパターンの分類能力の在る学習ベクトル量子化 (Learning Vector Quantization; LVQ) を用いる。

LVQ では、入力ベクトル x 、結合加重ベクトル m にカテゴリを割り振り、入力ベクトル x と結合加重ベクトル m とのカテゴリを比較し、一致していれば入力と結合重みベクトルの距離を近づけ、一致しなければ遠ざける操作を行う。この操作により、理論的に最適な識別境界を与えるベイズ識別境界を形成する。

LVQ のアルゴリズムとしては、LVQ1 から、LVQ1 を改良した LVQ2, LVQ3, The optimized-learning-rate LVQ1 (OLVQ1) が提案されている [19, 20]。OLVQ1 は LVQ1 のそれぞれの m_i に学習率 $\alpha_i(t)$ を割り当てたものである。本稿では学習の速さから OLVQ1 を用いる。

以下に OLVQ1 の手順を示す。

1. カテゴリに属する学習ベクトルの平均ベクトルを、入力層と出力層の間の重みの初期値として与える。
2. 入力層に入力ベクトル $x = (x_1, x_2, x_3, \dots, x_n)$ を入力する。
3. 出力層では各ニューロン i の重みベクトル m_i と入力ベクトル x との距離を式 (10) に従って計算する。

$$c = \arg \min_i \{ \|x - m_i\| \} \quad (10)$$

4. x と最も距離が近い m_i を勝者ベクトル m_c とする。
5. 式 (11) を用いて重みベクトルを更新する。

$$\begin{aligned} m_c(t+1) &= m_c(t) + \alpha_c(t)[x(t) - m_c(t)] \\ &\quad \text{if } x \text{ is classified correctly,} \\ m_c(t+1) &= m_c(t) - \alpha_c(t)[x(t) - m_c(t)] \\ &\quad \text{if the classification of } x \text{ is incorrect,} \\ m_i(t+1) &= m_i(t) \text{ for } i \neq c. \end{aligned} \quad (11)$$

ただし、 t は時間、 α_c は学習係数で式 (12) に従う。

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \quad (12)$$

ただし、 $s(t)$ は分類が正しければ +1, 正しくなければ -1 である。

6. 学習回数が一定回数以上であれば終了する。そうでなければ 2. に戻る。

OLVQ1 の学習は、勝者ベクトルが正しいクラスに属する場合には、結合重みを学習ベクトルに近づけるように、正しくないクラスに属する場合には、学習ベクトルから遠ざけるように、重みの更新を行う。

2.6 ジェスチャ推定

手指によるジェスチャ認識の場合、一連の手指動作をジェスチャ動作区間とその他の動作区間に分割しなければならない。この場合、連続 DP を用いてジェスチャ区間を指定せずにジェスチャのスポッティング認識を行う手法 [18]，手指の動作が極小となる時点を分割位置とする手法 [21] などがある。

計算機の入力デバイスとしてのインターフェースとしてジェスチャ認識を用いることを考慮すると、任意の動作から対象とするジェスチャを認識するよりは、ユーザが入力を意図したときにジェスチャを認識をしたほうが誤動作などを招かなくて都合がよい。

そこで本手法では、ユーザが手の動きが極小となり、手形状がシステムに登録されているジェスチャ入力キーと一致したときを動作の開始、または、終了の点として扱い、その間の区間をジェスチャ区間とみなし単純 DP による照合を行う。一般に画像からの動作区間の推定は、手指の動作が極小となる時点を正確に検出することは難しいが、本手法では動作区間の推定情報として手指動作が極小であるという情報に加え、そのときの手指形状がシステムに登録されているかという情報を加えることで、手指動作の動作区間の決定を実現している。

ジェスチャ推定のための特徴ベクトル S は、ジェスチャ区間の手指の追跡軌跡から得られる軌道ベクトルを用いる。時刻 t における手の位置を $p_{xy}(t) = (x_t, y_t)$ とすると、時刻 t の特徴ベクトル $s(t)$ は式 (13) となる。

$$\begin{aligned} s(t) &= (v(t), \theta(t)) \\ v(t) &= (x_t - x_{t-1}, y_t - y_{t-1}) \\ \theta(t) &= \cos^{-1} \frac{v(t) \cdot v(t-1)}{|v(t)||v(t-1)|} \end{aligned} \tag{13}$$

