

平成 27 年度 公立はこだて未来大学卒業論文

# Leap Motion による教育をインタラクティブにするシ ステムの提案

一ノ瀬 智太

情報アーキテクチャ学科 1012211

指導教員 美馬 義亮

提出日 2016 年 2 月 29 日

**Title in English**

**— one two three four five six seven eight nine ten —**

by

Tomohiro Ichinose

BA Thesis at Future University Hakodate, 2016

Advisor: Prof. Advisor, Coadvisor: Prof. Coadvisor

Department of Complex Media Architecture

Future University Hakodate

February 29, 2016

**Abstract**— Teachers use a blackboard and chalk in the field of school education. On the other hand, students use pencil and notebook was mainstream. However, classes that use a personal computer and tablet devices have been introduced in recent years. It is that the exchanges of the information from a teacher to a student are unilateral to be common when I compared the recent school education with the conventional thing. In this study, the purpose is to implement an application for realizing interactive teaching operations between teachers and students in the field of education, and evaluate usefulness.

**Keywords:** Keyrods1, Keyword2, Keyword3, Keyword4, Keyword5

**概 要:** 学校教育の場において教師は黒板とチョークを使い、学生は鉛筆やノートを使った授業が主流であった。しかし、近年ではパーソナルコンピュータやタブレット端末を使った授業が取り入れられている。近年の学校教育を従来のものと比較した際に共通することは、教師から学生への情報のやり取りが一方的なことである。本研究では、教育の場において教師と学生間での対話的な授業運営を実現するためのアプリケーションを実装し、その有用性を評価することを目的とする。

**キーワード:** キーワード 1, キーワード 2, キーワード 3, キーワード 4, キーワード 5

# 目次

<b>第1章 序論</b>	<b>1</b>
1.1 背景 . . . . .	1
1.2 目的 . . . . .	1
1.3 研究目標 . . . . .	1
<b>第2章 問題設定・研究アプローチ</b>	<b>2</b>
2.1 問題設定 . . . . .	2
2.2 研究アプローチ . . . . .	2
<b>第3章 関連研究</b>	<b>3</b>
3.1 クリッカー . . . . .	3
3.2 サイボウズ Live[] . . . . .	3
<b>第4章 提案するアプリケーションの概要</b>	<b>4</b>
4.1 提案するアプリケーションの特徴 . . . . .	4
4.2 クライアント側のプログラムの役割 . . . . .	4
4.3 サーバー側のプログラムの役割 . . . . .	5
4.4 実装方法 . . . . .	6
4.4.1 言語・ツール・開発環境 . . . . .	6
4.4.2 Cinder フレームワークについて . . . . .	6
4.4.3 Leap Motion について . . . . .	7
<b>第5章 実験と評価</b>	<b>14</b>
5.1 実験について . . . . .	14
5.1.1 評価方法について . . . . .	14
5.2 アンケート . . . . .	14
<b>第6章 考察</b>	<b>15</b>
6.1 評価結果 . . . . .	15
6.2 評価結果 . . . . .	15
<b>第7章 結論と今後の展開</b>	<b>16</b>
7.1 まとめ . . . . .	16
7.2 今後の方針 . . . . .	16

# 第1章 序論

## 1.1 背景

学制百年史 [1] によると大学の誕生は明治時代に遡る。一斉授業の場において、教師は知識を与え学生はそれを吸収する。こうした場において、一部の学生には頬杖をついたり、居眠りをしたりなどが散見される。その原因として、日々の生活の疲れだけでなく、教師の説明が理解できず、授業についていけないなどのストレスが挙げられる。こうした学生には他の学生に比べて学習意欲が低下しているといえる。

## 1.2 目的

本研究では、こうした学習意欲の低下に着目し、これを解決するために教師と学生の間に関向性をもたせ、学習意欲の向上を図ることを目的とする。その方法として、Leap Motion というセンサが手や指の位置、方向、曲げ、ジェスチャー等を検知できるセンサを用いて、関向性を実現するアプリケーションを実装し、学習意欲が向上することを目指す。具体的には学生が授業中に抱いている「助けてほしい」や「もう一度説明してほしい」などのメッセージを Leap Motion で入力し、教師に伝えるということを行う。

