Generating realistic, animated human gestures in order to model, analyse and recognize Irish Sign Language

Michèle Péporté (B.Sc.)

Submitted in fulfilment of the requirements for a

Master of Science

Degree

Dublin City University

School of Computing

Supervisor: Dr. Alistair Sutherland

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Science is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____

(Candidate) ID No.: _____

Date: _____

Abstract

The aim of this thesis is to generate a gesture recognition system which can recognize several signs of Irish Sign Language (ISL). This project is divided into three parts. The first part provides background information on ISL. An overview of the ISL structure is a prerequisite to identifying and understanding the difficulties encountered in the development of a recognition system.

The second part involves the generation of a data repository: synthetic and real-time video. Initially the synthetic data is created in a 3D animation package in order to simplify the creation of motion variations of the animated signer. The animation environment in our implementation allows for the generation of different versions of the same gesture with slight variations in the parameters of the motion. Secondly a database of ISL real-time video was created. This database contains 1400 different signs, including motion variation in each gesture.

The third part details step by step my novel classification system and the associated prototype recognition system. The classification system is constructed as a decision tree to identify each sign uniquely. The recognition system is based on only one component of the classification system and has been implemented as a Hidden Markov Model (HMM).

Acknowledgment

I would like to thank my supervisor Prof. Alistair Sutherland for guiding me through my work. I am very grateful to Sara Morrissey for sharing with me basic knowledge about the ISL but foremost for guiding and assisting me in correcting my Thesis.

I have to thank Siobhan O'Connor who is an ISL tutor. I had the chance to work with her for eight weeks and she introduced me in a deeper knowledge about the SL. Additionally, with her help I was able to record the real-time database.

I would like to thank the following people who have been kind enough to help me during the development of my project. Thanks to John and Victoria for their support and friendship, especially to Victoria for all her advice and help. Thanks to Martin for strengthen me in my work, distracting me if I need distraction and having faith in me.

I also want to thank my grandparents, Michel and Alice, their support facilitate me to fulfil my Masters in Ireland. In addition, thanks to my parents, Theo and Martine, and my two brothers, Pit and Ben, for all their support and encourage me to believe in myself.

Table of Contents

Table of Figures

Table of Graphs

List of Tables

Chapter 1 Introduction

The primary goal of a gesture recognition system is to create a simple means of interaction between human beings and computers. There are several ways to interact with computers such as using a mouse or a keyboard. At present using gestures for computer interaction is mainly used in the game domain (Kang et al., 2004). A lot of people are working in the area of gesture recognition to support human-computer interaction in many ways. For instance, the ability to interact via gestures could help a person in their interactions with computers, mobile devices and ultimately with their environment. It could also support the creation of gesture-based user interfaces for mobile phones, which will be easier to use than the current keypads. This should enable the use of virtual reality immersive applications. Gesture recognition systems will make a significant contribution to the availability of Sign Language (SL) recognition software on the mobile phone, which could greatly enhance the quality of life for the Deaf community. Unfortunately, working in this field is a very complex task because every human being moves differently.

This Chapter gives, in the first place, a general overview on recognition and gesture recognition focusing on Irish Sign Language. This includes the main contributions, the reason for and difficulties of developing a gesture recognition system for sign language and it provides a brief insight into possible solutions.

1.1 Sign Language

For the last few years, researchers are working in sign language using gesture recognition. Sign language is the native language of the Deaf communities (Ó'Baoill & Matthews, 2000). As in spoken language, every country has their own sign language, for instance Ireland has Irish Sign Language (ISL), Britain has British Sign Language (BSL) and France has French Sign Language (LSF). This means that the communication between an Irish Deaf person and French Deaf person can be as difficult as the communication between an Irish Hearing person and French Hearing person.

A computer-based sign language recognition and translation system could increase the participation of the Deaf communities in the world of hearing people. Up to the present day, the communication between Deaf and Hearing people has been very complicated and usually Deaf people have to communicate through an interpreter or in written form with hearing people. A translation system could simplify all this and facilitate the life of the Deaf in education, employment, culture and their representation in the hearing environment.

Sign languages are languages with their own structures. In spoken language, most of the communication is mainly formed through verbal communication: the spoken words and their sounds. In sign language, the communication proceeds through gestures and especially through non-manual features (NMF), explained in detail in section 3.2.3. Therefore sign language can be only partly understood by considering the gestures alone.

1.2 Recognition

Research in the field of recognition started with pattern recognition (Theodoris & Koutroumbas, 2006) which identifies objects in images by their features such as shape, size, colour or other characteristic details. These days, pattern recognition is used in almost all research areas. For instance in the medical environment, researchers use pattern recognition to analyse MRIs and can identify precise locations in the brain (Gemmar et al., 2008) or find unusual patterns. In the area of forensics, pattern recognition is used to identify finger and foot prints. Queens University in Belfast is running a project about shoeprint recognition using pattern recognition techniques.[1] Optical character recognition is a common field and is used to recognise, for example, handwriting or printed text (Impedovo et al., 1991).

Different researchers have developed various techniques in the area of recognition. Most of these techniques use common procedures. Firstly, the features of the object in the image are identified and then, these features are classified. Recognition is usually based on a set of patterns that have already been identified or classified and this set is used as training data for supervised learning. In the case of unsupervised learning, the system will not be given a defined set but will establish by itself the categories based on the characteristic details. Therefore pattern recognition belongs to the area of machine learning and very often to computer vision.

1.2.1 Face Recognition

Recently, a lot of recognition researchers have been working in the field of face recognition. Face recognition is used for different reasons such as interaction with the computer or identifying human beings. Sometimes face recognition is used as a

[1] www.ecit.qub.ac.uk

first step for another recognition area such as lip reading recognition (Cox, 2009). There has also been research done in the field of face aging (Park et al., 2008). Another purpose of face recognition can be found in security systems which use it in conjunction with other biometrics such as finger prints or iris recognition systems. In general, face recognition uses algorithms to analyse the relative position, size and shape of the eyes, nose, cheekbones or jaw. These features are then used to search for other images with matching patterns.

1.2.2 Facial Expression Recognition

In the last few years interest in the field of facial expression recognition has increased. Facial expression is a form of non-verbal communication used to express emotions. Facial expression is a very important criterion in sign language. The gestures alone express only 30% of the language. The other 70% is expressed through body language and especially through facial expression.[2] Unfortunately most approaches in sign language recognition do not include facial expression recognition and focus only on gesture recognition.

1.2.3 Gesture Recognition

Gesture recognition is used a lot to recognise human actions and human behaviour for different reasons. For example, some researchers working on gesture recognition want to improve security. Approaches have been developed to facilitate the security of an area with security cameras which are able to track moving objects and classify them into categories such as cars, animals or human beings.[3] The most common field for gesture recognition research would be the interaction between human beings and

[2] Anecdotal evidence in consultation with sign language tutors.
[3] This is a current project run by Queens University in Belfast - www.ecit.qub.ac.uk

computers. Basically, gesture recognition is used for detection and identification of human action and behaviour. A few projects use this information to reproduce natural human actions and behaviour in the animation area.[4] In gesture recognition, animated reproduction of human actions can help analysing various situations. Therefore I focused attention on system analysing with both synthetic and real-time video data. Initially some of the training data used for my developed system, is created in a 3D animation package. This simplifies the generation of signs with variation in parameters, which model in the environment such as camera positions, light angles and background textures. This is usually supplemented with large volumes of real-time video data as the algorithms are refined.

Most of the techniques in gesture recognition, explained in Chapter 2, are common to the procedures of the other recognition fields. Primarily, the features of the object in the image have to be identified. Identifying features in gesture recognition includes researching new methods in skin and hand detection. The support of skin and hand detection will facilitate advanced motion tracking on a variety of human subjects under various lighting conditions. Mathematical methods would be another way of identifying motion features. The second part involves classification of objects into classes based on their features. Therefore I developed a classification system, explained in section 5.1, which can identify each sign in my database, described in section 4.1, uniquely.

