

**Year 2 Project**

**Dactylology - Sign language translation**

**By: Henrietta Sohi(201186460), Dan Anderson(), Ben Hague(201146260), Yanqin Jiang(201298134)**

**Supervised by: Waleed Al-Nuaimy**

**Department of Electrical Engineering and Electronics**

**12 April 2018**

# Abstract

In the UK alone, 1 in 7 people have impaired hearing (1) and 1 million children and young people have long-term speech impediments (2). This is a sizable percentage of the population and yet little is done to aid their communication in everyday life. In this paper we explore the benefits and processes involved in the creation of a device designed to aid communication for people who are unable to communicate effectively with spoken word. A prototype device is devised using an Xbox Kinect sensor to explore the recognition of gestures in real time.

**Declaration**

**I confirm that I have read and understood the University’s definitions of plagiarism and collusion from the Code of Practice on Assessment. I confirm that I have neither committed plagiarism in the completion of this work nor have I colluded with any other party in the preparation and production of this work. The work presented here is my own and in my own words except where I have clearly indicated and acknowledged that I have quoted or used figures from published or unpublished sources (including the web). I understand the consequences of engaging in plagiarism and collusion as described in the Code of Practice on Assessment (Appendix L).**

Contents

[Abstract 2](#_Toc511300315)

[Chapter 1 6](#_Toc511300316)

[Introduction 6](#_Toc511300317)

[1.1 Background 6](#_Toc511300318)

[1.2 Objectives 7](#_Toc511300319)

[Chapter 2 8](#_Toc511300320)

[Materials and Processes 8](#_Toc511300321)

[2.1 Materials 8](#_Toc511300322)

[2.2 Processes 8](#_Toc511300323)

[2.3 Testing 9](#_Toc511300324)

[Chapter 3 11](#_Toc511300325)

[Results 11](#_Toc511300326)

[3.1 Error Analysis 11](#_Toc511300327)

[3.2 Referencing 12](#_Toc511300328)

[3.4 Writing Style and Formatting 13](#_Toc511300329)

[3.5 Multimedia 13](#_Toc511300330)

[Chapter 4 15](#_Toc511300331)

[Discussion and Conclusions 15](#_Toc511300332)

[References 16](#_Toc511300333)

[Appendices 17](#_Toc511300334)

[Appendix A 18](#_Toc511300335)

[Appendix B 18](#_Toc511300336)

[Appendix C 19](#_Toc511300337)

List of Figures

[Figure 1, Large Table of data for different Gestures in Matlab 9](file:///C:\Users\BenHa\OneDrive\Documents\Project\Template%20for%20Y2%20project%20report%20(Word)(Ben).docx#_Toc511301137)

List of Tables

[Table 1, Confusion Matrix of Subspace Discriminant Mddel 11](#_Toc511291225)

# Chapter 1

# Introduction

## Background

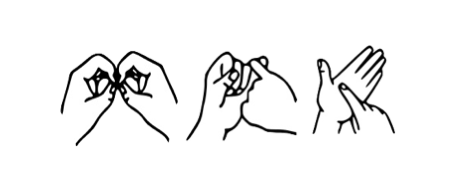
Since 2003 when the government formally recognised British Sign Language (BSL) as its own language it has become the common standard within the UK for individuals who are deaf or unable to talk. Currently, there are over 15,000 people who recognise BSL as their primary language. (3) The language makes use of space and involves movement of the hands, body face and head.

Figure 1, BSL spelled out in BSL (3)

The Kinect was first released in 2010 as an accessory to the popular Xbox 360. In its lifetime, 35 million units were sold (4). The Kinect identifies humanoid figures in real time and produces an output skeletal figure which can either be laid over the top of the existing image or used separately to analyse, record, or process the movements of the figure. The price point and availability of this technology is still 8 years later unmatched. This makes the Kinect an obvious platform for building prototypes on.

When the capabilities of the Kinect are analysed it becomes clear that it is a suitable candidate to act as a sensor for the BSL interpretation system.

