

2ZA-9

# KINECT を用いた平面上の 3D モーション認識アルゴリズムの開発

松本哲平<sup>†</sup> 築地立家<sup>‡</sup>  
東京電機大学 理工学部

## 1. はじめに

Kinect for Windows(以下 kinect)は Microsoft 社の発売している Xbox360専用のコントローラーであった Kinect を、Windows 用として新たに発売したものである。また開発用ツールである Kinect for Windows SDK もリリースされた。これにより、ジェスチャーや手の座標を捉えてマウス操作と同様なカーソル移動<sup>[1][2]</sup>やクリックを行うなど、様々な開発がされた。その際、ディスプレイの特定の位置をポインティングする際の精度やチャタリングが問題となる。例えば、中村 卓氏の研究<sup>[2]</sup>ではウェイティングと呼ばれる、ポインタの静止時間を設定することでチャタリングを防止している。しかしながらいずれの研究においても、ポインティングの精度は扱っていない。さらに、指認識の空間が kinect との相対位置で決定されるような3次元空間であるために、相対位置が変化するたびにパラメータ調整が必要となる。

本研究では平面板の四隅に基準となるマーカーを配置して、マーカーからの相対位置を割り出すことにより、kinect 指認識によるマウス操作を実現した (図1)。その結果、ポインティングの精度を高めること、および kinect から平面板までの相対位置が変化しても、平面板上の2次元空間からディスプレイ上の2次元空間へのマッピングは不変となることを、実証実験によって確かめた。

なお、Kinect はカメラと赤外線センサによって得られるデータを処理することによって指認識を行う。一方、指認識により特化したデバイスとして、CMOS イメージセンサを利用した LEAP が知られており、スワイプ等の PC 画面操作を実現することは可能である。しかしながら、LEAP を利用した既知の開発事例においても、ポインティングの精度は扱っていない。指認識の空間が LEAP からの相対位置で決まる 3次元空間であるために、指先とカーソルの位置を合わせることがむづかしく、精度の高いポインティングには難点がある。



図1. アクリル板を使用

Development of Kinect-based algorithms for finger tracking on the board  
<sup>†</sup>Tepei Matsumoto <sup>‡</sup>Tatsuei Tsukiji  
Tokyo Denki University School of Science and Engineering

## 2. 実験・開発環境

実験開発環境を表1に示す。kinect とのインタフェースは、Microsoft 社提供の Kinect for Windows Ver14を使用した。

表1. 環境構成

プロセッサ	Intel(R)Core(TM)i3-2100 CPU @ 3.10GHz 3.10GHz	システム	32ビットオペレーティングシステム
OS	Windows7 Home Premium	開発環境	Microsoft Visual Studio 10.0
開発言語	C++	ドライバ	Kinect for Windows ドライバ ver14
SDK	Microsoft Kinect SDK v1.0	距離	kinect 本体とユーザー間の距離は約 1.5m で実験を行っている。

## 3. 研究内容

### 3.1 kinect による基本的なマウス操作

本研究では、マウスの代わりに kinect 指認識を用いてマウス操作を実現する。具体的には、基本的なマウス操作であるカーソルの移動及び左右クリックに加えて、ピンチ(2本の指で広げる、或いは縮めるような動作)によるズームイン・ズームアウトや、フリック(手・指を払う動作)によって画面を前の画面に戻す機能を実装している。ピンチやフリックはいまや一般的になったスマートフォンの操作にも使われており、多くの人に受け入れられやすかつ直感的にパソコンが操作できればと考えたため、この2つの動作を採用した。kinect によって手、及び指を検出して、その指の本数や配置によってカーソルの移動、左・右クリック、Ctrl キー押下+マウスホイールの回転・逆回転によるズームイン・アウト、画面バック、を実装した。動作例としては、一本の指を立てたときの指先がマウスポインタとなり、その指先の移動がカーソル移動となり、立っている指を1本から2本に増やせば右クリックとなる。

しかし既存研究のように、kinect からの相対位置で決まる3次元空間で手を動かしてカーソルを移動させようとすると、画面の端となる基準点がないためどこまでが左右上下の端か、中央がどこか不明瞭なために、思った通りの位置にカーソルを動かさないことが多い(図2参照)。