[4] Miralab works on project to animate natural human actions and behaviour - www.miralab.unige.ch

1.3 Sign Language Recognition

Researchers all around the world are working on approaches to sign language recognition. This research makes a significant contribution to the availability of sign language recognition, which could greatly enhance the quality of life for the Deaf community. Additionally, the ability to interact via gestures in the more general sense will aid people in their interactions with computers, mobile devices and ultimately with their environment. A primary design goal is to ensure the program works on a standard personal computer with a normal web cam or on a mobile device and that it can be adapted to a new user after a short training session. The prototype, described in Chapter 5, currently works only on the created database.

In the last few years, researchers in Ireland have been working on sign language but nearly all the work has been done in the linguistic area.[5] Sign language recognition in computer vision is a very difficult task because there are many variations in the way human beings perform motions. On the other hand, there are no clear boundaries between individual signs in a sentence and individual signs can appear differently using the same meaning. Additionally, the recent sign language recognition systems (Kelly, 2008) can only recognize finger spelling corresponding to the letters of the spoken language alphabet.

Another important point is that most developed approaches for sign language recognition focus mainly on gesture recognition. By excluding NMFs which are explained in section 3.2.3, a sign is characterised by phonemes: hand shape, movements, palm orientation, the location of the hands in relation to the body and sometimes non-manual features. These three characteristics cannot identify every

[5] Center for Deaf Studies - http://www.tcd.ie/slscs/cds/

sign because several signs have multiple meaning and can only be comprehended by combining the gesture with NMFs.

The development of a full sign language recognition system involves some difficulties:

- Hand shape recognition

  Hand shape is a very important component of a sign. Some signs use the same motion and hand position in relation to the body, for this reason these signs can only be differentiated by the hand shape. Hand shape recognition can be very difficult because the orientation of the hand in space might not show the hand shape clearly or might cause it to look similar to another hand shape.

- Motion tracking

  There are different ways to track motions: using skin and hand detections or mathematical methods to represent the movements graphically. Motion detection can be made difficult because the database includes 2D videos and the signs are composed of 3D movements. This problem is explained and adressed in section 5.2.2.

- Variation in different users

  Two conditions can affect a system used by various users. First, human beings have different body shapes. At the current stage of the recognition system, this is not yet an issue because the database used, explained in Chapter 4, includes only one signer. Second, people execute movements

differently. Small changes in a movement can have big effects on the recognition, this is demonstrated in section 5.2.2.

- Variation in the signs

  Due to the evolution of Irish Sign Language, some signs exist in a male and a female version (Ó'Baoill & Matthews, 2000). Both signs are for the most-part quite dissimilar. To solve this problem, every version of each sign has to be included in the system on which the system is trained.

- Speed variations of the signers

  Every signer has their own speed of signing. Beginners sign very slowly and the signs are made clearly, whereas advanced signers sign very fast and use shortened signs. The speed in the sign influences the representation of the sign in the space.

- Environmental influence

  Changes in the environment can have great impact on the recognition of sign language. One problem involves different light conditions, which can reflect skin colour in various colour ranges. Examples of this impact are shown in section 5.2.2. Another problem which has to be addressed is camera position. The changes in camera position have a big impact on the representation of the movement. This is shown in section 5.2.2. In this thesis, all the real-time videos which were created have a black background and the signer is wearing a black shirt.

- One sign, multiple meanings

  Some words in spoken languages have multiple meanings and can be understood in the context of the full sentence. Similar to spoken language, multiple meanings of words can be found in sign language in the same way except that in sign language the meaning can be found out through NMFs as well as the context. This linguistic problem has not been addressed in this thesis.

- Facial expression and body language

  Chapter 3 will provide background knowledge of the sign languages. This will explain the importance of the NMFs in Irish Sign Language. However, the developed approaches for sign language recognition only focus on gesture recognition.

- Position of the hands in relation to the signers body

  One requirement for sign recognition is to detect the position of the hands in relation to the signer's body. In a sentence, the signer moves the hand from the end position of the last sign straight to the start position of next sign. In my prototype, I focus on single signs with a defined start and end position.

- Different position of the signer in the image

  Signers might not sit or stand every time at the exact same position in front of the camera which can influence the system. This problem can be solved using signing space detection. Everything which is not included in the signing space will not be considered in the recognition. The signing space in sign

language is explained in detail in section 3.2.2 and the implementation is described in section 5.2.1.

- No structure for translation

  Sign language does not have a defined structure but Ó'Baoill & Matthews (2000) describe several structures used in sign languages. For example, some words which would be used in spoken language are left out in sign language. More about this can be found in Chapter 3.

1.4 Main Contributions

The main contributions in this research are:

1. The creation of a real-time video database.

   Creating my own database was necessary because the only databases of Irish Sign Language (ISL) which exist until now, include only one sample of each sign. Therefore I created a database which contains 1400 different signs and each sign is recorded about 8 times. The different samples of each sign comprise variations in movements which can appear during signing. The variations in signing are used to train, test and analyse the recognition system. My created database is the only existing database of ISL which can be used for analysing ISL for automatic recognition approaches. The creation of the database is described in section 4.1

2. The generation of synthetic videos

Creating a database can be very time consuming. Additionally, a signer will always have slight variations in each sign. Therefore, I generated synthetic signs which are used to analyse different situations. The animated signs include several variations for example in the movement and environment. After implementing one sign, the generation of additional videos containing small changes is very efficient and precise because the same script can be used with minor variations. Additionally, the movements in 3D space are represented by a very smooth shape compared to real-time videos. This allows me to analyse the signs without taking noise into consideration. A detailed description and explanation of generating animated signs is provided in section 4.2.

3. The development of a classification system

Creating a classification system for ISL allows me to identify each sign uniquely. Currently, there exist various notation systems, explained in section 3.3, but those are used to write the sign language in symbols. I developed a few rules based on phonemes which enable me to identify every sign of my generated real-time video database. This classification system is built up as a decision tree. This system is explained in detail in section 5.1. The later developed recognition system is based on this classification system. However, the recognition system focuses only on one part of the classification system. The sign language consists of manual features (hand-shape, location and motion) and non-manual features (NMF). In all the

experimental work in this thesis I looked only at manual features. Sign languages have complex grammar which affects the way signs appear in a sentence. But in all the experimental work in this thesis I looked only at single decontextualised signs.

4. Modelling the signs of the database using Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique used to reduce high dimensional data into low dimensional data. I apply PCA to sequences of images taken from the real-time video database which reduces these images to three dimensional data. This allows me to illustrate a sequence of images in a graph. These graphs represent the changes from one image to the next one. Therefore I defined a few conditions, the signer had to wear a black shirt and the background had to be black. This means that the environment in each video is the same and the only changes appear is the movement of the signer. Knowing this, PCA is used to represent the movements of the signers in the graph. PCA is explained in section 2.2 and the representations of the images in a graph are described in section 5.2.2.

5. The development of a recognition system of the sign in the PCA space using Hidden Markov Models

Finally, I was able to create a recognition model to identify signs through their movement. I used decontextualised signs of the database: synthetic for analysing, and real-time video for testing and training. This recognition

system focuses only on the movement of the signer's hand. The representation of the movement in graphs using PCA as mentioned before, are used to create the observed symbols which are used as input for the recognition model. For each sign, an individual small Hidden Markov Model (HMM) was created and all individual HMMs are connected to each other. The recognition system can identify 12 different signs with a recognition rate of 97.91%. The recognition system is explained in section 5.2.

1.5 Outline of the Thesis

This thesis is divided into five main parts. Chapter Two gives a literature review on previous and current research done in the domain of gesture recognition. Chapter Three outlines background knowledge of Irish Sign Language including its development and structure. Chapter Four describes the generation of real-time videos and animation videos which both was created and used in this approach. The complete developed systems will be explained in Chapter Five. First a classification system was built up to identify each sign uniquely. Thereafter, a gesture recognition system was implemented. This system focuses on the movements of the ISL signs and uses the created data. The synthetic data is used for system analysis on various influences on the signer and their environment. The real-video are used to train and test the recognition system. Finally, Chapter Six includes the conclusion and future work.

