**Gesture Interpreter**

Andrew Ladlow

Contents

1. [Introduction 3](#_Toc434190603)

[1.1 Motivation 3](#_Toc434190604)

[1.2 Aims 3](#_Toc434190605)

[1.3 Report Overview 3](#_Toc434190606)

1. [Background 5](#_Toc434190607)

[2.1 Related Work 5](#_Toc434190608)

[2.2 Gesture Recognition 5](#_Toc434190609)

[2.3 Similar Applications 6](#_Toc434190610)

[2.3.1 LeapTrainer 6](#_Toc434190611)

[2.3.2 UNI 6](#_Toc434190612)

1. [Design 8](#_Toc434190613)

[3.1 A 8](#_Toc434190614)

1. [Implementation 9](#_Toc434190615)

[4.1 A 9](#_Toc434190616)

1. [The System in Operation 10](#_Toc434190617)

[5.1 A 10](#_Toc434190618)

1. [Testing and Evaluation 11](#_Toc434190619)

[6.1 A 11](#_Toc434190620)

1. [Conclusions 12](#_Toc434190621)

[7.1 Future Work 12](#_Toc434190622)

[References 12](#_Toc434190623)

**Chapter 1**

# Introduction

## 1.1 Motivation

Human-computer interaction has, for many years, been limited to mouse pointing and typing based devices. With the more recent development of smartphones we have seen numerous breakthroughs with touch based devices. These methods are all inherently unnatural when compared with real world interaction, however. In all of these cases users must interact through some kind of interface as opposed to naturally communicating via hand signals or body language.

Currently, there are several released devices which aid in monitoring, recording and displaying these more natural forms of communication. Devices such as the ‘Microsoft Kinect’, ‘ASUS Xtion Pro’ and ‘Apple Primesense Carmine’ all feature some degree of motion capture – the issue lies in the accuracy of recognition methods, especially with more precise or intricate gestures. The gesture interpreter aims to utilise one of these devices, the Leap Motion, in order to facilitate natural gesture communication – particularly gestures found in the British Sign Language (BSL).

According to Rudak, et al [1] the Leap Motion is capable of recognising movement with an overall average accuracy of 0.7mm, a degree that devices such as the Kinect were unable to reach in comparison. In addition, a recent V2 firmware update released for the Leap Motion improved tracking performance and enhanced the features of its API.

The proposed application will aid users who wish to learn or reinforce their current knowledge of the BSL. The low cost of the Leap Motion device means it could be purchased by users themselves or incorporated into current sign language courses in order to streamline the learning process.

## 1.2 Aims

The application should fulfil the following main objectives in order to achieve the goal described above:

* Allow for recording of custom, user defined gestures.
* Accurately recognise the formation of gestures and distinguish between them without generating false positives. (Degree of accuracy?)
* Transform the recognised gesture into speech output.

## 1.3 Report Overview

The structure of the report is as follows:

Chapter 1 discusses the problem and the proposed solution, including its aims. Chapter 2 analyses background research and related work as well as any closely related applications that currently exist. Chapter 3 … Chapter 4… Chapter 5… Chapter 6… Finally, chapter 7…

**Chapter 2**

# Background

## 2.1 Related Work

Sign language in education?

The use of the Leap Motion controller in order to detect sign language gestures was previously investigated by Araullo, et al [2], though in this case the Australian Sign Language (Auslan) was used rather than the BSL. The authors noted that although the controller showed potential, further development of the API was required – this was mainly due to inaccurate hand detection in certain scenarios e.g. pinching fingers together, interlocking hands or holding one hand above the other. These tests were carried out prior to the release of the V2 firmware update for the Leap Motion in 2014 which boasted better tracking performance. As such it’s likely that the current accuracy of the device has since improved.

## 2.2 Gesture Recognition

Probably the most important aspect of the system is its gesture recognition – as we’re dealing with data in real time, a suitable algorithm should calculate match results on the fly without causing delay or otherwise negatively affecting the user. Most approaches apply some form of machine learning on the Leap Motion frame data, or a specific subset of it. Some feasible examples are shown below.

The use of dynamic time warping (DTW) was explored by Russell, et al [3]. The appeal of DTW is that it isn’t reliant on the time taken or speed of each input in order to accurately compare them – this is particularly useful with gestures which are often performed at varying speeds. The authors demonstrate how DTW could be applied to 2D handwriting gestures and suggested that it could be extended for use with 3D data, i.e. gestures. They conclude that the DTW approach used is suitable for real-time comparison but it remains to be seen whether that is still the case when dealing with the increased complexity of gestures compared with just handwriting.

