

# 手先認識センサを用いた仮想タッチパネルの開発

横道 政裕<sup>a)</sup>・井上 貴博<sup>b)</sup>・石原 誠康<sup>c)</sup>・甲斐 崇浩<sup>d)</sup>

## Development of a Virtual Touch Panel System with 3D Hand Motion Sensor

Masahiro YOKOMICHI, Takahiro INOUE, Seiko ISHIHARA, Takahiro KAI

### Abstract

Recently, many computer devices have been equipped with touch panels. They make user interfaces simpler by integrating the input and output for users. However, their size may restrict the fusibilities and it is not easy to control the accessibilities. To overcome these shortcomings, virtual touch panel systems have been studied by many researchers. This report proposes a virtual touch panel system which consists of HMD, monocular camera, and Leap Motion Sensor, which is one of the 3D hand motion sensor and can measure the position of the hand and the fingers accurately. The virtual touch panel is generated by user's finger motion and is displayed to HMD screen with being superimposed to the image taken by the camera. In order to implement these functions, i) a finger gesture to determine the position and the size of the touch panel is defined, ii) a calibration method for the camera and the Leap Motion Sensor is proposed. The effectiveness of the proposed system is examined by real world experiments.

**Keywords:** Virtual Touch Panel, Leap Motion Sensor, Finger Gesture, Calibration

### 1. はじめに

近年、PC やスマートフォン、タブレットなどの情報機器の多くはユーザーインターフェース用の機器としてタッチパネルを搭載している。タッチパネルを搭載した端末の特徴として、操作が直感的でわかりやすい、複数人数での端末の共有が容易、といった点が挙げられる。一方で、タッチパネルには物理的制約問題と、アクセスコントロール問題が存在するとされている。物理的制約問題とは、タッチパネルの大きさによって、表示可能な情報量や利用形態が制限されることであり、アクセスコントロール問題とは、パネルに表示された情報を不特定多数の人間が閲覧することが可能であることによってアクセス制限が困難であることである。

これらの問題を解決する方法として、ナチュラルユーザーインターフェース(NUI)用の機器を使用し、拡張現実感(AR)技術と3次元コンピュータビジョン技術を用いて仮想的なタッチパネルを生成・表示する手法が提案されている<sup>1)2)</sup>。加茂らが提案した AiR Surface<sup>1)</sup>では、HMD にステレオカメラを搭載することで、指先位置の3次元計測を行い、カメラで撮影した画像にタッチパネルを重畳表示する手法を用いている。また、Ubi<sup>2)</sup>では Kinect センサで取得した指先などの3次元位置を基にしてプロジェクタでタッチパネルを投影する手法を提案している。

どちらの手法でも物理的制約問題を解決しているが、前者では HMD を用いることでアクセスコントロール問題を回避している一方で、ステレオカメラを用いることによる3次元計測精度の低下や、環境条件へのロバスト性が低いなどといった欠点を有しており、後者では環境に対するロバスト性の高い3次元計測が可能であるものの、Kinect センサの大きさにより HMD への搭載は困難であり、アクセスコントロール問題を解決することは容易ではない。また、AiR Surface では、仮想タッチパネルのワールド座標系での位置を固定して表示するために、AR Toolkit などで使用されているマーカを使用しており、その点においても使用可能な環境が限定されるといった課題が存在する。

本研究では、これらの研究と同様に、HMD と3次元コンピュータビジョン技術、拡張現実感を用いて指先の位置に正確に仮想タッチパネルを表示するシステムを開発することを目的とする。照明条件に対するロバスト性や類似物体の誤検出、ユーザへの負担を考慮し、本研究ではマーカを使用せず、2013年に販売が開始された手先の検出に特化した小型の3次元モーションセンサである Leap Motion Sensor を用いる。このセンサは小型で HMD への搭載が可能であり、また、手先の3次元計測を高精度で行うことが可能である。

仮想タッチパネルを HMD のスクリーンに適切な位置に、適切な姿勢で重畳表示するには i) タッチパネルの操作に関するジェスチャの定義とセンサでの検出、ii) センサで検出した指先位置あるいは生成するタッチパネルの位置と表示画像上での位置との関係を求める、といった

a) 環境ロボティクス学科担当 准教授

b) 情報システム工学科 4年生

c) 大学院修士課程情報システム工学専攻 2年生

d) 教育研究支援技術センター 技術職員

機能が必要である。本研究では、特別な機器を必要としないことを念頭に置いて、タッチパネル生成用のジェスチャを定義し、ステレオカメラ等で使用されるキャリブレーションの手法を用いてオフラインでのカメラとモーションセンサの相対的な位置の推定を行う方法を提案する。提案したシステムの有効性を実環境での実験によって検証する。