Chapter 2 Literature Review

In the last few years, a lot of research has been done in the field of gesture recognition and it has become very popular in various areas such as tracking and analysing human motions or sign language recognition. In general, gesture recognition has received much attention in human computer interaction because it leads to a simpler interaction between human being and computer.

The research in sign language recognition comprises the understanding and recognition of human hand motions and facial expressions and also includes body language. However, the sign language recognition approaches developed so far have not included facial expression or body language recognition but focused only on motion recognition using computer vision principles such as Principal Component Analysis (Shlens, 2005) or Markov Models (Bunke & Caelli, 2001).

However, to complete a sign language recognition system some important areas have to be considered;

- Variation in motion for motion tracking
- Environmental interference for hand detection
- Linguistic features such as structure and grammar
- Size of vocabulary
- Recognition speed
- Variation between different users

Chapter 1 outlined the main contributions and some difficulties which have to be dealt with while developing a gesture recognition system for ISL. This chapter describes the most common techniques used in recognition with the main focus on developing a prototype for a sign language recognition system. Additionally, this chapter will point out some of the different techniques I applied in this thesis.

2.1 Feature Extraction in Recognition

In the area of computing, pattern recognition has become very important over the last few years and is used in several domains such as pattern, speech and face recognition as well as facial expression and gesture recognition. Each area uses different methods. However, the process is every time the same: identify new data through features and classify the object.

There is no general definition of the term *feature*, but it can be described as the interesting, characteristic details of an image. Feature detection is an image processing operation which normally processes every pixel to find the specific detail in the image. Feature detection can be grouped into the following sections (Sauer, 1997) (Whelan & Molly, 2001):

1. Edge detection (one-dimensional structure)

   The edges of the objects in an image are detected; this is mostly used for shape recognition,

2. Corners / interest points (two-dimensional structure)

This method performs firstly edge detection and then analyses the edges to find big changes in directions (corners),

3. Blobs / region of interest

   Blob detection describes image structure in terms of regions, as opposed to corners, which are more point-like. Blob detection can detect areas in an image which are too smooth to be detected by a corner detector. This method is often used in the field of gesture recognition,

4. Ridges (one-dimensional curve)

   Ridge detection is calculated over grey-level images. This method is often used for road extraction or extracting blood vessels in medical images.

Wu *et al.* (1999) reviewed different techniques for 2D feature detection for gesture recognition. They found that it is possible to recognize a static hand position by extracting features such as fingertips, hand contour, colour and textures. However, these features can be influenced by environmental interference, lighting changes or position and orientation of the hand. Gestures can be recognized using features such as colour tracking for skin detection, motion tracking, template matching or blob tracking. They describe simple motion tracking computed from an image sequence which in some cases is enough for a simple gesture recognition system. The ideas of skin detection and creating a blob around the objects of interest are excellent. Therefore, object detection and the method of drawing blob around the region of interest, were used for signing space detection, described in section 5.2.1. However, their research is based on 2D feature detection and my recognition system needs 3D

feature detection. Therefore I decided that another method has to be adapted for recognition of the movement in the sign.

Additionally to feature detection, there exists feature selection which is a process, used in machine learning. Using feature selection means the data includes too many features to be processed and for that reason the most important features have to be selected. It is necessary because most time it is computationally impossible to use all features, especially working with images. Using images the dimension of the data increases very fast. To reduce the number of features a common technique called Principal Component Analysis (PCA) (Shlens, 2005) can be used and is explained in more detail in section 2.2. The features, which are selected from the data to recognize the object, are used in a learning algorithm such as neural networks, Markov models, or Bayesian networks.

2.2 Principal Component Analysis

Principal Component Analysis (PCA) (Shlens, 2005) is a mathematical orthogonal linear transformation which transforms data into a new coordinate system. Therefore it is a feature extraction and data representation technique. In this case PCA is used to reduce high dimensional image and represent the changes in the image into a new 3D coordinate system called an eigenspace. For example, an image with the size of 20x30 pixels has a dimensionality of 600 pixels. More often, the image dimensionality is higher than 1000 pixels. For this reason it is important to reduce the size of the image in advance by extracting the region of interest. PCA can reduce the dimensionality of 600 to 3 and then is able to represent each image in the calculated eigenspace. PCA is mostly used on image sequences to represent the changes from

one picture to the next. An advantage of PCA is that it is very sensitive to any changes in an image sequence, so it cannot represent two similar images to the same point in the eigenspace. Considering a number of instances of the same gesture, each instance can be represented by a different sequence of points in the same eigenspace. This allows me to model and analyse variations of the same gesture and its environment influence.

The first step in PCA is finding the covariance matrix of the set of images. The covariance matrix is calculated from a set of images and compares each pixel to the every other pixel through the image sequence. The method generates a new set of variables called "principal components". Each principal component is a linear combination of the original variables. All the principal components are orthogonal to each other and form the orthogonal basis for the eigenspace called eigenvectors. Each gesture is projected into its own eigenspace.

PCA can be applied in sign language recognition in two different ways. As mentioned before, each image is represented as a point in space; therefore a sequence of images will be shown as a sequence of points in the eigenspace. Coogan (2007) used this method focusing on static hand shape recognition, the sequence of points would form a cluster which could be used to identify the hand shapes. I do not focus on the hand shape but on the movement of the hands. In general, the distance between each point expresses how similar the images are. Therefore the representation of hand movements forms a trajectory in the space instead of a cluster. I use those trajectories to identify the motions of a sign.

Yang *et al*. (2004) developed a new technique for image feature extraction and representation using a two-dimensional principal component analysis (2DPCA). A 2DPCA is based on 2D matrices in comparison to a basic PCA which is based on 1D

vector. An image covariance matrix can be constructed straight away using the original image matrices and without a previous transformation to a vector. Therefore the size of the image using a 2DPCA is much smaller. In theory, 2DPCA is easier to evaluate and needs less time to determine the corresponding eigenvector. According to their results and conclusion, a 2D PCA is computationally more efficient than a basic PCA and it can improve the speed of feature extraction. However, in their results 2DPCA was not as sufficient as the basic PCA because 2DPCA requires more coefficients for image representation than PCA. According to their results, I decided that the simple PCA is efficient enough to represent the movements in a 3D space.

Luukka (2009) works with data to determine the state of patients in a post-operative recovery area and if they should be sent to the next area. This data is reduced to a lower dimensionality using PCA so they can get rid of some possible noise, after that they can defuzzify[6] the data and finally use a classifier to get the decision of what should be done with the patient next. He has presented a method for a robust PCA analysis. The method has been tested on real-time and synthetic images. In their conclusion they say, that their work has to been extended in several ways before they can apply it on patients. However, their method confirmed my previous decision to use PCA on real-time and synthetic videos for the representation of movement in a 3D environment.

2.3 Gesture Recognition

The goal of gesture recognition is to interpret human gestures using computer applications. Currently, this field focuses on facial emotions and hand gestures.

[6] Defuzzification is the process of producing a quantifiable result in fuzzy logic.

Gesture recognition can be applied in several areas such as sign language or identifying pointing gestures. Huang *et al.* (1995) gives an overview of different techniques used in hand gesture interfaces such as glove–based, vision–based or techniques that use drawing gestures with a stylus or computer mouse. They conclude that the future of gesture recognition relies on computer vision techniques instead of using glove- and stylus- based approaches. In their opinion the first and most straight forward method is using one or a few simple video cameras to gain visual information about the person and its environment.

