

モーションセンサを用いた 直感的ポインティングシステムの提案と実装

山本 英礼 松原 俊一 Martin J. Dürst

青山学院大学理工学部情報テクノロジー学科

1 はじめに

近年、人がコンピュータを扱う機会が増加し、様々な状況においてポインティングシステムが必要とされている。従来はマウスによるポインティングが主流であったが、近年タッチパネルなどの直感的なシステムが普及してきている。さらにセンサ技術の発展に伴い、デバイスに触れることなくコンピュータを操作するタッチレスシステムも開発されている。

本研究では、3次元モーションセンサである Leap Motion¹ を用いてタッチレスで直感的なポインティングシステムを提案する。

2 Leap Motion

2.1 Leap Motion Controller

Leap Motion Controller はアメリカ合衆国の Leap Motion 社によって 2012 年に開発された 3 次元モーションセンサである。2 台の赤外線カメラを用い、3 次元画像を撮影、解析を行うことで空中での手指の位置や動きが認識可能である。

2.2 Leap Motion SDK

各種言語で Leap Motion を利用するためのライブラリやサンプルコードなどが Leap Motion 社から提供されており、本研究では C# を用いて実装する。Leap Motion SDK では、認識している手の数や指の本数、各指の関節や指先の位置、指や掌が向いている方向、またその移動速度などのデータが取得可能である。座標はデバイスの中心を原点とした 3 次元座標系であり、この座標系を本論文内において Leap 座標系と呼ぶ。

3 提案手法

本研究では、Windows 上でのポインティング操作を空中の手の動きで代用することを目的とする。従来マウスやキーボードで行っていた機能をタッチレスな形で実現する。カーソル移動を人差し指の指差し動作により、マウスのクリックやボタンにあたる操作をそれ以外

の指によるジェスチャで実現する。また、カーソル移動を行っていない方の手のジェスチャにより Windows 用ショートカットキーの操作、マウスホイール操作、スクリーンキーボードの表示機能を利用可能にする。さらにキャリブレーション機能を実装し、モニターと Leap Motion の位置関係を取得することで Leap Motion の設置場所に自由度を持たせる。

4 実装

4.1 カーソル位置の計算

片手の人差し指で指差した位置にカーソルを移動する機能を実装する。初めに認識された手をカーソル操作の手とし、その人差し指の指先の座標と向いている方向のベクトルを取得する。この二つの情報とモニターの位置情報から画面上の指差している位置を推定する。

$$I.x = (M.z - P.z) / D.z * D.x + P.x$$

$$I.y = (M.z - P.z) / D.z * D.y + P.y$$

M は Leap 座標系でのモニターの中心の座標、 P は指先の座標、 D は指さしている方向のベクトルとする。これにより Leap 座標系においてモニター上の指差している座標 I が推定される。さらに座標変換を行い、モニターのピクセル座標に変換する。

$$X : (I.x - M.x) * dpm + WindowWidth / 2$$

$$Y : (M.y - I.y) * dpm + WindowHeight / 2$$

dpm はモニターの解像度 (dot/mm)、 $WindowWidth$ 及び $WindowHeight$ はピクセルでのモニターの幅と高さである。これによりカーソルの移動先が推定される。

指先の震えなどによるカーソルの振動を抑えるため、指先座標と方向のベクトルは平均値を使用している。位置推定の計算には 0.3 秒分のデータの移動平均を用いている。平均値を利用することによる大きなカーソル移動の遅延を防止するため、指の速度が閾値を超えると古いデータを強制削除する設定も行った。これにより、カーソルが瞬間的に移動することがあるものの、高速の移動にも対応した。

4.2 キャリブレーション機能

キャリブレーションを開始すると、画面が暗くなり、画面中央に円形のマークと時間表示用の枠が表示され

Hideyuki Yamamoto, Shunichi Matsubara and Martin J. Dürst
Department of Integrated Information Technology, College of
Science and Engineering, Aoyama Gakuin University
5-10-1 Fuchinobe, Chuo-ku, Sagami-hara, Kanagawa 252-5258,
Japan
duerst@sw.it.aoyama.ac.jp