## 2. システムの概要

本節では、本システムで使用する三次元手先認識センサである Leap Motion Sensor の概略を述べ、続けてジェスチャの設定、キャリブレーションの手法について述べる。

### 2.1. Leap Motion Sensor による指先の検出

Leap Motion Sensor は2基の赤外線カメラと3つの赤

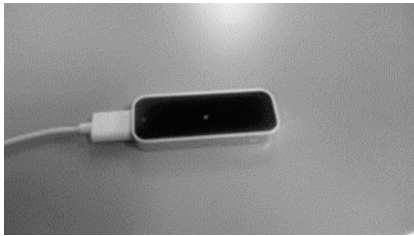


図1. Leap Motion Sensor の外観

外線照射 LED から構成されており、赤外線照射 LED によって照らされた手や指を2基の赤外線カメラで撮影し、画像解析によって3次元空間での手や指の位置を割り出すことにより、手先の認識を行っている。

また、手や指に限らずペンのようなポイントを指し示すツールを0.01mmの精度で認識することが可能であり、両手と10本の指をそれぞれ独立して同時に認識することができる。

Leap Motion Sensor を利用したソフトウェアを作成するための SDK が同社から無償で提供されており、自由にアプリケーションソフトを作成することが可能である。センサから出力され、ソフトウェアから使用することが可能な測定データは最新、または直前のそれぞれの手の指先の3次元的位置と、手のひらの中心位置と曲率の値(手のひらがどちらを向いているかを知るために使用する)等であり、2基のカメラからステレオマッチングを行っていれば取得が可能であると推測される密な3次元の点群情報などは利用することができない。

### 2.2 ジェスチャの設定

本研究では、操作用デバイスを使用した場合と比較して、追加の機器が不要で、その習得のためのコストが低いことから素手を使用する方法を用いる。また、素手での操作はユーザにとって日常で使用している自然な操作であり、よりスムーズに操作を行うことができることか

ら、素手を用いて仮想タッチパネルの操作を行う方法を採用する。

次に、それぞれの操作に対応したジェスチャを決定する必要があるが、その中で、仮想タッチパネルを新規に生成する際のジェスチャについては、以下の3つの条件を満たすものが望ましいと考えられる。

- 平面の領域を定義することが可能
- 操作を素早く行うことが可能
- 自然な動作で、無理のない動作である

それらの条件を満たすものとして、本研究では指を1本だけ伸ばした状態で作成したい大きさの仮想タッチパネルの対角線をなぞる、というジェスチャを採用する。このジェスチャを採用する理由として、PCのマウスで範囲選択を行う動作と似通っており、ジェスチャの内容も単純で無理なく実行できることが挙げられる。しかしながら、単純にこの動作を行うだけでは、初期位置と終了位置からその線分を対角線とする矩形を一意に定めることは不可能である。そこで、タッチパネルが作成される平面をカメラの光軸と垂直な面に限定することで2点から仮想タッチパネルの作成する面を一意に定めることが可能となる。具体的な手順は以下の通りである。

以下はカメラ座標系を、右方向をx軸、下方向をy軸、そしてカメラから前方に向かう方向をz軸とした右手座標系を用いる。

初期位置を $p(x_1, y_1, z_1)$ 、終了位置を $q(x_2, y_2, z_2)$ とすると、線分 $pq$ を $Z = z_1$ で定義される平面にz軸方向から正射影し、仮想的な終了点 $q'(x_2, y_2, z_1)$ を作成することにより、線分 $pq'$ を対角線とする鉛直平面上の矩形を定義することが可能となる。

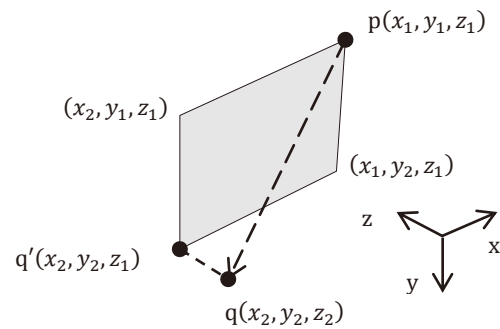


図2. 指先位置と仮想タッチパネルの関係

### 2.3 キャリブレーション

仮想タッチパネルをユーザに提示するためには、カメラで撮影された画像上の適切な位置に、適切な姿勢でタッチパネルを重畳表示させる必要がある。更に、仮想タッチパネルの手前に現実環境内の物体が存在する場合には、物体による遮蔽を考慮してタッチパネルの表示を行わねばならない。このような処理を実現するためには、カメラで撮影された現実環境上の物体の実際の位置と画