Gesture recognition involves identifying gestures through their movements. Quek (1994) developed a set of rules for gesture segmentation based on the gesture pattern. He does not describe the movements themselves but the start and finish of gesture. For instance, the first rule would say that the gestures have to start from a certain resting position. The next rule would define that slow motions between resting position are not gestures and another rule specifies hand gestures to be signed in a certain space, which in my case would be the signing space. The idea of using rules is excellent, but the rules have to be defined specifically for each problem. In my case, I recognize only decontextualised signs. Therefore I could use the idea of rules only to define start and end position. The following section gives an overview of different techniques developed over the last few years.

2.3.1 Motion Tracking / Motion Capturing using Animated Data

Motion capturing (Hipp & Gumhold, 2002) or motion tracking describes the process of detecting movements and converting them into digital data. In the domain of film making, it refers to recording actions of human beings and using this information to

animate digital characters with human movement in a three dimensional animation environment. In the field of gesture recognition, the animation environment is used to create more data. This data can then be easily modified and used for testing and analysis. Therefore basic background knowledge about human movement is needed.

Using animated video involves different problems. Every recognition system needs a lot of data for training and testing. This problem can be solved in using animated videos. Matt (2002) created a collision handler which calculates the possibility of a collision only for those body parts where a collision could be expected. This collision handler works by applying rules, which describe the direction and the amount of movement and include the other body part with which the applied body part is colliding. A collision hander would be useful in creating animated videos. At this stage, I only used one character and implemented each sign individually. For this reason I decided not to use a collision handler.

Human hand motion modelling is a very complex task because the hand consists of many small connected parts. At the same time, the hand motion is limited which increases the complexity. Vogel *et al*. (1998) developed a new technique to identify moving objects estimated by their shape and motion. Firstly they developed a human body-part identification algorithm which can identify real human body-parts and the three dimensional shape of an animated character. The second part consists of extracting the three dimensional shape of the arm to be able to track the position and orientation of the body part. They use those two steps to track the motion of an arm to be able to recognize American Sign Language (ASL) gestures. This approach identifies the objects from a predefined and stored set of motions that shows the object structure. According to their results, they are very satisfied with their approach, even if they only used a part of the motion analysis so far. I agree that a

gesture recognition system improves using 3D motions instead of 2D motions. However, I chose not using their techniques to capture the motion but I use PCA which will be explained in section 2.2.

Awad (2007) developed a motion tracking approach with the focus on sign language recognition. He uses skin segmentation to detect the hands and the face and get colour information. The colour information is combined with information about the position and motion of the user's hands and face. Every skin object will be surrounded by a blob or search window. He implemented a set of rules as an algorithm to detect the motions after identifying the hands and face. He has shown that this combination of colour, motion and position information can provide accurate segmentation of the hands and face in sign language recognition. His approach works well for simple gestures in 2D. In my case I try to identify each sign in a 3D space because ISL is more complex and to identify all signs the smallest variations in their motion has to be detected which is only possible in a 3D space.

2.3.2 Hand and Motion Detection

In general, detection consists of the collection of information needed for recognition. One problem in gesture recognition is finding a method to capture motions which can be detected using vision techniques, such as hand and skin detection, and described mathematically using, for instance, PCA. The following part gives an overview of different techniques used in recent years to detect, capture and track hand motions.

2.3.2.1 Hand and Skin Detection

One possibility for motion capturing is hand detection. There are several techniques to detect the hand such as using gloves, sensors or skin detection. One common method for hand detection is using gloves. Two different types of gloves are used: colour detection or wired data gloves. Grobel *et al.* (1997) and Kelly (2008) used coloured gloves to detect SL hand shapes in videos. The disadvantage is that the user will not be able to use the system without the coloured gloves, and in most cases it has to be a specific colour on which the system is trained. Kadous (1996) used Power gloves to detect the hand. These wired data gloves usually deliver very good and precise data. In this case a negative side-effect would be the reduction of the natural movement because the user has to be connected to the computer through wires all the time. Other researchers use sensors attached to the hand or gloves. The same circumstance as in the previous case will arise. The wires will affect the natural movements and influence the system (Iwai et al.,1996).

Skin colour detection is a well-known and very difficult procedure. To develop an efficient working skin detection system depends on the adjustment and specification of the skin colour which comprises a wide range of colour for one person. Each person has a different range of skin tones so therefore the system has to be adjusted for every user. Another important aspect is the influence of light changes. The skin colour reflects differently using different lights so the colour range increases. Zarit *et al*. (1999) have modelled a comparison of five colour spaces and two non-parametric skin modelling methods using a lookup table and Bayes skin probability map. The two main problems in skin colour detection, they describe, are to find a colour range and allocate them to the skin colour. The skin colour space can be RGB, normalized RGB, HSV, YCrCb or YUV. These are different representations of colour space. The

skin detection can classify pixels by two different methods. Four years later, Vethnevets *et al*. (2003) wrote a survey on skin colour detection techniques developed up to that time. These techniques can be used for face detection and localization of human body parts and based on the same skin colour space as Zarit *et al.* (1999) mentioned. Both conclude that skin detection is a very complex and extensive procedure which has to be adjusted for each person individually. A disadvantage is that a colour range has to be generated for each person individually.

Suk *et al*. (2008) use a technique developed by Argyros *et al*. (2004).They detect the hand and face using skin colour detection and draw a blob around each object. Argyros *et al.* uses edge detection instead of drawing a blob. The disadvantage of their hand detection is that it often fails to track the hand when the hand makes unexpected non-linear motions.

Coogan (2007) used a technique developed by Awad (2007). The hands and face are initially detected by locating skin pixels in the image. Skin pixels are defined in a RGB colour space. It is essential that there are only three skin objects in the image with the head being the uppermost and largest one, and the hands will be lower and on the left and right of the head. Both used the condition of black background. Their skin detection is satisfying for their use, but for example edge detection could not be added because some skin part which lies in shadow, the pixel will not be identified as skin pixels. This results that the skin pixel is not in the colour range anymore.

After the discussed skin detection methods, I conclude that at this stage of my recognition system, using skin detection would be too time dependent; therefore I used binary converted images to detect the hands, which is described in section 5.2.1

2.3.3 Networks

This section gives a general understanding of networks. Networks are used to model complex artificial learning algorithms which have the ability to find patterns in data. The main idea behind networks is to have a network in which the core is unknown. Networks can be trained through known data and it adjusts itself artificially based on that data to be able to match unknown data to the known data. Different types of networks exist, the main ones used are briefly explained below.

2.3.3.1 Neural Networks

A neural network is an artificial representation of the human brain and tries to simulate the human learning process. A neural network is made up of connected elements called neurons. The output of a neural network depends on the interaction between the individual neurons within the network. Neural networks are able to learn through different artificial learning techniques, therefore they cannot be programmed to perform a specific task. This can be a disadvantage because the network finds out how to solve the problem itself and can operate in an unpredictable way.

2.3.3.2 Bayesian Networks

Bayesian networks (Korb & Nicholson, 2003) are probabilistic graphical models which can represent a set of variables and their probabilistic dependencies. Nodes represent variables and arcs the connections between the variables. A Dynamic Bayesian Network (DBN) (Suk et al., 2008) is a Bayesian network that can model sequences of variables. These sequences are often time-series or sequences of symbols. The Hidden Markov Model (HMM) (Bunke & Caelli, 2001) and the Kalman filter can be considered as the simplest DBN. Primarily Bayesian networks were used in speech recognition, more recently these networks are commonly used in

gesture recognition (Korb & Nicholson, 2003). Figure 1 gives an example of a simple DBN. This network shows how changes are made over time and the current state dependency of the previous states. The *State* represents the hidden states of the model and the *O* represents the observed states. Every hidden state will be influenced by their observed state at the moment of the process. In this case each hidden state has two observed states which together will have an affect on the hidden state. Additionally, the hidden state two depends on the hidden state one. The dependency is calculated by probabilities and the result of the DBN is a likelihood value which defines how likely the tested data fits to the trained data.