¹<https://www.leapmotion.com/>

る．そのマークを人差し指でタッチしたまま手を静止させるとカウントダウンが始まり，3秒後にキャリブレーションが完了する．これによって物理的に Leap 座標系における画面中心の位置を取得する．

4.3 片手で行うジェスチャ

カーソル移動を行う手では，人差し指以外の指を使用するジェスチャを実装した．ジェスチャ操作を行っていない状態でのカーソル移動を行う手は指を広げた状態とし，各種操作は指を曲げることで実現される．これらと後述するスクリーンキーボードを用いることでマウス及びキーボードの基本操作をタッチレスに行うことが可能になる．

4.3.1 マウス左ボタンの操作

マウスの左ボタンの操作には親指の屈伸を利用する．親指の屈伸を判定し，曲がっている場合左ボタン操作のフラグが立つ．そして1秒以内に親指が伸びるとマウス左ボタンを一度クリックする．1秒間曲げ続けると左ボタンを押下し，ドラッグ操作が可能になる．

4.3.2 マウスの右ボタンの操作

マウス右ボタンの判定には小指の速度と人差し指の速度を用いる．小指のみをすばやく曲げる動作をすることで発生する人差し指との速度の差が閾値を超えると一度クリックする．2本の指の速度の差を利用することで，手全体を移動させた際に誤作動を防ぐ．

4.3.3 マウスホイールボタンの操作

マウスホイールボタンの操作には中指の屈伸を利用する．4.3.1節と同様に SDK の機能を用いて指が曲がっていると判定されている間，マウスホイールボタンを押下する．

4.4 両手で行うジェスチャ

カーソル移動を行う手と別の手を使用することでマウスホイールの回転と Windows のショートカットキー操作を実装した．

4.4.1 マウスホイールの回転

マウスホイールの回転はスクロールやズームで多く使用されるため，直感的に操作できるようスワイプによるものとする．判定には人差し指と中指の速度を用いる．2本の指を同時に上または下方向にすばやく移動させ，両指の速度が閾値を超えるとマウスホイールの1目盛分回転させる．

4.4.2 「戻る」、「進む」のショートカット操作

Windows では Alt + キーで「戻る」，Alt + キーで「進む」のショートカットがデフォルトで搭載されているため，ブラウザの閲覧などで頻繁に使用されるこの機能を実装する．こちらは左右にすばやく2本の指を移動させるスワイプにより判定を行う．

4.4.3 スクリーンキーボードの表示

Windows 7では，タッチパネルでキーボードを利用するためのスクリーンキーボードが実装されている．掌を上に向けるジェスチャによりスクリーンキーボードの表示，非表示を行う．判定には掌の法線ベクトルを用いる．法線ベクトルの y の値が 0.9 を超えると掌が上に向いたと判定し，スクリーンキーボードの表示，非表示を切り替える．

5 評価実験

本研究では，図1に示す実験用 Windows フォームアプリケーションを作成し，既存のシステムとの比較実験を行った．比較対象はマウス，トラックボール，タッチパネルとする．上部の実験開始ボタンを押すと，そのボタン上に1～9の数字がランダムに1度ずつ表示される．下部に並んだボタンから同じ数字のボタンをクリックすると上部の数字が変更されるため再度同じボタンをクリックする．この操作を全ての数字がクリックされるまで繰り返し，カーソルが正しいパネル上に移動するのにかかる時間と，クリック操作が完了するまでの時間を逐次計測する．この実験を各システムにつき5回繰り返す．



図 1: 実験用アプリケーション

6 実験結果

被験者は筆者らと同じ研究室に所属する 20 台男性 6 人であった．被験者は全員，普段トラックボールを使用していないく，トラックボールに対する習熟度は全く無いといえた．各デバイスについて実験を行い，正しいパネルにカーソルを移動するまでの時間と，その後クリック操作が完了するまでの時間の平均値を計測した．その結果，提案手法を用いた場合のカーソル移動の平均時間は 0.99 秒，クリックにかかる時間は 0.88 秒となった．タッチパネルでは 0.86 秒と 0.02 秒，トラックボールでは 0.97 秒と 0.55 秒，マウスでは 0.65 秒と 0.36 秒であった．

7 むすび

本研究では，3次元モーションセンサである Leap Motion を用いてタッチレスで直感的なポインティングシステムを提案，実装した．