像上の位置との関係を求める必要がある。また、複数台のカメラや、カメラ以外のセンサを併用している場合には、それらの相対的な位置関係を求める必要もある。このような処理をキャリブレーションという。

本研究では、にカメラと Leap Motion Sensor との間のキャリブレーションを、

1. カメラの内部パラメータの推定
2. カメラと Leap Motion Sensor との間の外部パラメータの推定

とに分けて行う。これらの推定には Zhang の手法<sup>3)</sup>を拡張し、コンピュータビジョン用のライブラリ OpenCV で実装されている手法を用いる。

内部パラメータとは、カメラ座標系と画像座標系との間の変換を表現するパラメータであり、一般的にはカメラの焦点距離、主点位置、そしてレンズの歪パラメータを求めることに相当する。カメラ座標系での点の位置を  $(x, y, z)^T$  とし、画像上での撮像点の位置を  $(u, v)^T$  とすると、これらの位置の関係として、

$$\begin{aligned} \mathbf{u} &= f_x \cdot x'' + c_x, & \mathbf{v} &= f_y \cdot y'' + c_y \\ x' &= x/z, y' = y/z \\ x'' &= x' \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} \\ &\quad + 2p_1x'y' + p_2(r^2 + 2x'^2) \\ y'' &= y' \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} \\ &\quad + p_1(r^2 + 2y'^2) + 2p_2x'y' \end{aligned} \quad (1)$$

$$\mathbf{A} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

というモデルを考える。ここで、 $f_x, f_y$  はそれぞれ  $x, y$  方向の焦点距離[pix]であり、 $c_x, c_y$  は主点位置[pix]である。また、 $k_1, k_2, \dots, k_6, p_1, p_2$  はレンズの歪に関するパラメータであり  $r^2 = (x')^2 + (y')^2$  である。これらの歪パラメータをまとめて  $\mathbf{k}$  とする。

また、外部パラメータに関しては、ワールド座標系での点の位置  $(X, Y, Z)^T$  とカメラ座標での点の位置の関係

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2)$$

を用いる。ここで、行列  $\mathbf{R} = \{r_{ij}\} \in GL(3)$  は二つの座標系の回転や点対称変換を表す行列で、通常は回転行列となる。また、ベクトル  $\mathbf{t} = (t_1, t_2, t_3)^T$  は二つの座標系の並進移動を表す。(1)式と(2)式をまとめた画像上の点の位置とワールド座標系との間の変換式を次式で表す。

$$\mathbf{p} := \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_u(x, y, z) \\ a_v(x, y, z) \end{bmatrix} =: a'(X, Y, Z; \mathbf{R}, \mathbf{t}, \mathbf{A}, \mathbf{k})$$

Zhang の手法ではワールド座標系での位置と画像上の位置の複数のデータを基に、投影誤差の最小化問題

$$\min_{\mathbf{R}, \mathbf{t}, \mathbf{A}, \mathbf{k}} \sum_{i=1}^n \|\mathbf{p}_i - a'(X_i, Y_i, Z_i; \mathbf{R}, \mathbf{t}, \mathbf{A}, \mathbf{k})\|^2$$

を解くことでパラメータを推定する。最適化の手法としては、非線形最適化の一解法である、Newton 法や、Levenberg-Marquardt 法などが用いられる。

内部パラメータの推定

内部パラメータの推定においては、Leap Motion Sensor は関係なく、カメラのみで行うことが可能なので、カメラのキャリブレーションで一般的に用いられる平面上のパターンを撮影することでキャリブレーションを行う。この場合はパターンを設置した平面を仮想的に  $Z = 0$  で定義されるものとして複数枚のパターンを撮影してパラメータの推定を行うことが可能である。