We plan to create software which records BSL gestures performed by a user, interprets these and converts them into on screen text in multiple languages. In 2013 a similar project was undertaken by a collaboration of the Chinese Academy of Sciences and Microsoft Research Asia (5). A demonstration model was produced in 6-months which translated Chinese Sign Language into written and spoken signs. However, it took a team of 5 to teach the system how to recognise patterns for one word and by the close of the project, 300 out of a possible 4000 Chinese Sign Language words could be translated by the Kinect (6). In comparison this project has been allocated 6 weeks and shall translate 10 BSL gestures (as appropriate for the target audience). As a result, in terms of future development it should take much less time to update the library of recognised gestures and so be a more viable product which will evolve alongside spoken and sign language.

The general idea for the process behind the software is as follows:

1. The software is set to record…
2. A gesture is performed in front of the kinect camera multiple times
3. These repeated actions serve as the training data to be fed into a machine learning algorithm
4. The algorithm will try to cluster the actions and therefore predict what word the action is meant to represent.

## Objectives

Our objectives are as follows:

Chose a range of sign language gestures

Gather co-ordinates of sign language gestures when they are performed to form a library

Compare to our library of gestures and recognise

Present the text equivalent of the gestures on a monitor

The project should be completed by 9th March 2018

# Chapter 2

# Materials and Processes

## 2.1 Materials

Microsoft Kinect Sensor V2

This is needed to capture the sign language gestures performed, and skeletons track and their joint movements. It does so using an IR camera which projects a pattern of light and then calculates how long it takes rays which were incident on a surface (ie. a person) to return.

Matlab 9.3 R2017b

This is the version of Matlab to be installed on the computer/laptop to be used with the Kinect. Being the latest version, the library contains a greater number of precompiled routines which are relevant to the project and so reduce and simplify the amount of programming. Must have the Image Acquisition Toolbox Support Package for Kinect For Windows Sensor installed with all dependencies (7).

Xbox Kinect Adapter for Windows

Needed to provide power to the Kinect and connect the Kinect to the laptop to transfer data captured by a moving skeleton. This can be avoided if willing to modify the Kinect (7).

Computer /Laptop

Should run Microsoft Windows 10 64-bit in order for MATLAB to work.