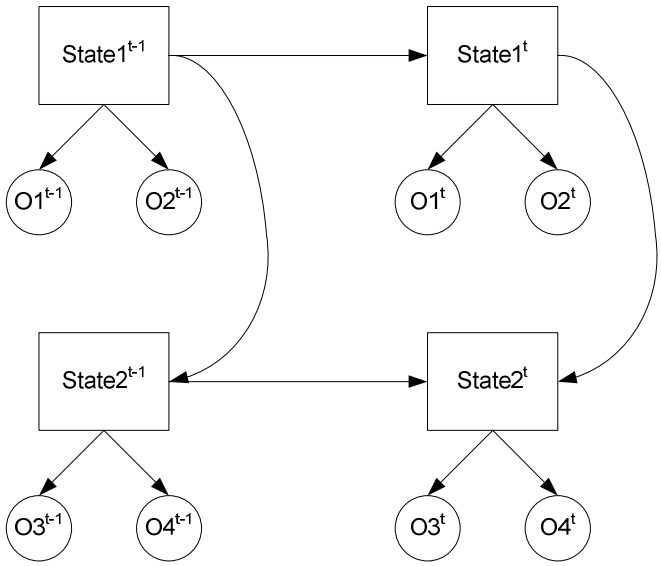


Figure 1: Dynamic Bayesian Network. ***The network model contains four hidden states (State) and their corresponding observed states O, alongside their associated dependencies.***

Suk *et al*. (2008) describe a DBN based on a hand gesture recognition method with the main focus being the ability to control a media player or PowerPoint. Firstly, their proposed approach tracks the hands while creating a blob around each, and models the hand motion using Gaussian distribution (Argyros & Lourakis, 2004)

where the mean represents the location of the hand centroid. Their proposed DBN has three hidden states and five observation states. Two hidden variables model the motion, one for the left hand and one for the right hand, and each is associated with two observation variable one with the features of the hand motion and the other with the position of the hand relative to the face. The third hidden state models the relation between the two hands. Their recognition system was tested with a recognition rate of 99.59%. The test data are real-time videos. They captured ten different videos and of each video they have seven samples, which give them a total of 490 videos sequences for training and testing their system. Figure 2 shows the states of the proposed DBN model. State 1 represents the relative position of the two hands in front of the body. State 2 and State 3 represent the change of the motions of each hand, therefore each state represent one hand: left and right hand.

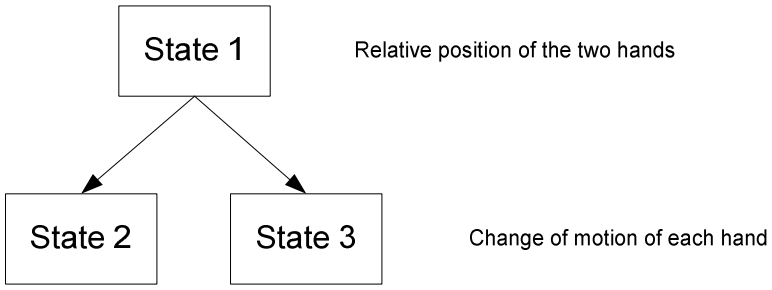


Figure 2: Proposed model by Suk. ***The network model contains three hidden states (State): State1 model the relation between the two hands and State 2 and 3 model the motions of each hand.***

2.3.3.3 Hidden Markov Model

Hidden Markov Models (HMM) are simple Bayesian networks, a kind of statistical model, and they are used to find patterns which appear over a period of time. The model has to be a Markov process with adjustable parameters which model the transitions from state to state depending on all the previous states. There are two

different types of Markov model, the deterministic model and the non-deterministic model. The deterministic model declares that the state of the model depends deterministically only on the previous states of the model. Whereas the non-deterministic model expects that the choice has to be made probabilistically. The probabilities do not change in time.

In most cases, the pattern which has to be found, is not described sufficiently by a Markov process, therefore the HMM is used which has two sets of states: the observable states and the hidden states. The challenge is to determine the hidden variables from the observation variables. Figure 3 gives a simple example of a HMM. This example represents a model to find out the weather conditions from observed states of seaweed (Boyle). The connections between the observable states and the hidden states represent the probability of an observed state given a certain hidden state. All the hidden states are connected to each other.

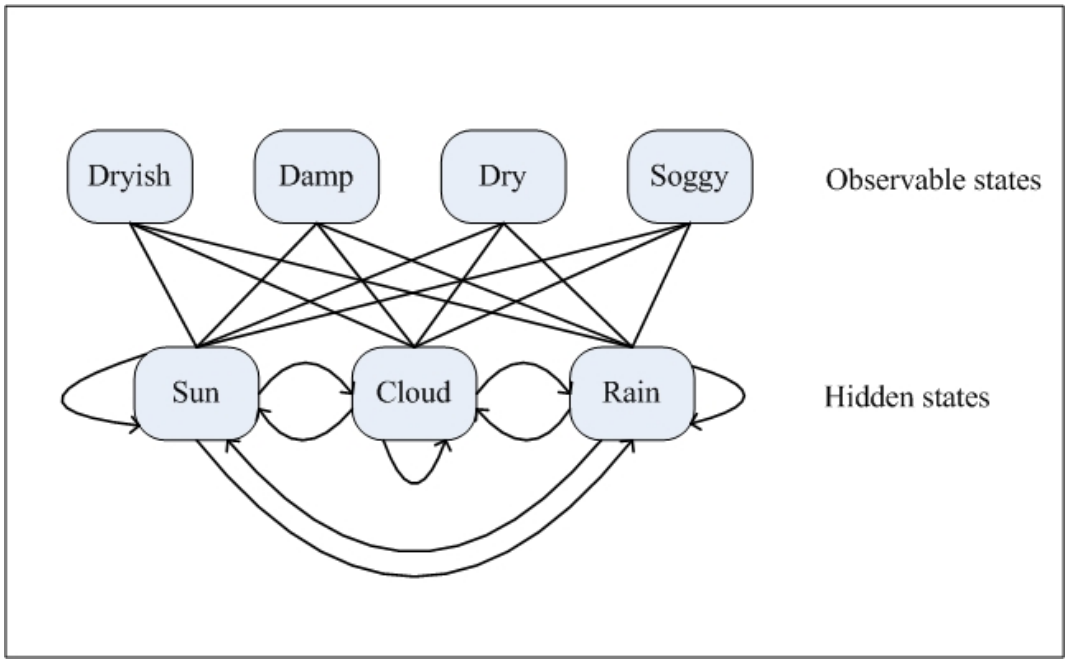


Figure 3: Example of a simple Hidden Markov Model. ***This network model shows four observed states and three hidden state. This model is used to find out the weather by observing the state of seaweed***

HMMs can be trained using the Baum-Welch algorithm (Bunke & Caelli, 2001). The Baum-Welch algorithm is commonly used to find the unknown parameters of an HMM. This process can compute the maximum likelihood estimation and posterior mode estimation for the parameter of an HMM. The Viterbi algorithm (Vogler & Metaxas, 1998) can be used to compute the most likely sequence of hidden states for a given set of data. And the Forward algorithm (Boyle) can compute the likelihood of a set of data for a given HMM.

Huang *et al*. (1995) give a good overview of the current techniques used for gesture recognition up to 1995. Most approaches are based on glove-based devices to recognise the hand. The popularity of using Bayesian networks or Markov models was increasing during that time. Within a few years, a lot of research in the field of gesture recognition was made using HMMs. Grobel *et al*. (1997) developed a recognition approach to recognise isolated signs (262 signs) wearing coloured gloves with a recognition rate of 91.3% using HMMs. Their videos in the database were recorded with a single camera. Their system proves that sign language can be recognized using a HMM. Unfortunately they use coloured gloves to capture the feature information.

Coogan (2007) designed a simple gesture recognition approach using an HMM. The features, which are used as an input for the recognition approach, are a combination of the hand shape information and information of the position of the hand in the image. He uses a vocabulary of 17 dynamic gestures and each of them had 20 examples. The identification of the start and end point had to be done manually and the approach focused on only one-handed gestures. He tested the system using different numbers of states and had an average recognition rate of 98%.