これより、ジェスチャの推定は、特徴ベクトル $\{S : s_1(v, \theta), s_2(v, \theta), \dots, s_i(v, \theta), \dots, s_N(v, \theta)\}$ と辞書内の参照特徴ベクトル $\{T : t_1(v, \theta), t_2(v, \theta), \dots, t_j(v, \theta), \dots, t_M(v, \theta)\}$ とのコストを DP マッチングにより算出し、その中から最小のものを選択することにより行う。

3 評価実験

マウス様の機能を実現するには、ポインティング、マウスの右ボタン、左ボタンを判別しなければならない。また、ジェスチャの入力キーとなる手形状を加えると最低でも 4 クラスを識別しなければならない。本システムでは、手形状として図 4 の 4 種類を設定した。図 4 の (a) はポインティング、(b) はマウスの右ボタン、(c) は左ボタン、(d) はジェスチャの入力キーに対応する。

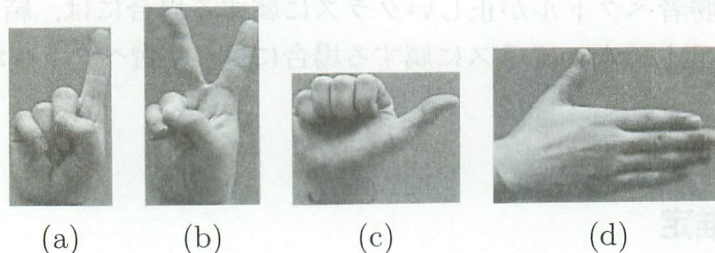


図 4: 認識対象となる手形状



図 5: 手指画像サイズ 100%の実験画像の例



図 6: 複雑背景下での手指画像サイズ 100%の実験画像の例

本システムの有効性を示すために評価実験を行った。実験条件を以下に示す。原画像の解像度が 360×240 pixels, それに対する LPM 画像の解像度が 120×120 pixels, 60×60 pixels, 40×40 pixels, 30×30 pixels の場合に対して検討を行った。

ユーザが手を提示する位置を中心に手以外が入らない程度の領域を Panasonic 製 Digital Video Camera (以下、DVC と略す) で撮影した。図 4 の 4 種類の手形状を対象とし、4 人のユーザにそれぞれの動作を行なわせ、その際に、手首を適度に左右に傾けさせることで、図 4 から傾きのある画像を加えた。撮影した画像は IEEE1394 経由で 360×240 の解像度で PC に取り込んだ。撮影した画像から、一人当たり一つの手形状パターンが 200 枚になるように切り出すことで、総計 3200 枚 (4 人 \times 4 種類 (手形状) \times 200 枚) の画像をえた。

実験画像は、各画像から手領域を外接矩形枠で切り出し、これを縮小することで 5 種類の大きさの手画像を作成し、解像度 360×240 の単色の背景画像の中心と手画像の重心が一致するようにそれぞれ合成することで、総計 16000 枚 (5 種類 (大きさ) \times 3200 枚) の画像を得た。この場合の手画像の大きさは、ユーザの上半身がカメラに収まるときの手の大きさを 100% とし、この大きさの 50%, 75%, 100%, 125%, 150% となるようにした。図 5 に対象画像の例を示す。

OLVQ1 の学習には各カテゴリの割合が等しくなるように総サンプルを 2 つの部分集合に分割し、片方を学習用に、もう片方を評価用に用いる分割学習法を行い、また、識別率はそれらを入れ替えて二つの平均をとる相互較正を行なった値とした。

OLVQ1 にて学習したニューラルネットを用いて、図 6 に示す複雑な背景と合成した画像を用いて識別率を評価した。

本システムでの結果を表 1 に示す。表中の手の大きさの 1 から 5 は、それぞれ手の大きさの

表 1: 対数極座標によるリサンプリングと手の大きさとの識別率の比較

手の大きさ 解像度	1	2	3	4	5	平均値	標準偏差
120 × 120	84.1%	91.5%	92.3%	93.6%	90.8%	90.5%	3.7
60 × 60	87.3%	95.5%	95.4%	94.1%	90.3%	92.5%	3.6
40 × 40	88.4%	95.2%	94.9%	93.8%	92.4%	92.9%	2.8
30 × 30	88.6%	95.4%	95.1%	95.3%	91.6%	93.2%	3.0