## 1.3 研究目標

## 第2章 問題設定・研究アプローチ

### 2.1 問題設定

本研究での問題は一斉授業の場において双方向性を実現し、学習意欲の向上を目指す。Leap Motion というセンサが手や指の動きを検知し入力操作を行い、その時に音を立てることがないという性質をもっているのをこれを使用する。本研究での一斉授業の場の位置付けとして、教師と学生の双方がパーソナルコンピュータを使用していることを想定する。具体的な例として、教師はパーソナルコンピュータで資料をプロジェクターに投影しながら授業を行い、一方、学生はメモをとったり課題に取り組むために使用しているものとする。

### 2.2 研究アプローチ

ここでは実際のアプリケーションを使うときのシナリオを記述する。まず、実際の授業の中で学生は「大きな声で話してほしい」や「もう一度説明してほしい」といった感情を抱く。これらの感情を Leap Motion を使ったアプリケーションで発信し、プロジェクターに投影する。その発信を基に教師がアクションを起こし、対話的な授業運営を実現する。一方で、学生側の発信が多くなり過ぎた場合、教師へのストレスが懸念される。そのような場合は、学生側からの発信を制限したり、教師側が Leap Motion を使ってストレスを軽減できるようにする。

## 第3章 関連研究

ここでは対話的授業運営の関連研究としてクリッカーを、意思決定ツールとしてサイボウズ Live を取り上げる。

### 3.1 クリッカー

双方向性授業の実現としてクリッカー [4] を用いた事例がある。クリッカー (授業応答システム) とは赤外線リモコンによる学生解答システムである。授業中に教師が問題を出し、学生がクリッカーを用いて解答するというものである。クリッカーを用いた結果として、「役に立つ」という問いに対して「役に立つ」と答えた学生が 6 割を上回り、「勉強する気になったか」という問いに対しては「やる気が出た」という学生が 7 割を上回っている。

### 3.2 サイボウズ Live[]

サイボウズ Live はサイボウズ社が開発した意思決定ツールである。主な機能として、以下の 5 つが紹介されている。

- タイムライン：メンバーに聞きたいこと、伝えたいことを書く
- 掲示板：掲示板のコメント欄でメンバーとディスカッションする
- イベント：プロジェクトのイベントやスケジュールを登録・共有できる
- ToDo リスト：メンバーと ToDo を共有できる
- 共有フォルダ：1 グループにつき 1G バイトまで無料でファイルを保存可能

また、2014 年 2 月 26 日よりチャット機能が追加された。チャット機能の利点は以下の 3 つが紹介されている。

1 つ目は、PC でもスマートフォンでもメッセージがすぐ届く点である。サイボウズ Live のチャット機能は PC だけでなく、スマートフォンでも使うことができる。投稿したコメントは相手の画面にただちに用事されるので、オンラインのミーティングがスムーズに行える。

2 つ目は、複数人でも 1 対 1 でもチャットが出来る点である。複数人のメンバーを指定して、チャットスペースを作成できる。また、サイボウズ Live 上で知人と繋がると、自動的に 1 対 1 のチャットスペースが作成されるので、簡単にチャットのやりとりをスタートできる。

3 つ目は、グループ機能と併用すれば、情報の整理が楽になるという点である。チャットのやりとりで決定したことは、先に記述した ToDo リストに登録しておくことができる。

## 第4章 提案するアプリケーションの概要

提案するアプリケーションは、授業やゼミ等で聞き手が話者にメッセージを発信するための意思決定ツールである.

#### 4.1 提案するアプリケーションの特徴

このアプリケーションの特徴は、主に2つある。

1つ目は, Leap Motion を使ってリアルタイムに入力操作を行うことである. ほにやら  
ららほにやらららほにやらららほにやらららほにやららら

2つ目は、クライアント側とサーバー側の二つのプログラムに分かれていることである。このアプリケーションでは、クライアント側とサーバー側間のデータを送受信する手段として、ソケット通信を行った。

## 4.2 クライアント側のプログラムの役割

クライアント側は、聞き手が発信したいメッセージを発信するためのものである。クライアント側からサーバー側のプログラムへ発信される値は以下の5つである。

## クライアント側から発信する値について

- アカウントナンバー
- 手の数
- サークルジェスチャー
- スクリーンタップジェスチャー
- メッセージナンバー

まず、アカウントナンバーは固定値であり、それぞれの聞き手には予め番号が振り分けられている。これにより誰がどのような値を送ったかを特定しサーバー側で処理する。

次に、手の数は聞き手が Leap Motion に手をかざしているかの判定に使う。発信されている値が 0 の時は、手をかざしていないことを発信し、0 よりも大きい値の時、聞き手が手をかざしていることを表す。