## 2.2 Processes

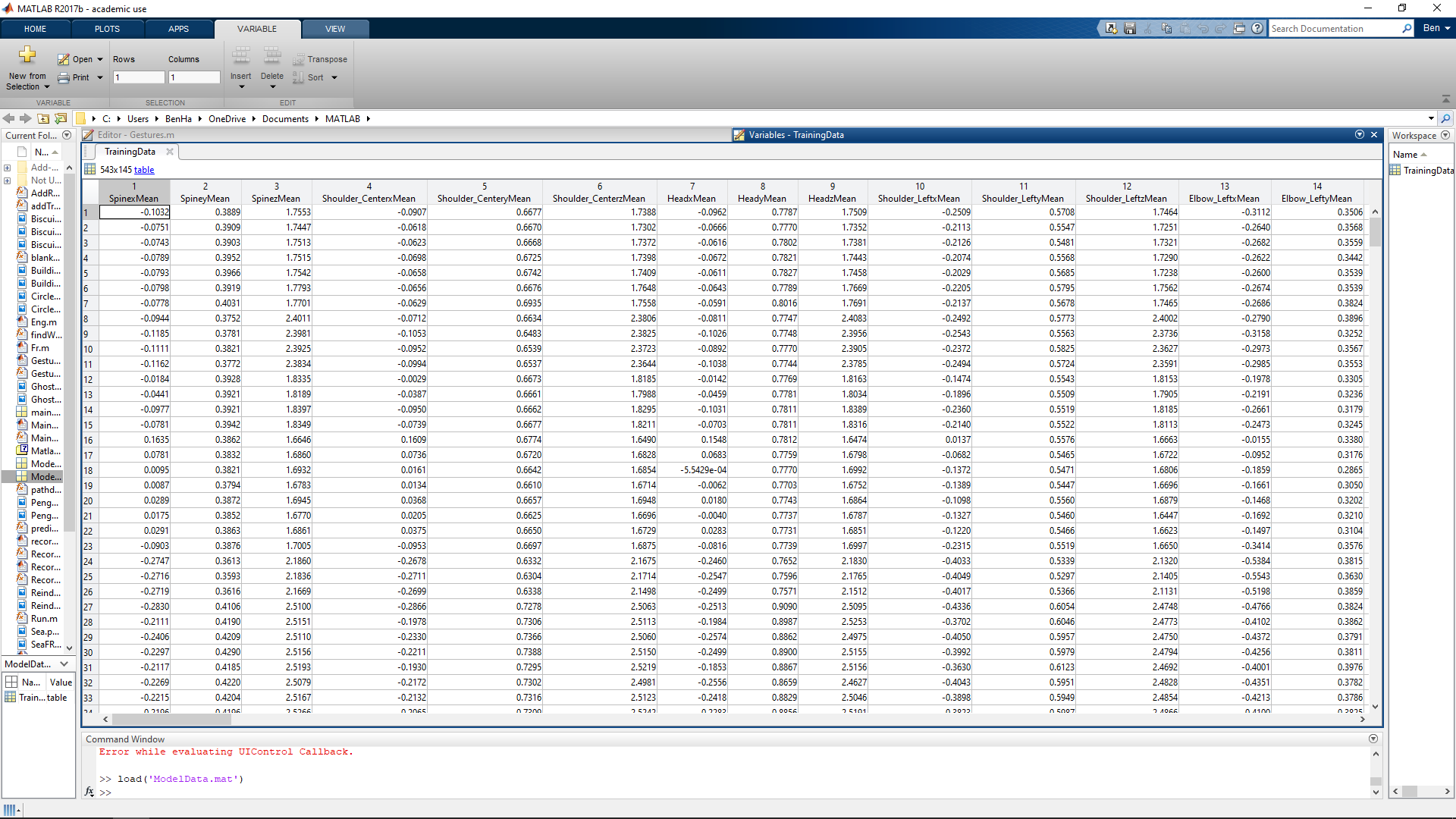
1. Setup
   1. Connect Kinect to Computer
   2. Install drivers as required
2. Capturing gestures
   1. Create the input device from the Kinect sensor, this is the device that the metadata is read from.
   2. Configure the trigger length to a predicted length of each gesture (30 frames, ~1 second)
   3. Trigger the capture device.
   4. Store the metadata
   5. Process the data, (see Data Processing)
3. Data Processing
   1. Create Placeholders for each the of the 16 joint coordinates (X,Y,Z) and the number of frames.
   2. Extract the data for the appropriate skeletal image data from the metadata variable and store in their relevant placeholders for each frame
   3. Place the data in a combined variable of the whole dataset
   4. Calculate the mean, standard deviation, and principal Component analysis for each joint over the 30 frames
   5. Combine the elements into an output table
   6. Store the data into a table with the gesture number
4. Training
   1. Record Each Gesture a number of times we used 30-70 repetitions per gesture (See Capturing Gestures)

Figure 2, Large Table of data for different Gestures in Matlab

* 1. Store the data into a large table
  2. Load the large data set into the classification learner selecting gesture as the response
  3. Test All models and export the one with the highest success rate (Subspace Discriminant Model)

1. Recognising and Identifying Gestures
   1. Capture a gesture and process the data
   2. Feed the data through the trained model
   3. Receive the output gesture

## 2.3 Testing

At the time of training the data had been tested through each possible model with a 50 fold validation from this the most successful model is chosen. Additional data was reserved and recorded from a 3rd party to ensure that the system worked reliably with the desired output. It was important to ensure that the data was collected from a variety of body positions, sitting standing and not fully facing the camera to ensure that the size of the sample data covered enough real world possibilities that it would predict the correct output the majority of the time.

# Chapter 3

# Results

There are a few different areas where we could analyse our results from these will be determined at various stages along the testing process. To begin with the results for 50 folds validation are used. Data recorded separately is then used to validate accuracy and to finish the live recorded data is used.