外部パラメータの推定

外部パラメータの推定においては、Leap Motion Sensor で取得した指先位置と、画像上での指先の位置のデータを取得する必要がある。ワールド座標系としては、Leap Motion Sensor の座標系を用いるものとすれば、測定値がワールド座標系での位置となる。画像上の指先の位置の取得については、画像の肌色検出・輪郭抽出と凸曲線検出による手法が存在するが、誤差が大きく、また、本システムではキャリブレーションはオフラインで実行すればよいので、画像上でユーザの指先位置を目視で取得する方法を用いる。

## 2.4 表示画像の生成

キャリブレーションによって得られたパラメータを用いて Leap Motion Sensor で取得された指のワールド座標系での位置を画像に投影することが可能となる。具体的には(1), (2)式を用いて  $(X, Y, Z)$  から  $(u, v)$  を求めることが可能である。さらに、カメラから入力された画像に仮想タッチパネルを重畳表示し、HMD に表示する画像の作成を行う。2.1 節で述べた範囲選択姿勢をユーザがとったかどうかの判定はこの処理部で行っている。Leap Motion Sensor が指を 1 本だけ検出してから、設定した時間が経過した後の指の位置との距離が閾値以下であった場合、一つの点を指定しているとしてその座標を仮想タッチパネル作成の初期位置とする。終了位置の選択も同じように行い、仮想タッチパネルの重畳表示を行う。

初期位置決定後に指が Leap Motion Sensor の検知範囲外に移動する、範囲選択姿勢を解く、など終了位置を指定する状態でなくなった場合の対応として、初期位置を保持しておき、ユーザが再び範囲選択姿勢をとり Leap Motion Sensor の検知する指が 1 本になった段階で仮想タッチパネル作成を再開する方法などが考えられる。

## 3. 実験

### 3.1 実験機器と方法



図 3. 実験装置

図 3 に示すように、USB カメラと HMD、Leap Motion Sensor を組み合わせたデバイスを用いて、カメラと Leap Motion Sensor のキャリブレーションを行った後、指先で指定したカメラ画像上の位置に仮想的なタッチパネルが作成されるかどうかを確認する実験を行う。実験で使った USB カメラは BUFFALO 社の BSW13K10H で、画像の解像度は、640 x 480 である。キャリブレーションの手順は以下の通りである。

1. 事前にチェッカーパターンを用いてカメラの内部パラメータを推定する。
2. カメラ画像上ですべての指が Leap Motion Sensor に

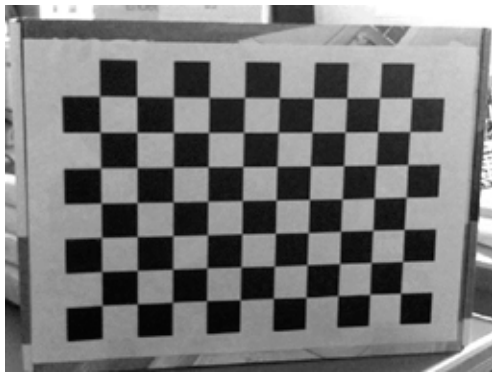


図 4. キャリブレーションパターン

よって認識されており、画像中にも撮影されていることを確認し、画像をキャプチャする。

3. キャプチャした画像上の指先の位置を手動で特定し、Zhang の手法で行列  $R$  とベクトル  $t$  を推定する。

### 3.2 実験結果

手順 1 で使用したパターンを図 4 に、推定された内部パラメータを表 1 に示す。ただし歪みパラメータについては、 $k_3, k_4, \dots, k_6 = 0$  として推定を行っている。

表 1. カメラの内部パラメータ

項目	数値
焦点距離(pixel)	$1.081 \times 10^3$
主点位置(pixel)	$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} 3.260 \times 10^2 \\ 2.291 \times 10^2 \end{bmatrix}$
歪み係数	$\begin{bmatrix} k_1 \\ k_2 \\ p_1 \\ p_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 2.756 \times 10^{-1} \\ -3.720 \times 10^0 \\ 1.851 \times 10^{-3} \\ 2.306 \times 10^{-3} \\ 3.046 \times 10^1 \end{bmatrix}$

次に、手順 2, 3 で使用した画像と、外部パラメータの推定結果を図 5 に示す。推定された外部パラメータは以下の通りである。



図 5. 外部パラメータの推定に用いた画像

$[R: t]$

$$= \begin{bmatrix} 0.99902 & 0.00405 & 0.04398 & -20.05487 \\ -0.00154 & 0.99837 & -0.05706 & -31.78556 \\ -0.04414 & 0.05694 & 0.9974 & -56.2061 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

