

## 7 Leap Motion を用いた入力操作の検討

高橋一樹

指導教員 ソソラ先生

### 1. はじめに

指の動きを検知してコンピュータを操作することが可能な新しいデバイスである「Leap Motion」に興味を持った。

音声認識で機器の操作ができる技術がある。これを手の動き（ジェスチャー）で代用できないかと考え、このテーマを選定した。

### 2. 研究概要

#### 2.1 目的

Leap Motion で手の動き（ジェスチャー）を取得し、あらかじめ記録してあるデータと比較する。ジェスチャーを判別し、意味のある文字や数字などで出力することを目指す。



図 1 完成イメージ

#### 2.2 Leap Motion とは

手のジェスチャーを認識する小型デバイス。

USB で接続することで使用できる。

検知できる範囲は半径 50 センチ、中心角 110° で 1/100mm の精度で認識できる。

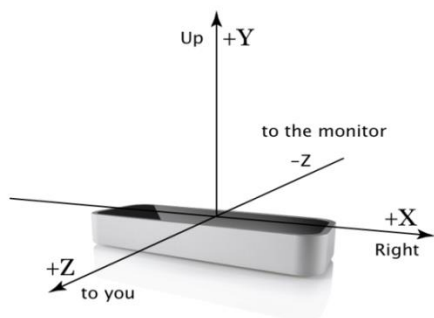


図 2 Leap Motion Controller

#### 2.3 Unity とは

統合開発環境を内蔵したゲームエンジン。

Windows や Mac など複数のプラットフォームに対応している。このエンジン自体は C 言語/C++ で書かれているが、C#, Unity Script(JavaScript)にも対応している。

本研究では C# で開発し、認識したジェスチャーの判別と表示に使用する。

#### 2.4 開発環境

OS	Windows7/8.1
ゲームエンジン	Unity 5.4.0 f3
使用機器	Leap Motion
使用言語	C#

### 3. ジェスチャー認識プログラム

#### ステップ 1

Leap Motion で取得した手を Unity 上に表示。

手の関節をセンサで取得し、関節間を円柱のオブジェクトで補間することで手のモデルとした。しかし、下記の問題が発生した。

- ◆ すべての関節を取得するためには負荷がかかること。
- ◆ 使用しない座標のデータが増えることで処理が複雑になってしまうこと。

解消するために指先 5 点と手の甲の計 6 点のみ表示に変更とする。



図 3 表示される手のモデル（仮）

## ステップ 2

## ■ ジェスチャーの登録

関節ごとの座標データを記録する.

今回使用する主なデータ群

hand : 手に関する情報

◆ type…手の左右

◆ palmPosition…手の中心位置

finger : 指に関する情報

◆ type…どの指か

◆ tipPosition…指先

◆ dipPosition…第一関節

◆ pipPosition…第二関節

◆ mcpPosition…指の付け根

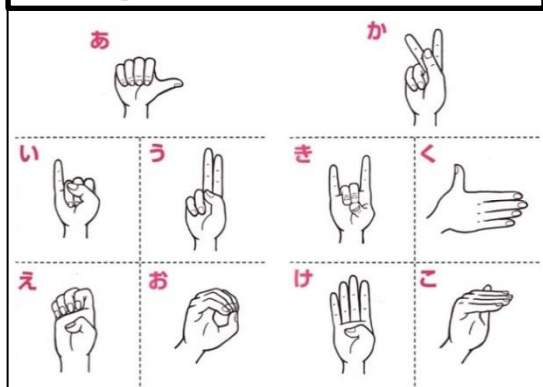


図 4 手話（指文字）の一部

## ■ 座標情報の比較

比較アルゴリズムを作成し, 判定結果を出力する.



図 5 「チョキ」を判別した場合

## 問題点

座標の完全一致では判定が厳しく不可.

## 改善策

登録されたデータと完全に一致するジェスチャーを再現することは困難と判断し, 座標にある程度の誤差を加味して判定する方法に切り替えた.

## 結果

座標に誤差を含めると判別が可能となったが誤認識が多発.

## 4. 終わりに

## ■ 進捗状況

開発環境の構築とサンプルや参考サイトでの基礎的な学習は終了.

Leap Motion で取得した手を Unity 上で再現することができている.

現在は取得したジェスチャーを判別するためのアルゴリズムを考案している.

精度は低いがグー, チョキ, パーなど簡単な形を判別できるようにした.

## ■ 今後の課題

判別アルゴリズムの完成

グー, チョキ, パーでの精度向上を目指す.

他のジェスチャーの読み込み.

処理負荷の軽減・高速化

動作フレームの検討.

操作性の向上

Leap Motion の認識範囲を視覚化.

認識できる高さや手の角度について検討.

## ■ 今後の作業計画

12 月	モーショントレースアプリ開発
1 月	アルゴリズムの作成
2 月	発表資料作成

## 5. 参考文献

Unity : <http://japan.unity3d.com/>

Leap Motion :

<https://www.leapmotion.com/?lang=jp>

手話 :

<https://ja.wikipedia.org/wiki/%E6%89%8B%E8%A9%B1>