サークルジェスチャーとスクリーンタップジェスチャーは、プログラムを起動してからジェスチャーが行われた回数を発信する。この2つのジェスチャーは話し合いをしているとき

やアプリケーションを使っているときに自然に取れる値である。

最後にメッセージナンバーは、選択したメッセージを表す。発信するメッセージは9つ用意されており、それぞれ対応した値が発信される。メッセージナンバーは-1と0から8の値で構成されており、-1のときは選択されていないことを表し、0から8の値ではそれぞれのメッセージ送られる。

これらは最後1つの文字列になり、値をカンマで区切られた後、サーバー側へ発信される。

## クライアント側の UI

クライアント側の UI はこのようになっております。クライアント側の UI はこういう意図で作られました。それぞれのメッセージとメッセージナンバーの対応は以下のようになっております。クライアント側の UI はこのようになっております。クライアント側の UI はこういう意図で作られました。それぞれのメッセージとメッセージナンバーの対応は以下のようになっております。クライアント側の UI はこのようになっております。クライアント側の UI はこういう意図で作られました。それぞれのメッセージとメッセージナンバーの対応は以下のようになっております。クライアント側の UI はこのようになっております。クライアント側の UI はこういう意図で作られました。それぞれのメッセージとメッセージナンバーの対応は以下のようになっております。

## クライアント側からメッセージを発信する手順について

クライアント側で行われる動きは以下のとおりである。(図参照)  
まず、聞き手は、自分がメッセージを発信したいと思った時、Leap Motion を使ってメッセージの選択をする。メッセージを選択した直後、サーバー側プログラムへ値が送られる。以降、送られた値はサーバー側で処理される。

## 4.3 サーバー側のプログラムの役割

サーバー側では、聞き手のメッセージを集計し、その集計結果を発信するためのものである。まず、送られてきたメッセージを区切りで分解する。その値を構造体へ代入する。

### メッセージナンバーの集計

送られてきたメッセージは構造体へ代入後、どのメッセージが多く発信されたかを算出する。構造体へ代入されるメッセージナンバーは0から8の値であり、-1のときは代入されない。これにより聞き手が発信した最新のメッセージナンバーが構造体へ代入される。一番多く受け取ったメッセージナンバーを算出後、その値にあった描写処理を行う。



## 描写処理について

聞き手から送られたメッセージに関してまるまるのような描写処理が行われる. 聞き手から送られたメッセージに関してまるまるのような描写処理が行われる. 聞き手から送られたメッセージに関してまるまるのような描写処理が行われる. 聞き手から送られたメッセージに関してまるまるのような描写処理が行われる. 聞き手から送られたメッセージに関してまるまるのような描写処理が行われる. 聞き手から送られたメッセージに関してまるまるのような描写処理が行われる. 聞き手から送られたメッセージに関してまるまるのような描写処理が行われる.

## 4.4 実装方法

### 4.4.1 言語・ツール・開発環境

使用言語は主に C++ である. 一部, C 言語での記述もある. IDE は Xcode で Leap Motion を入力装置として使う. また開発環境は表 4.1 の通りである.

表 4.1: 開発環境	
OS	OS X Yosemite Macbook Air
プロセッサ	1.6 GHz Intel Core i5
メモリ	4GB 1600 MHz DDR3

### 4.4.2 Cinder フレームワークについて

Cinder は C++ で書かれたフレームワークである. 画像や動画, 音声処理に長けており, ライブラリが用意されている. 今回は, アプリケーションの作り方が本で紹介されていたことと Cinder を使った導入事例が多くなってきていることから, Cinder を使って開発を進めることにした.

## openFrameworks との違い

ここでは openFrameworks[] について述べる. openFrameworks は C++ のオープンソースツールキットである. openFrameworks は, Cinder と同様に, 描写処理等に長けたフレームワークでもあり, Leap Motion を使ったアプリケーションで使われるフレームワークとして, この2つが使われていることが多い. openFrameworks は Cinder よりも歴史がある. openFrameworks は 2004 年に開発され広まっていったのに対し, Cinder は 2010 年から広まった. 図 4.1 の google トрендでは, 近年, Cinder の方が openFrameworks よりも流行りであることがわかる.