図 5 での点は、初期推定値  $R = I_{3 \times 3}$ ,  $t = [0, 0, 0]^T$  における Leap Motion Sensor で認識された指先位置の投影結果である。推定された外部パラメータを用いて Leap Motion Sensor で認識された指の位置を投影したものを図 6 に示す。図より、正確に投影されていることがわかる。



図 6. 画像への指先位置の投影結果

推定されたパラメータを用いて指先を様々な位置に移動させて Leap Motion Sensor で計測した指先位置を重畳表示した結果を図 7、図 8 に示す。図 7 は、指先を画像の左端から右端へと移動させ、図 8 では指のカメラに対しての距離を変えて検出を行っている。図から、左右に

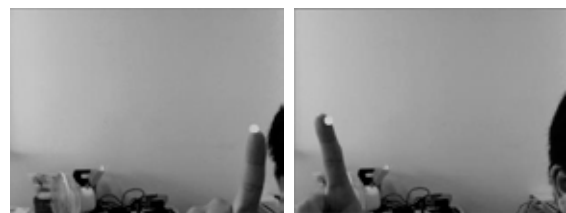


図 7. 左右端への移動時の投影結果



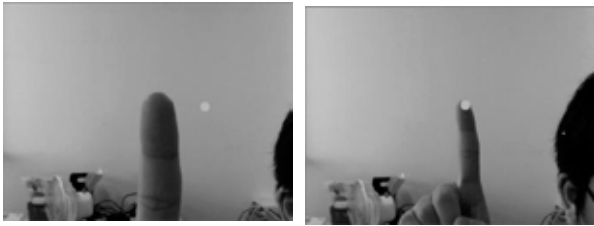


図 8. 前後への移動時の投影結果

動かした場合は正確に指先を追従できているが、カメラからの距離を変えると認識に誤差が発生することが確認できる。

### 3.3 仮想タッチパネルの作成

次に、指先を移動させ、仮想タッチパネルを表示させた結果を図 9 に示す。左の図の水色の点が最初に選択した点で、右の図で緑色の点の地点を指で指定することで仮想タッチパネルを表示させている。

図から、指でカメラ画像上の 2 点を指定することで仮想

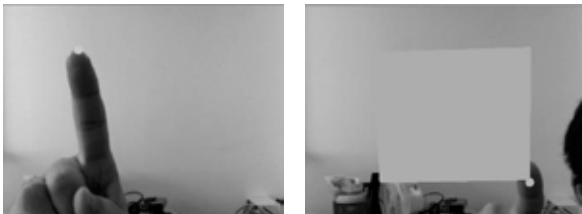


図 9. 仮想タッチパネルの生成結果

タッチパネルが重畳表示されていることが確認できる。しかし、指を一定の位置に留めることが予想以上に難しく、仮想タッチパネルの作成に手間取る可能性があることが判明した。

## 4. 結論

本研究では、カメラと 3 次元モーションセンサを用い

て仮想タッチパネルを生成するシステムを提案した。カメラとセンサ間のキャリブレーションを行い、指先で指定した位置に仮想タッチパネルを重畳表示することが可能であることが実験から確認された。今後の課題としては、現在はユーザが指先の位置を手動で指定しているの、ユーザに対する負担を減らすためにキャリブレーション作業を自動で行うことや、精度向上のために複数枚の画像を用いてキャリブレーションを行うことが挙げられる。

また、現時点ではカメラに対する相対位置しか計測することができないため、ユーザの頭部が移動した際に、仮想タッチパネルもそれに追従して移動してしまう。文献 1) のシステムではマーカを環境内に設置することでカメラの自己位置推定を行っているが、環境への制約を減らすために、カメラによる SLAM 技術を適用することで自己位置推定を行う手法を開発する必要がある。その他にもユーザへの自然な表示を行う為には、オクルージョンへの対策やヘッドマウントディスプレイの左右のディスプレイにステレオ画像に加工したカメラ入力画像を表示させる方法などが挙げられる。

## 参考文献

- 1) 加茂浩之: AiR surface :拡張現実感を用いた仮想タッチパネルインタフェース, 平成 22 年度 筑波大学情報学群卒業論文, 2011.
- 2) Ubi ホームページ: <http://www.ubi-interactive.com/>(2014 年 2 月 24 日アクセス)
- 3) Z. Zhang: A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 11, pp. 1330–1334, 2000.