Park *et al*. (2008) present a real-time 3D pointing gesture recognition algorithm for natural human-robot interaction which can be used in mobile space. They utilised hand position mapping to estimate the pointing direction. After that, they introduce a left-right HMM which consists of three phases: non-gesture, move_to and point_to. Each phase is composed of three states and a mixture of two Gaussian densities which was used to represent the observation symbol probability distribution. This HMM is trained by the Baum-Welch algorithm using a set of sequence examples. They conclude that large number of states guarantees more accurate tracking results but more states require more training data and training time. Their proposed pointing gesture recognition system showed better results than previous approaches. In their case, HMM works very well but they only recognize three phases. Sign language signs are composed of more than three phases and the signs are more complex than straight movement in the space.

As shown in this section, HMM in gesture recognition works very well for simple gestures in small quantities (e.g. 20 differing gestures). Having said this, an increase in complexity or vocabulary size requires the use of DBN and so these factors are crucial when determining the correct approach to take.

2.4 Summary

This chapter gives an overview of the past research which has been done in gesture recognition. Additionally it outlines the current techniques used in many areas of recognition and recently used in gesture recognition. The research in this chapter focuses on finding solutions for the contributions of section 1.1.

The generation of real-time videos will be discussed in section 4.1. Additionally, creating synthetic videos is explained in section 4.2. However, the use of animated videos can be problematic. The research shows that previous systems used animated videos successfully in ASL recognition (Vogler & Metaxas, 1998).

The classification system mentioned in section 1.1 is not been considered in this chapter. The research of developing this system will be addressed in section 3.3. The final development of this system will be explained in section 5.1.

Skin detection is very complex and requires a lot of time for each to find user the right colour range which is proven by Vethnevets *et al.* (2003). Additionally, the skin detection might not detect every skin pixel and some of the skin object will be left out due to shadow (Coogan, 2007). I use hand and face detection to define a signing space for each person. This is explained in section 5.2.1. For the moment, I can use edge detection and drawing blobs around hands and face on binary images. This works only because the background and the signers' shirt are black. However in future work, I will have to use skin detection.

Yang *et al.* (2004) and Luukka (2009) prove that PCA is very efficient in reducing the dimension of the data and representing the data in a 3D space. PCA allows me to represent the movement of the signer in a 3D graph. This is only possible because the hand motions of the signer are the only changes in the videos.

As previous recognition systems prove, HMMs work well using simple gestures. But Suk *et al.* (2008) show, if the gestures get more complex and the size of the vocabulary increases HMMs are not sufficient anymore. Therefore they combine HMMs with DBN. In this thesis, I focused on HMMs only, using 12 signs for training and testing. The gesture recognition system is explained in section 5.2.

Chapter 3 Sign Languages

This chapter gives basic overview of the linguistics of Irish Sign Language (ISL). First of all, the section 3.1 introduces a basic overview of Sign Languages (SL) focusing on ISL. A few statements can be explained such as ISL does not have an officially recognised structure (Ó'Baoill & Matthews, 2000). Individual signs can be performed in different ways which makes it more difficult to learn and understand the language.

Section 3.2 looks at some components of the language called phonemes which contains hand shape, motion, palm orientation, location and sometimes non-manual feature. The structure of the language and the contrasts with spoken language are outlined. For instance, the majority of the signs do not involve finger spelling but signs to represent words. This basic knowledge of ISL is very important to understand the difficulties in creating a recognition system for SL.

The last part in this chapter explains several notation systems which were developed in various SLs. These notation systems are used to write SL down. These ideas of notation systems are used later in section 5.1 to develop and create our own classification system which can identify each sign uniquely.

3.1 Overview

3.1.1 Sign Languages

Sign language is the visual communication channel used by Deaf people. The first publication of a sign language was in 1620 by Juan Pablo Bonet (Ó'Baoill & Matthews, 2000). Bonet's work introduced the first education technique for Deaf people using a visual alphabet based on finger spelling to support their communication. Over the years different types of visual alphabets were published and more signs were created to describe words without using finger spelling.

Similar to spoken languages, most countries have their own sign language which is in general not based on their spoken language. Three main sign languages used in English-speaking countries are Irish Sign Language (ISL), British Sign Language (BSL) and American Sign Language (ASL).

The following two figures show how different two sign languages can be. Figure 4 is an example of the alphabet of the BSL. This alphabet is signed using two hands. Figure 5 represent the alphabet of the ASL. This alphabet is signed using one hand and some of the signs are close by related to the ISL alphabet signs shown in Figure 7.

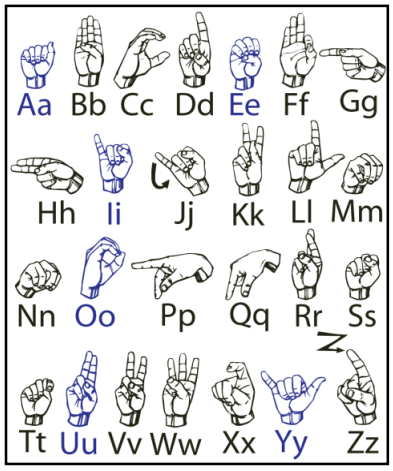


Figure 4: Alphabet of the American Sign Language. ***This alphabeth is signed using one hand.***

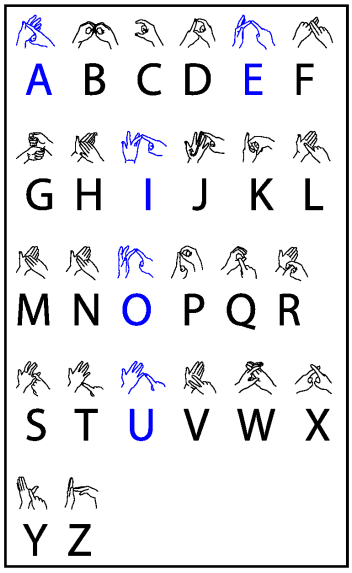


Figure 5: Alphabet of the British Sign Language. ***This alphabeth is signed using two hands.***

In addition to the above three sign languages, almost every country has their own SL. There exists for instance, Chinese Sign Language, French Sign Language and German Sign Language. Every sign language has its own structure and is as complex as any spoken language.

In international meetings the most common spoken language is English. International signs are used at international Deaf events such as the Deaflympics[7] and meetings of the World Federation of the Deaf[8]. International signs are not taught in the standard courses for sign language because they are not used in everyday life.

[7] http://www.deaflympics.com/
[8] http://www.wfdeaf.org/

3.1.2 Irish Sign Language

Irish Sign Language is the main communication process for Deaf people in Ireland.[9] The first signs used in Ireland were British signs and they were introduced in 1816 (Ó'Baoill & Matthews, 2000). Thirty years later, in 1846, French Sign Language was brought in and gave rise to the first Irish signs which were introduced in girls' schools. For this reason ISL is related to French Sign language. Irish Sign Language was taught in boys' schools eleven years later. In those days girls were educated by women and boys by men therefore Irish Sign Language exists in a male and a female form. Nowadays it is mainly the male signs that are taught in schools. ISL and BSL are so different from each other that communication between an Irish signer and British signer can be as difficult as the communication between a native English speaker and non-native English speaker.[10] Irish Sign Language was first used in Dublin and does not have a lot in common with either spoken Irish or English, which makes it more difficult to find a way to convert Irish Sign Language.[11]

Apart from the existence of male and female signs ISL has its own type of construction, which cannot be compared with the structure of spoken or written English. The Figure 6 taken from (Ó'Baoill & Matthews, 2000): 180, 185, shows that

ISL:	YOU COME HERE WHAT-FOR
Translation:	What are you doing here?
ISL:	PEOPLE CROWD TIGHT ME-MOVE-THROUGH-THEM
Translation:	It was difficult to get through the crowd.