K-Nearest Neighbour (KNN) and Support Vector Machines (SVM) were proposed by Chuan, et al [7] as recognition algorithms for the American Sign Language using the Leap Motion. Tests were carried out using the 26 letters of the alphabet - results showed a recognition accuracy rate of 72.78% and 79.83% for the two methods, respectively. The authors mention some possible reasons for the low accuracy with both algorithms; compared with the BSL alphabet, the ASL is signed using only one hand – as a result some letter representations are very close to one another which led to misclassifications of the Leap Motion data. The use of BSL with these methods could show improved results as all of its gestures require two hands to perform, hence are more varied.

Hidden Markov Models / Markov Chains?

Neural networks / Support Vector Machines?

The $P recognizer, designed by Wobbrock, et al [4], is another example of a gesture recognition algorithm. As described in the paper, the $P is “a 2-D gesture recognizer designed for rapid prototyping of gesture-based user interfaces”. The $P aims to overcome the complex task of matching user gestures by instead treating them as groups or “clouds” of points and evaluating each one in turn. Even the simplest of gestures could be created in many different ways depending on the properties of its strokes e.g. start and end points, order or time, and direction. The use of a point cloud helps remove any ambiguity from the gesture which simplifies comparison and recognition. According to the paper, the algorithm requires only 70 lines of code to function and delivered over 99% accuracy in user-dependant testing.

## 2.3 Similar Applications

#### 2.3.1 LeapTrainer

‘LeapTrainer.js’, created by O’Leary [5], is a browser based gesture and pose learning and recognition framework for the Leap Motion. Developed in JavaScript, LeapTrainer allows users to create and store gestures then replay them at will. The software recognises a gesture using a template matching algorithm, based off the $P recognizer discussed above. The software supports both gestures and ‘poses’ – the difference being that a gesture is the movement of one or both hands at or above a specific velocity, whereas a pose is the stationary position of one or both hands over a period of time. This separation of gestures into motion / pose based is particularly useful as it allows both types of gestures to be recorded by the user, without them having to explicitly define a type or be limited to one or the other.

#### 2.3.2 UNI

A commercial application, UNI, is currently in development and is scheduled to be released in summer 2016 by MotionSavvy [6]. UNI bares similarities to the proposed application in that it utilises the Leap Motion in order to translate gestures into spoken text. However, the software is closed-source and based on a subscription model which limits further adaptation or extension. It is currently unknown which operating systems UNI will support and how well the software will achieve the goals described on the website.

**Chapter 3**

# Design

## 3.1 A

Comparison of explicit and implicit gesture recognition / segmentation e.g. pressing keyboard to start stop vs. detection based on hand speed or position etc.

**Chapter 4**

# Implementation

## 4.1 A

Mention of frame drop / slowdown when comparing every bone direction between current frame and deserialized frame from file, similarly when just comparing palm positions and finger tip positions but to a lesser degree - not viable for real time. Must use a subset of relevant frame data as opposed to storing whole frames etc.

**Chapter 5**

# The System in Operation

## 5.1 A

A

**Chapter 6**

# Testing and Evaluation

## 6.1 A

A

**Chapter 7**

# Conclusions

## 7.1 Future Work

Accuracy concerns

Limited to gestures not requiring body or face

Must be stationary to use - possible combination with oculus VR or other head mounted display

### References

[7] Chuan, C. H., Regina, E., & Guardino, C. (2014, December). American Sign Language Recognition Using Leap Motion Sensor. In Machine Learning and Applications (ICMLA), 2014 13th International Conference on (pp. 541-544). IEEE, 2014.

[6] MotionSavvy. (2015) UNI. Website, <http://www.motionsavvy.com/>. Last accessed 9 October 2015.

[5] O’Leary R. (2013) LeapTrainer.js. GitHub repository, <https://github.com/roboleary/LeapTrainer.js>. Last accessed 9 October 2015.

[2] Potter, L. E., Araullo, J., & Carter, L. (2013) The leap motion controller: a view on sign language. In Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration (pp. 175-178). ACM, 2013.

[4] Vatavu R.D., Anthony L., & Wobbrock J. O. (2012) Gestures as Point Clouds: A $P Recognizer for User Interface Prototypes. On line publication, University of Washington, <http://faculty.washington.edu/wobbrock/pubs/icmi-12.pdf>. Last accessed 13 October 2015.

[3] Vikram, S., Li, L., & Russell, S. (2013) Writing and sketching in the air, recognizing and controlling on the fly. In CHI'13 Extended Abstracts on Human Factors in Computing Systems (pp. 1179-1184). ACM, 2013.

[1] Weichert, F., Bachmann, D., Rudak, B., & Fisseler, D. (2013) Analysis of the Accuracy and Robustness of the Leap Motion Controller. On line publication, PubMed Central, <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3690061/pdf/sensors-13-06380.pdf>. Last accessed 16 October 2015.