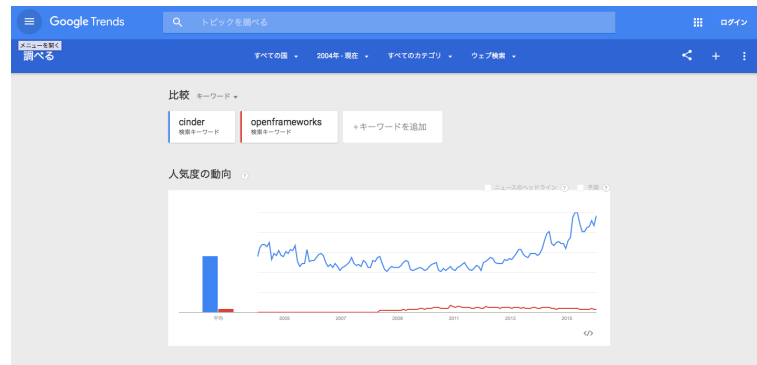


図 4.1: Cinder と openFrameworks

#### 4.4.3 Leap Motion について

Leap Motion(図 4.2 ) とはアメリカの Leap Motion 社が 2013 年にリリースした手や指を検知することに特化したセンサである. Leap Motion は教育や医療など幅広く使われている. 開発のドキュメントとしては, Java や C++ などの 6 つが用意されている (図 4.3 参照).

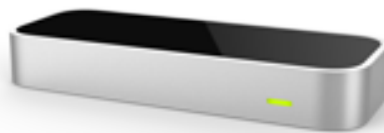


図 4.2: Leap Motion



図 4.3: 開発ドキュメント

## Leap Motion を使用するために

Leap Motion を使用するために、まず自身の PC にインストーラがインストールされていなければならない。ここではその手順を示す。

Leap Motion を使うためには、3つの手順が必要である。

1. インストーラを公式ホームページのセットアップ (図 4.4) からダウンロードする。
2. ダウンロードしたものを解凍する。図 4.5 のような画面が表示されるので、指示に従ってインストールする。
3. メニューバーにアイコン (図 4.6 ) が表示されていたら完了。

## Leap Motion を使用する前の確認事項

Leap Motion のアプリケーションを使用する前に確認しておくべきことを2つあげる。以下に述べることは図 4.6 の Leap Motion のアイコンから設定を選択し図 4.7 の Leap Motion の設定画面から確認することができる。これらの設定後、Visualizer 等で確認するとなお良い。

1つは、操作の高さを調節しておくことである。Leap Motion では自動調節機能があり、操作の高さが可変になるが、初めて Leap Motion 使うとこれに気づきにくい。実際に使ってもらい、手が疲れるという声があった。ユーザーによって使いやすさに違いが出るので確認しておくべきである。

もう1つは、道具の設定がオフになっているかどうかを確認することである。Leap Motion ではペンなど棒状の道具を検知できる。今回のアプリケーションでは、道具を使って操作することを想定しなかったため、道具の検知を行っていない。もし、道具を使ってのアプリケーションを製作する場合では、チェックされていないと検知できないため確認する必要がある。