Figure 6 : Irish Sign Language example. ***These sentences show that the sign language can not be converted word by word or sign by sign.***

[9] The information can be found on the website of Media Sign - http://www.signmedia.com/
[10] The information can be found on the website of Deaf culture – http://www.deaf.ie
[11] The information can be found on the website of the Deaf society – http://www.irishdeafsociety.ie

in most cases it is not possible to translate word by word or sign by sign, only the entire sentence expresses the complete meaning.

3.2 Sign Language Linguistic Features

3.2.1 Alphabet

The alphabet is used to spell names or words which do not have their own specific sign. It is composed of 23 static signs and three letters are dynamic signs (J, X and Z). The following figure 7 shows the alphabet of ISL.

Some signs are combinations of a few alphabet signs, executed quickly so as to make a sign. There exists finger spelling and finger-signs. Finger spelling is a word which is sign by each letter individually, for instance "ZOO". Finger-signs compromise only a few letters of the word to make the sign, for instance this is used for all the names of the months. For example January would be built up of J-a-n with an accompanying rotation of the wrist.

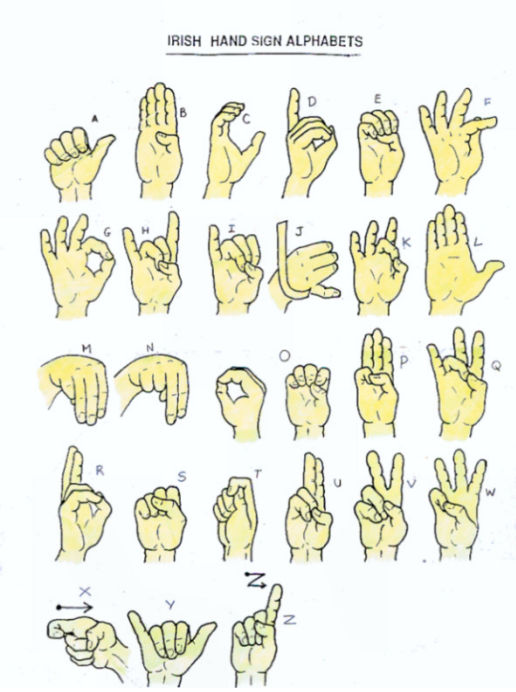


Figure 7: Alphabet of the Irish Sign Language. ***This alphabet is signed using one hand.***

3.2.2 Signing Space

The signing space is an area, which extends from the waist up to the face and all signs are articulated within it. Only very few signs are performed outside of the signing space, above the head such as “cloudy” or below the waist, such as “skirt”. The size of the space itself depends on the signer because of the body shape but also because of slight variations in the movement during signing. Some people do a sign closer to the body than other people, this might be because they might not be physically able to sign that specific sign in the signing space. Figure 8 shows an animated person from the front and the side. The small rectangle demonstrates the space used for finger spelling. The signing space is represented by a big rectangle in the first figure and in the second figure from the side. Using the signing space in a recognition system allows us to reduce the size of the data. The system focuses only on the region of interest (in this case the signing space) and does not need to process the other part of the images. The algorithm which I developed for the signing space detection can be found in section 5.2.1.

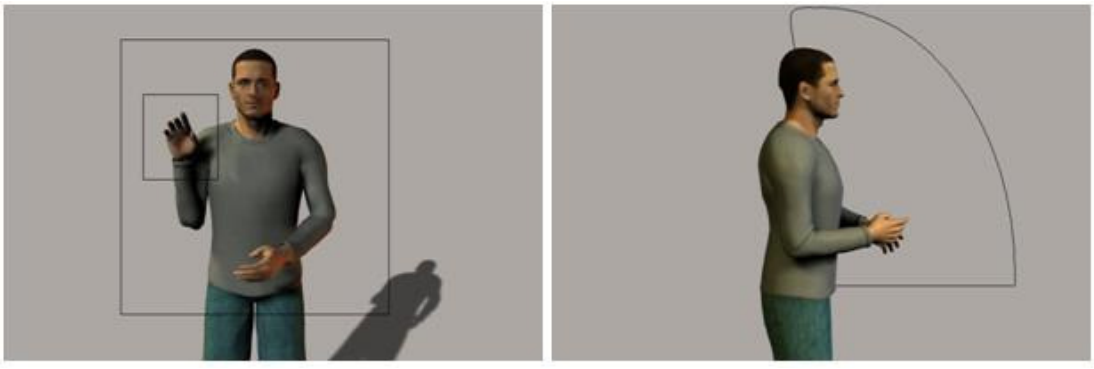


Figure 8: Signing space from different angles. ***In the left image, the small rectangle shows the finger speliing space and the big rectangle shows the signing space. On the right image, the signing space in front of the body is shown.***

3.2.3 Non-Manual Features

Communication is a process in which two people give and receive information. The most common way of transmitting information we use is speech, or verbal language. The exchange of information without using speech is nonverbal communication, or body language, and it is the oldest way for humans to exchange information. (Boyes, 2005)

Sign Language is a composition of articulation of hand signs, body and facial expression and lip movements. The hand signs themselves are a combination of phonemes which include movement, position of the hand, position of the hand in front of the body and the hand shape. The language can only be completely understood if the observer brings all those parts together.

Sign language can be divided in two main parts: signs and non-manual features (NMFs). The NMFs are used to express emotions and consist of facial expressions such as eyebrows, mouth, eyes and cheeks (Ó'Baoill & Matthews, 2000). Focusing on the body the NMFs are a composition of head position (e.g. tilting) and upper body and shoulder movement. NMF do not only include emotions. NMFs also facilitate the description of a location, which is very often shown by an arm moving in that direction combined with a look at the described place in a virtual space. For example, it explains the location on the shelf if a box was placed there.

Some signs in the language have identical hand signs, and meaning can only be determined using these NMF. For instance, the sign for "pain" and "expensive" are identical in the movements and hand shapes but not in the NMFs. The sign "pain" and "expensive" are composed of an open hand shape such as the letter "L" of the ISL alphabet but the fingers do not touch each other. Additionally the hand would

face down and wave beside the body. If face expression is left out, the second person will not be able to figure out the sign by itself.

As explained in this section the NMFs are very important for the understanding of SL. NMFs recognition can be done in the field of facial expression recognition and would be a big project itself. This thesis focuses on gesture recognition and therefore only the hand gestures are taken in consideration for the developed classification and recognition system.

3.2.4 Classifiers

As of today, there has been no official defined structure for the ISL. There exist only a few suggestions on ISL structure, which should be followed for a better understanding. Most of all, the more advanced signer tends to be more experienced in signing and therefore uses a lot of signs called classifiers.[12] A classifier is a sign, which refers to a set of individual signs to express numbers, quantities, shapes or sizes of humans, animals or objects. For example, the index finger could refer to a pencil or a person. A classifier is used like a pronoun in spoken language to avoid using a large amount of signs. In this way the classifier aids visualizing the exchanged information. For instance, locations in a room are explained using a classifier and NMF. The signer would use a classifier to describe the object, and the eyes gaze combined with the direction the hand movement will define the location. (Ó'Baoill & Matthews, 2000)

[12] These classifiers are not to be confused with the classification system described in section 5.1. Both are two different things. The classifiers are signs which refer to a set of individual signs for a better understanding of the sentence. The classification system has been developed to identify each sign individually using features.

3.2.5 Dominant Hand

Numerous signs use two hands to express meaning. In only a small number of signs, both hands move in the same direction using identical hand shapes. Usually, both hands act independently using different hand shapes and different directions (Ó'Baoill & Matthews, 2000). The main or most important movement in the sign is signed with the dominant hand, which is the right hand for a right-handed person and the left for a left-handed person. This project focuses on signs with the right hand as the dominant one.