## 3.1 Validation Results

Table 1, Confusion Matrix of Subspace Discriminant Model

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Confusion Matrix of Subspace Discriminant Model | | | | | | | | | | | | |
| True Gesture | | 1 | 34 |  |  |  |  |  |  |  |  |  |
| 2 |  | 53 |  |  |  |  |  |  |  |  |
| 3 | 1 | 1 | 41 |  |  |  |  |  |  |  |
| 4 |  |  |  | 30 |  |  |  |  |  |  |
| 5 |  |  |  |  | 70 |  |  |  |  |  |
| 6 |  |  |  |  |  | 43 |  |  |  |  |
| 7 |  | 1 |  |  |  |  | 68 |  |  | 1 |
| 8 |  |  |  |  | 1 | 2 |  | 54 | 1 |  |
| 9 |  |  | 1 |  |  |  | 2 | 2 | 73 | 1 |
| 10 |  | 1 |  |  |  |  |  |  |  | 62 |
|  |  | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | Predicted Gesture | | | | | | | | | | | |

The confusion matrix shown in Table 1 shows the reliability of the models testing process. It also shows us which gestures were easy for the system to differentiate. Gesture 4, which was ghost, was a gesture which was not confused with any other gesture during this stage of testing whereas gesture 9, which was squirrel, was commonly misunderstood for other gestures. The greater number of red boxes present in each line and column show us which gestures should receive more training to allow them to be reliably differentiated from other gestures. This gives our model a statistical accuracy of 97.2%.

## 3.2 Blind Data Results

For this test a 3rd party unfamiliar with our system “spoke” a series of words which we recorded into the system and then ran through the model to test the output. This was not a conclusive test as the 3rd party chose which gestures they wished to test.

|  |  |
| --- | --- |
| Actual Gesture | Predicted Gesture |
| 2 | 2 |
| 2 | 2 |
| 2 | 2 |
| 2 | 2 |
| 3 | 3 |
| 3 | 3 |
| 3 | 3 |
| 3 | 3 |
| 5 | 5 |
| 5 | 5 |
| 5 | 5 |
| 5 | 5 |
| 9 | 3 |
| 9 | 9 |
| 9 | 9 |
| 9 | 9 |

Table 2, Blind 3rd party Gesture Predictions

Table 2 shows us that the system was able to differentiate gestures made by other people which were not in the training sample of data, this also showed us that the system was ready to receive real time data.

## 3.3 Live Data Results

The final system test was real time data, for this each member of our team “spoke” each word 2 times and the system was given the data in real time to decipher the gesture into English.

Table 3, Real Time Gesture Recognition Test Results

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Gesture | Ben | | Hettie | | Dan | | Yanqin | |
| 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 6 | 6 | 6 | 8 | 6 | 6 | 6 | 8 | 6 |
| 7 | 7 | 7 | 3 | 7 | 3 | 7 | 7 | 7 |
| 8 | 6 | 6 | 8 | 8 | 8 | 6 | 8 | 8 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |

The Results from this test shown in Table 3 demonstrated that the 97.2% as predicted in the first stage of testing was a theoretical value and didn’t equally show the strong and weaker gestures. in particular the system struggled to differentiate between gesture 6, Reindeer, and gesture 8 Space, This could be due to the similarities between the range of movements made in the gestures which are shown in Figure 3 and Figure 4.