図 4.4: Leap Motion のセットアップ

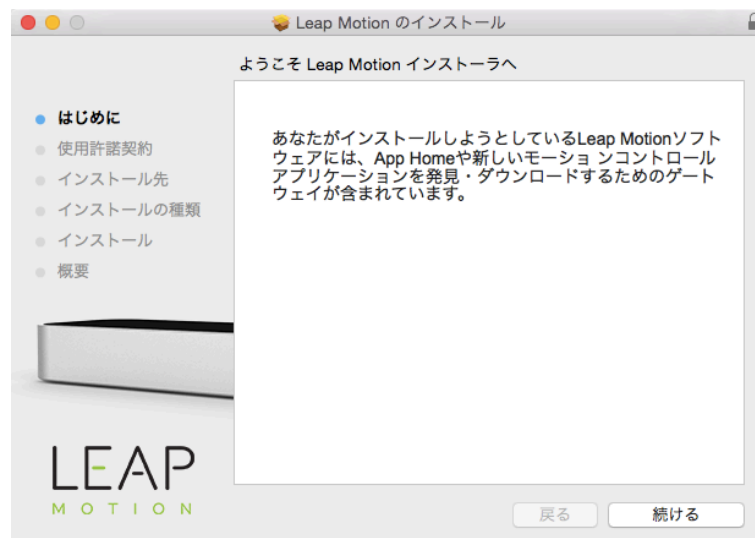


図 4.5: インストール画面



図 4.6: Leap Motion のアイコン

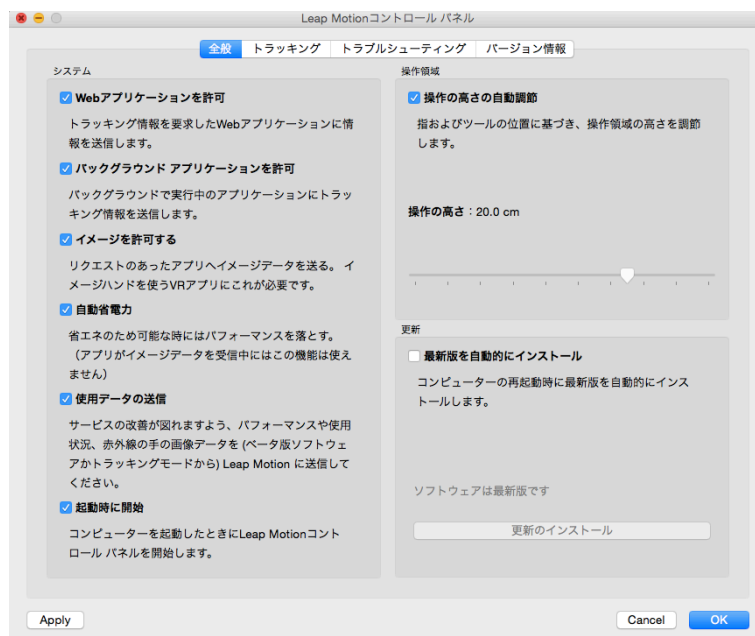


図 4.7: Leap Motion の設定画面

## Leap MotionSDK のダウンロード

Leap Motion を使ったアプリケーションを開発するために SDK を自分の PC にインストールする必要がある。Leap MotionSDK も Leap Motion インストーラーと同様に、公式ホームページのセットアップ (図 4.4) よりダウンロードできる。

## Leap Motion の評価

Leap Motion の入力装置としての特性を追求した。ここでは Leap Motion の良い点と欠点をあげる。Leap Motion の良い点は、3 つある。

1 つ目は、Leap Motion では手や指の位置や方向、ジェスチャーなどを検知し、リアルタイムで入力操作ができるという点である。ジェスチャーは 4 種類あり、以下に示す。先にも述べたが今回のアプリケーションではサークルジェスチャーとスクリーンタップジェスチャーの 2 種類を検知する。

- サークルジェスチャー：スクリーンに向かって円を描くジェスチャー (図 4.8)
- スクリーンタップジェスチャー：スクリーンに向かってタップするジェスチャー (図 4.9)
- スワイプジェスチャー：スクリーン方向にスワイプするジェスチャー (図 4.10)
- キータップジェスチャー：Leap Motion 方向にタップするジェスチャー (図 4.11)