の 50%, 75%, 100%, 125%, 150%に対応し, 解像度は LPM 画像の解像度に対応する。

ここで, 本システムで用いる LPM 画像の解像度だが, 実験結果からは 30 × 30 のときに最も識別率が高いが, 30 × 30 の解像度では手の追跡誤差が大きく追跡に失敗することがあった。手の追跡の失敗はポインティング操作に大きな支障ももたらすことから, 追跡精度との兼ね合いから LPM 画像の解像度は 60 × 60 とした。解像度 60 × 60 時の追跡精度は, 直径 15pixel の球体が移動する動画像において, 平均誤差は 6.2pixel, 標準偏差 2.9, 最大誤差 11.2pixel であった [22]。許容できる誤差の最大値を物体の大きさと同程度と仮定すると, 本システムの追従能力は, ユーザが通常のポインティング操作を十分に行なうことができるといえる。

実際に本手法を用いたマウス機能とランチャー機能を組み込んだアプリケーションを作成して, 評価実験を行なった。DVC は図 7 に示すように, ユーザの手元が画面内に入るように上方から撮影するように設置した。このときの DVC と手元の距離は 120cm, 画面範囲は垂直方向で 50cm, 水平方向で 70cm となっている。

Intel Pentium III 500MHz を搭載した PC を用い, Panasonic 製 DVC からの映像を IEEE 1394 インターフェース経由で取り込んだ。通常の照明環境下の室内で撮影し, 画像サイズ 360 × 240 画素, RGB 各 256 階調である。また, 実験前に「ポインティング」「右クリック」「左クリック」「モード切り替え」の 4 つの基本操作と手形状の対応付け, および, ジェスチャ操作モード時のジェスチャの登録をおこなった。

手形状の識別結果を画面上のマウスカーソルの形状で表すことで, 使用するユーザに識別結果を知らせることで, 誤識別を行った場合でも容易に修正することが出来た。評価実験において, 1 フレームあたり処理時間は平均約 30ms と実用的な速度で安定して検出することができた。

4 まとめ

本研究ではユーザに対して拘束感や不自然な使用感を与えない非接触型センサである単眼カメラを用い, 入力された画像からジェスチャを推定する手法を提案した。時系列画像を直交座標系から対数極座標系に変換を行ない, 時間差分, 空間差分, 色情報を用いて手領域を抽出し, 手形状の識別は, 対数極座標空間の高次局所自己相関特徴を OLVQ1 にて学習させたニューラルネットにより行なった。また, より快適な計算機とのインタラクションを実現するために, ポ

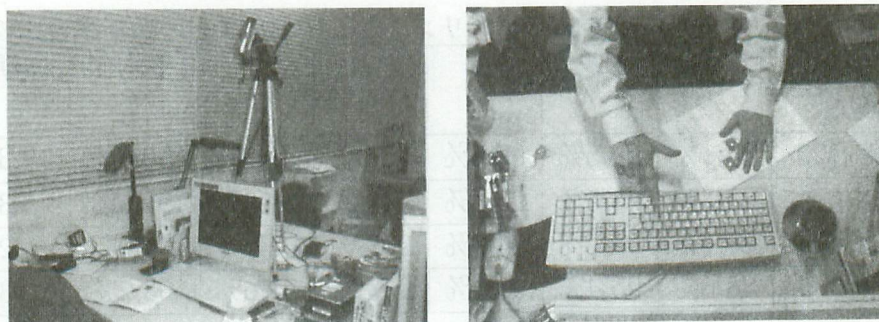


図 7: 実験環境

インテュイティブによる直接操作型の機能に加え、手の動作軌跡からのシンボリックサインを識別する機能を組み込んだ。評価実験にて、通常の計算機とデジタルビデオカメラにて構築したシステムを用いて、1フレームあたり処理時間は平均約 30ms と実用的な速度で安定して検出することができ、また手領域の追跡、抽出されたフレーム区間に対してシンボリックサインの推定が実現した。