### 3.2 アクリル板を用いたのマウス操作

そこで本研究では、透明なアクリル板を用いることにした。kinect と指の間にアクリル板を挟み、板に取り付けたマーカーの内側を仮想の画面範囲とする。具体的には、アクリル板の四隅に基準となるマーカー（図1の赤丸）を配置し、それを OpenCV の関数で検出して、その内側を有効範囲として定める。これにより、マーカー間の距離の割合をそのままモニター内の距離の割合とすることで、マーカーを基準にマウス操作を行うことができる。またアクリル板の配置が変化しても、アクリル板上の2次元空間枠（図1の青色長方形）からディスプレイ上の2次元空間枠へのマッピングは不変となる。

なお、図2はアクリル板を使用せずに kinect でマウス操作を行い、カーソルを右上まで移動させた時の画像である。このように、kinect からの相対位置によっては、ディスプレイ空間に対応する上下左右の端が大きすぎたり小さすぎたりしてしまう。

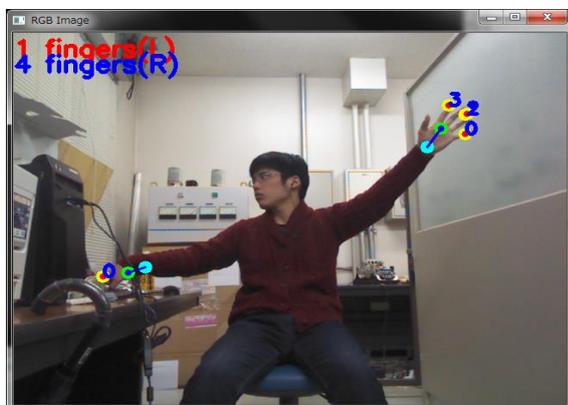


図2. 補助道具無し

図3は、中心からの半径が1cm 間隔の多重円の中心を指先でポイントして10回クリックした結果である。誤差として1cm の範囲で点が打てるという結果が得られた。チャタリングの発生を抑えるために距離感度を調整したり静止時間を設けることにより、さらに精度を高めることが可能と考えられる。

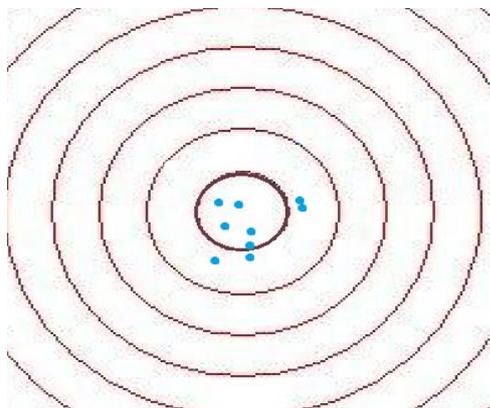


図3. kinect クリック操作で円の中心に点を打った結果

### 4. まとめ

kinect 指認識によりマウス操作を実現するために、アクリル板及びマーカーを用いて仮想の画面枠にすることで、マウスポインティングの精度が向上した。単純に四隅がわかりやすくなり、どのくらい手を動かせばどのくらいカーソルが動くのかが直感的にわかるため、意図した場所にマウスカーソルを動かせるようになった。

またアクリル板がない場合、手や指に近い形をしたものを誤認識してしまうとカーソル移動がぶれてしまう。しかしアクリル板がある場合、マーカーの内側にある手のみを判別してカーソルを動かすため、枠の外側にあるものに対して誤認識をしても問題なくなるため、ストレス無く操作が可能になる。

### 参考文献

- [1] 齋藤 裕佑：『仮想三次元作業空間における骨格認識を用いた操作手法に関する研究』人工知能学会全国大会（第26回）論文集,3L1-R-12-5頁（2012）
- [2] 中村 卓：『大画面環境におけるポインティングハンドジェスチャによるインタラクション手法の研究』電子情報通信学会論文誌, Volj96-D, NO4, pp.978-988（2013）
- [3] 中村 薫, 齋藤 俊太, 宮城 英人：『KINECT for Windows SDK プログラミング C++編』秀和システム（2012）