

Leap Motion による手のトラッキング

5年B組 中根 彩乃

指導教員 藤野 智美

1. 要約

本研究では、手の三次元の動きをセンシングし、画面上の画像を操作するシステムを開発した。なお、開発環境は Xcode を、手の動きを認識するセンサーとして Leap Motion を用いた。

キーワード Leap Motion、Xcode、三次元空間、画像解析

2. 研究の背景と目的

今日、様々なデジタルアートがあるが、その中でも人の動きに伴って映像が変化するものに興味を持った。そこで、本研究では手の動きに伴って画面上で図形を描くシステムを作成した。人の動きを認識するためのセンサーには、手の動きを認識する事に特化した Leap Motion を用いた。

の中である。

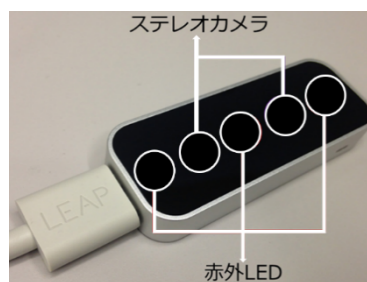


図1 Leap Motion

3. Leap Motion について

Leap Motion とは、2012 年に Leap Motion 社から販売された、2 基の赤外線カメラと赤外線照射 LED から構成された小型のセンサーである。Leap Motion に手をかざすと、赤外線 LED に照らされた手や指を赤外線カメラで撮影し、画像解析により 3D 空間での手や指の位置を割り出すことができる。両手と 10 本の指を 0.01mm の精度で認識することが可能で、検出数については、両手だけでなく、Leap Motion センサーの範囲に入っていればいくつでも検出可能なため、複数名でも利用できる。認識可能な範囲はデバイスの 2.5cm 上方、約 60×60×60cm の逆ピラミッド型の 3D 空間

4. 研究内容

4.1 手の位置情報から手を描画する

<方法>

【1】ヘッダファイルを定義する

C++で使用出来る vector 配列(動的配列)を使用した。この手法により、通常の配列のようにサイズをあらかじめ指定せず、実行時に自動的にサイズ変更を可能にし、手が動いたときに配列が自動的に変化するようにした。

【2】vector 配列を宣言する

vector 配列では三次元座標を扱い、三次元ベクトルを格納する。位置と速度の座標を管理することで手の位置と動作の検知が可能となる。また、この配列を用いることで

指や関節の位置情報が保存できるようにした。なお、一つの手は 20 個の三次元座標の集まりで表されている。

【3】指の位置情報を格納する

- ① Leap Motion で取得した手の位置情報を simple Hands Vector 配列に格納する。
- ②以前に保存した手の位置情報を消去し、手を画面上に描画できるようにマッピングを行う。
- ③手の関節や指先の新しい位置情報を格納し、手が認識され、かつ更新された時に手の座標を更新する命令を呼び出す。

【4】手を描画する

手の数だけ for 文を回して手の関節や指先などの描画を行う。

- ①【3】の③で格納したそれぞれの手の指先、関節の位置を各指四つの球体で表す。
- ②指の各点を線で結び、骨の位置を線で表現する。

<結果>



図 2 手の描画

4.2 手の中心間の距離を直径とする球を描く

<方法>

【1】手の中心を示す球を描く

- ①新たに手の中心座標を保存する vector 配列を宣言する。
- ②指の位置情報を格納した場合のように、手の中心座標を vector 配列に追加するプロ

グラムを作成する。

- ③取得した手の中心座標に球を描く。

<結果>

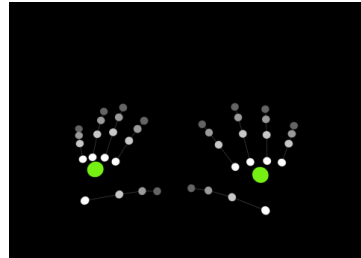


図 3 手の中心を示す球の描画

【2】手の中心座標を使って球を描く

- ①手が 2 つの時に球体を描くことを指定する条件分岐を、if 文を用いて書く。
- ②手の中心座標の 2 つからその距離を出し、これを直径とする球を描く。

また、球を描くとき、中心座標は手の中心である二点の中心となるようにとる。

<結果>

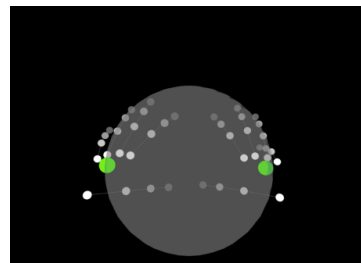


図 4 球の描画

4.3 手の中心間の距離の範囲によって画像を変化させる

<方法>

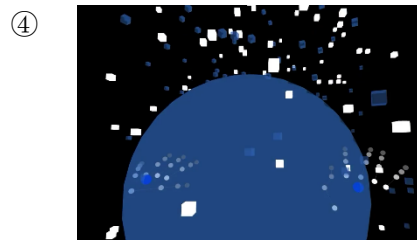
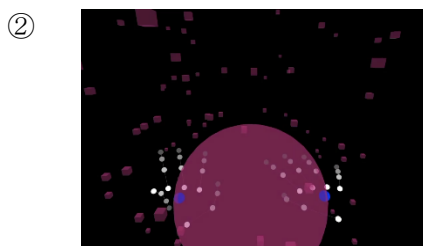
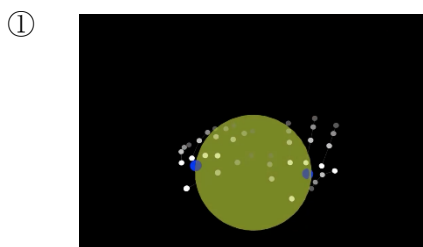
距離の範囲を 4 つ指示し、表示する画像を指定する。距離が狭い範囲から順に以下の操作を指定した。

- ① 4.2 の方法で球を描く。
- ② 時間で色が変化する立方体を表示する。

- ③ 白い立方体を表示し、 x 軸回転させる。
④ 白い立方体を x 軸で回転させ、かつ時間によって色が変化する立方体を y 軸で回転させる。

このとき、時間によって色が変化する部分では、プログラムが開始してからの時間を計算し、HSBの値を時間ごとに変化させた。HSB色空間を操作することで色彩、彩度、明度を変化させることができる。なお、色彩の指定部分に時間を計算した結果を代入することで色の変化を可能にした。また、このプログラムでは2つの手によって表現される球を、前述したプログラムを用いることで時間によって色を変化させた。

<結果>



5. 今後の展望

今回は手の動きを認識して画面の中に映し出したが、今後は手だけでなく全身の動きを認識する他、変化する画像や映像を画面の中だけで表すのではなく、大きな領域に映し出したいと考えている。その方法として、Kinectセンサーを用いて全身を認識することを検討している。

6. 参考文献

- [1]Leap SDKで指を検出してみよう
<https://www.buildinsider.net/small/leapmotioncs/002>
[2]Life is Tech ! TECH FILE
<https://members.life-is-tech.com/textbooks/259/html>

7. 謝辞

今回の研究を行うにあたり、顧問の藤野先生にご指導いただきました。この場を借りて、深く御礼申し上げます。