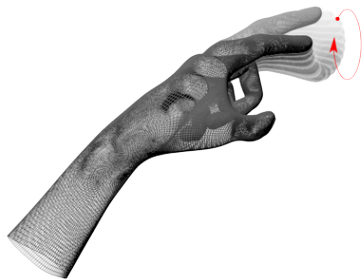


図 4.8: サークルジェスチャー

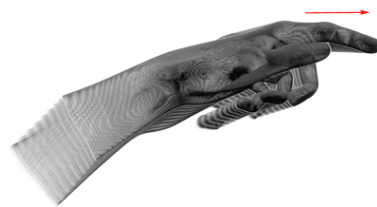


図 4.9: スクリーンタップジェスチャー

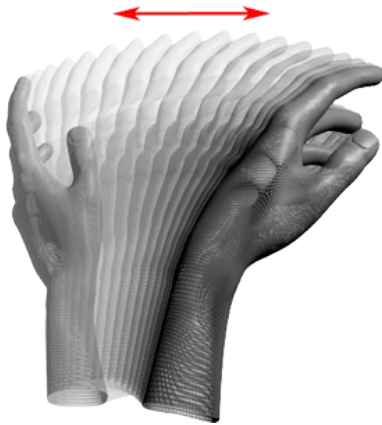


図 4.10: スワイプジェスチャー



図 4.11: キータップジェスチャー

2つ目は、指の曲げを判定できるので、入力パターンが豊富であることがいえる。Leap Motion はどの指が曲げられているか、曲げられている本数は何本か、どのくらい曲げられているか判定することができる。また、右手と左手の判定もできるので、以上のことを使えば入力パターンを増やすことが可能である。

3つ目は、Leap Motion をキーボードやマウスなどの入力装置と比較したとき、音を出さずに入力できるという利点がある。キーボードやマウスにはボタンがあり、押すとカチッと音になる。しかし、Leap Motion は指の曲げやジェスチャー等が入力操作として扱われるので音が出ることはない。

一方で、Leap Motion の欠点として次の2つのことを述べる。

1つ目は、センサに不得意な方向があることである。Leap Motion は赤外線センサーで感知しており、図 4.12 のような逆四角錐の範囲で検知される。そのため手を地と垂直に向けると手のひらで指を隠してしまうため、誤検知される危険性がある。よって、意識して地と平行にかざすことで Leap Motion が誤操作が発生しにくいことがわかった。

もう1つは、複数接続が不可能であることである。1台の PC に接続できるのは1つの Leap Motion で、2つ以上繋いだ場合、1台目の Leap Motion が優先され、2台目は機能しないことがわかった。

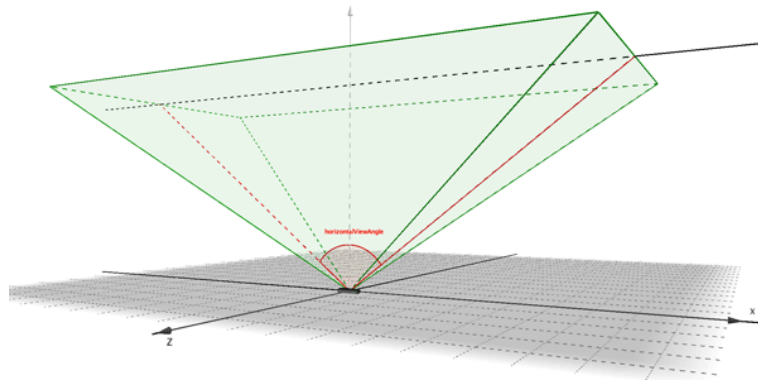


図 4.12: Leap Motion のセンサ範囲

ソケット通信について



## 第5章 実験と評価

### 5.1 実験について

ここでは, FUN を用いて記述した場合とそれ以外の言語で書いた場合の比較を行なう.

#### 5.1.1 評価方法について

### 5.2 アンケート

## 第6章 考察

### 6.1 評価結果

### 6.2 評価結果

## 第7章 結論と今後の展開

### 7.1 まとめ

### 7.2 今後の方針

*Short Title in English*

## 謝辞

謝辞を書く.

## 参考文献

- [1] 著者, 「タイトル」, 2003.

## 付録その1

付録その1(プログラムのソースリストなど)を必要があれば載せる

## 付録その2

付録その2(関連資料など)を必要があれば載せる

## 図 目 次

4.1	Cidner と openFrameworks . . . . .	7
4.2	Leap Motion . . . . .	7
4.3	開発ドキュメント . . . . .	7
4.4	Leap Motion のセットアップ . . . . .	9
4.5	インストール画面 . . . . .	9
4.6	Leap Motion のアイコン . . . . .	10
4.7	Leap Motion の設定画面 . . . . .	10
4.8	サークルジェスチャー . . . . .	11
4.9	スクリーンタップジェスチャー . . . . .	11
4.10	スワイプジェスチャー . . . . .	12
4.11	キータップジェスチャー . . . . .	12
4.12	Leap Motion のセンサ範囲 . . . . .	13



# 表 目 次

4.1 開発環境 . . . . .	6
--------------------	---