参考文献

- [1] 辻敏夫, 福田修, 村上満, 金子真, “ニューラルネットを利用した EMG 制御型ポインティングデバイス,” 計測自動制御学会論文集, vol.37, no.5, pp.425-431, 2001.
- [2] 久米祐一郎, 井上啓, “両足操作型ポインティングデバイスの検討,” 映像情報メディア学会誌, vol.54, no.6, pp.871-874, 2000.
- [3] 塚田浩二, 安村通晃, “Ubi-Finger: モバイル指向ジェスチャ入力デバイスの研究,” 情報処理学会論文誌, vol.43, no.12, pp.3675-3684, 2002.
- [4] 宮尾淳一, “手話学習システムのための手話単語特徴に基づく教授法”, 信学論 D-I, vol.J83-D-I, no.10, pp.1120-1128, 2000.
- [5] 澤田秀之, 橋本周司, “加速度センサを用いたジェスチャ認識と音楽制御への応用,” 信学論 A, vol.79-A, no.2, pp.452-459, 1996.
- [6] 野中秀俊, 伊達惇, “超音波の位相差を利用したポインティング装置 (SPD) の開発,” 計測自動制御学会論文集, vol.29, no.7, pp.735-744, 1993.
- [7] 渡辺博己, 本郷仁志, 安本護, 山本和彦, “マルチカメラを用いた全方位ポインティングジェスチャの方向推定,” 電学論 C, vol.121, no.9, pp.1388-1394, 2001.
- [8] 高松亮, 佐藤誠, “最適視点視野による掌の追跡と手指形状認識に基づくポインティングデバイスの提案,” ヒューマンインターフェース学会, vol.1, no.1, pp.45-52, 1999.

- [9] E.L. Schwartz, "Computational anatomy and functional architecture of striate cortex: a spatial mapping approach to perceptual coding," *Vision Research*, vol.20, no.8, pp.645–668, 1980.
- [10] Richard Wallace, Ping-Wen Ong, Ben Bederson, and Eric L. Schwartz, "Space Variant Image Processing," *International Journal of Computer Vision*, vol.13, no.1, pp.71–90, 1994.
- [11] Noboru Okajima, Hiroki Nitta and Wataru Mitsuhashi "Motion Estimation and Target Tracking in The Log-Polar Geometry," *Technical Digest of the 17th Sensor Symposium*, pp.381–384, 2000.
- [12] Giorgio Bonmassar, and Eric L. Schwartz, "Space-Variant Fourier Analysis: The Exponential Chirp Transform," *IEEE Pattern Analysis and Machine Vision*, vol.19, no.10, pp.1080–1089, 1997.
- [13] 野村智宣, 三橋渉, "中心窩センサーの設計と物体認識への応用," *信学技報 EID2000–313*, pp.77–82, 2001.
- [14] 末松良一, 山田宏尚, "中心窩を有する新しい視覚センサの開発——広角高歪曲レンズの開発," *計測自動制御学会論文集*, pp.1556–1563, vol.31, no.10, 1995.
- [15] 清水創太, 末松良一, 矢羽田茂人, "広角高歪曲レンズ中心窩視覚センサシステムの開発," *日本機械学会論文集 C 編*, pp.4257–4263, vol.63, no.616, 1997.
- [16] N. Otsu and T. Kurita, "A new scheme for practical, flexible and intelligent vision systems," *Proc.IAPR Workshop on Computer Vision*, pp.431–435, 1988.
- [17] J.A. McLaughlin and J. Raviv, "Nth-order autocorrelations in pattern recognition," *Information and Control*, vol.12, pp.121–142, 1968.
- [18] 西村拓一, 向井理朗, 野崎俊輔, 岡隆一, "動作者適応のためのオンライン教示可能なジェスチャ動画像のスポットティング認識システム," *電子情報通信学会論文誌 D-II*, vol.J81–D-II, no.8, pp.1822–1830, 1998.
- [19] Teuvo Kohonen, "Self-Organizing Maps," *Springer Series in Information Sciences*, vol.30, 1995.
- [20] Teuvo Kohonen, "Self-Organization and Associative Memory," *Springer Series in Information Sciences*, vol.8, 1984.
- [21] 佐川浩彦, 竹内勝, "手話認識における手動作セグメンテーション方式," *ヒューマンインターフースシンポジウム '99*, pp.749–754, 1999.

- [22] Satoru Odo, Kiyoshi Hoshino, Hand shape recognition using higher order local autocorrelation features in log polar coordinate space 286–292, *Journal of Robotics and Mechatronics*, vol.15, no.3, 2003.