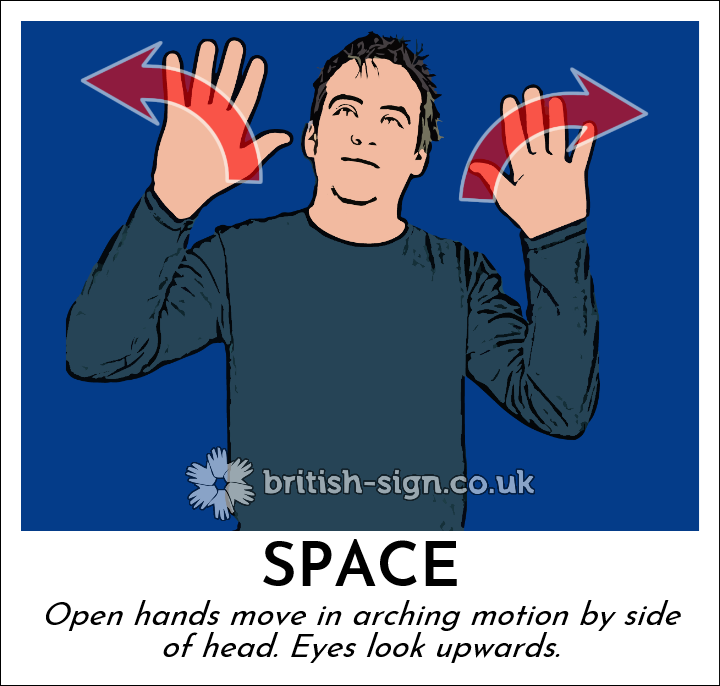


Figure 3, BSL Gesture for Space

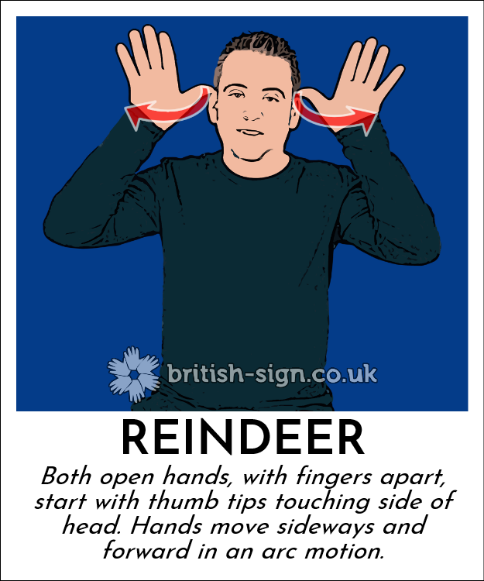


Figure 4, BSL Gesture for Reindeer

As the range in movement is similar the system requires more training to identify the slight differences between these different signs in a reliable way.

# Chapter 4

# Discussion and Conclusions

# References

1. **Centre, Disability Resource.** UK Statistics & Facts. *Disability Resource Centre.* [Online] [Cited: 11 April 2018.] https://www.disability.co.uk/sites/default/files/resources/UKStatistics%26Facts.pdf.

2. **Communication difficulties – Facts and Stats. *The Communication Trust.* [Online] [Cited: 11 April 2018.] https://www.thecommunicationtrust.org.uk/media/2612/communication\_difficulties\_-\_facts\_and\_stats.pdf.**

**3. Tmol42. British Sign Language. *Wikipedia.* [Online] [Cited: 11 April 2018.] https://en.wikipedia.org/wiki/British\_Sign\_Language.**

**4. Reisinger, Don. Microsoft Has Finally Killed the Kinect Xbox Sensor. *Fortune.* [Online] 25 10 2017. [Cited: 11 April 2108.] http://fortune.com/2017/10/25/microsoft-kinect-xbox-sensor/.**

**5. Wu, Goubin. Kinect Sign Language Translator. *Microsoft Research Blog.* [Online] 29 October 2013. [Cited: 12 April 2018.] https://www.microsoft.com/en-us/research/blog/kinect-sign-language-translator-part-1/.**

**6. Protalinski, Emil. Microsoft Research uses Kinect to translate between spoken and sign languages in real time. *The Next Web.* [Online] 30 October 2013. [Cited: 12 April 2018.] https://thenextweb.com/microsoft/2013/10/30/microsoft-research-uses-kinect-translate-spoken-sign-languages-real-time/.**

**7. Tatsh IT Services. Xbox One Kinect 2.0 USB3 Mod. *Tatsh IT Services.* [Online] 14 July 2014. [Cited: 2018 4 12.] http://www.tatsch.it/kinect-2-usb-3-modification/.**

# Appendices

## Appendix A

## Appendix B

**IMPORTANT NOTE: DO NOT PUT THE CODE AS A SCREENSHOT. THE CODE SHOULD BE PUT IN PLAIN TEXT IN THE REPORT.**

## Appendix C