3.2.6 Tenses

In ISL, tenses are not represented in the same way as in written or spoken language. ISL includes four tenses: present, past, future and the future in the past. The verbs in sign language are signed, and do not have a different form to apply the tenses. (Ó'Baoill & Matthews, 2000) The common and simplest tense is the present, and it is articulated in the signing space, which is explained in the beginning of this chapter. In general, the past is signed using the dominant hand and using the sign 'back' which is indicated by moving the hand over the shoulder and point behind the signer.

The future tense can be signed in two different ways. On the one hand, the sign 'will' can be used and combined with the actual verb. On the other hand, the future can also be expressed using key words such as "tomorrow" or "next year".

3.2.7 Personal Name

A few words in ISL do not have a sign and are therefore signed using finger spellings, for instance, 'ill', 'zoo' and new words. Among friends, everyone has a name-sign. It is an Irish tradition to use alphabetical letters to assign a name-sign to

each person. Generally, the chosen letters are based on the initial letters of the person's name. This respectful tradition appears to be only used in Ireland where many signs are initialised. (Foran, 1996)

3.3 Sign Notation Systems

Since the early 19th century, linguistic and sign language researchers have developed several notation systems (Ó'Baoill & Matthews, 2000) for SLs in order to find a way to write sign language. August Bébian worked out a notation system called Mimographie and it is used in French Sign Language. However, education for Deaf people came into existence and attention was focused on lip reading rather than sign language.

Early in the 1960s Dr. William Stokoe (Stokoe, 1972) developed a notation system using three parameters of the phonemes, hand-shape, hand-location and hand-motion, which describe ASL. Figure 9 gives an example of the symbols of the Stokoe notation system. The big letters indicate the hand shape, the small letters or symbols attached to these hand shape symbols, indicates arm position or face

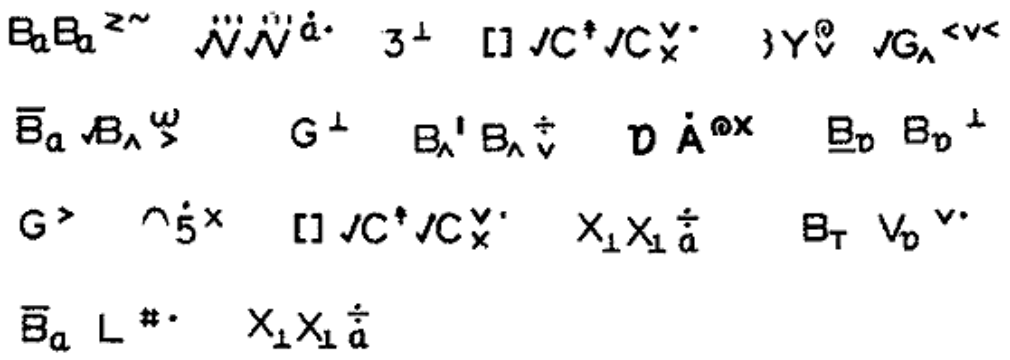


Figure 9: Notation system of William Stokoe. ***Each group of symbols represent one sign of the American Sign Language. Each sign use a symbol for handshape, arm position and motion.***

expressions. The superscripted symbols describe the motions and can describe if the motions are repeated or reversed.

Figure 10 is one example of the symbols of Stokoe notation system.

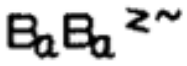


Figure 10: Single example of Stokoe notation system. ***The symbol 'B' indicates the hand shape, 'a' the position in the signing space and 'z' indicates the motion.***

B means that both hands used a flat hand which can be open or spread hand and thumb out or in. The small *a* describes the position where the sign will be signed, in this example proximal side of forearm or wrist. The small *z* describe the motion, in this case a left and right motion. Finally, the ~ means that the motions are made in alternation. The example above is a sign of ASL. (Stokoe, 2005)

He published the notation technique in his book 'Sign Language Structure' (Stokoe, 1980) and it has been used by SL researchers. Stokoe Notation was created for ASL.

In 1974, Valerie Sutton (1995) developed a sign language notation system called SignWriting. Her goal was to find a simple way of writing down the movements of sign language. Later, she invented her own system for a notation system to record dance movements.

SignWriting can be written down in four different ways. Firstly, the most complex task of SignWriting uses symbols to represent the full body and to write down every movement of the body. The second approach she invented focuses only on the phonemes such as the head and hand shape in order to give information about the specific location and movement of the head and hand in the signing space.

Another two approaches use handwriting and record less specific details because they write down only a summary of the full signed conversation.

For each approach there is no defined situation in which it will be used. The most complex method of notation is generally used for formal documents in order to record all details in an important signed conversation. The third and fourth approaches are mainly used to write a note to a friend who is fluent in their sign language. Figure 11 gives an example of Sutton SignWriting. [13]

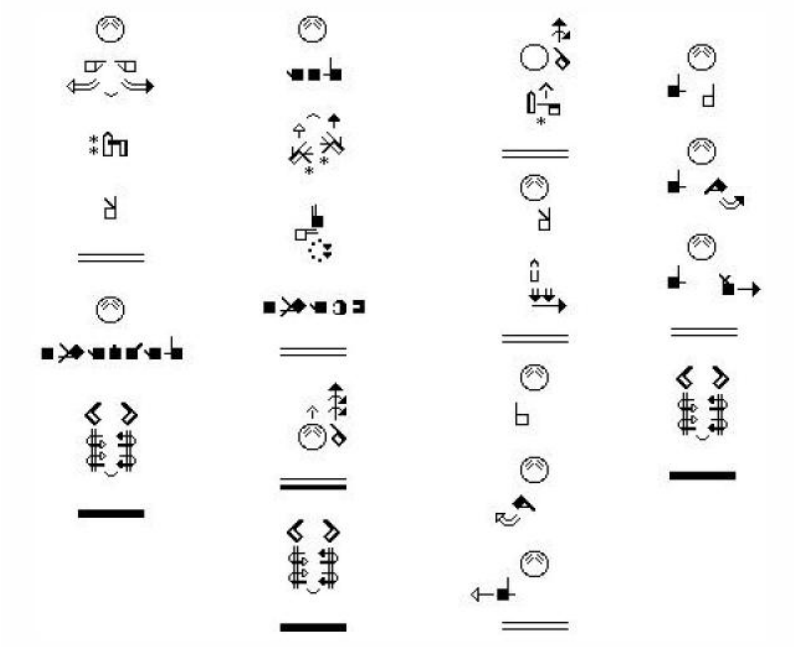


Figure 11: Example of Sutton SignWriting Notation. ***This is an example of the most complex notation system of SignWriting. The different symbols indicate: hand shape, head position, hand movements and information about the body position and movement.***

In this figure, each sign which has been made is described in detail. Starting with the position of the signer's head to express which way the person is facing while signing

[13] Official website of SignWriting - www.signwriting.org

to the others. This is followed by symbols of the hands which can be additionally defined with arrows to express the movements. Furthermore there are symbols which

Figure 12: Single example of Sutton SignWriting. ***This symbols represent a sign of the American Sign Language and they indicate head position, hand shape and hand movement.***

give information about the body movements and position. Figure 12 shows two single examples of Sutton SignWriting. The symbol with the circle indicates the orientation of the head, in this case the signer is face the other person. The two symbols below the head symbolize the right and left hand. In this case, both indicate the *d* hand shape. The symbol of the left hand is filled black which indicates that the palm of the hand is facing away of the body. In the second example both hands facing away from the body but in the first example, the right hand is facing the body. The second sign has an addition symbol which looks like an arrow. The arrows describe the movement of the hand.

In 1990 another electronic notation system for sign language was developed and is called HamNoSys (Hamburg Notation System) (Prillwitz et al. 1987). The goal here was to have an international notation system, which can be used for all sign languages in order to facilitate communication. This system records details about the two hands individually and the details about different parts of each finger. This

approach is able to record details about the NMF as well. Figure 13 shows an example of the sign “difficult” of the ASL using symbols of HamNoSys. Additionally the meaning of each symbol is added.

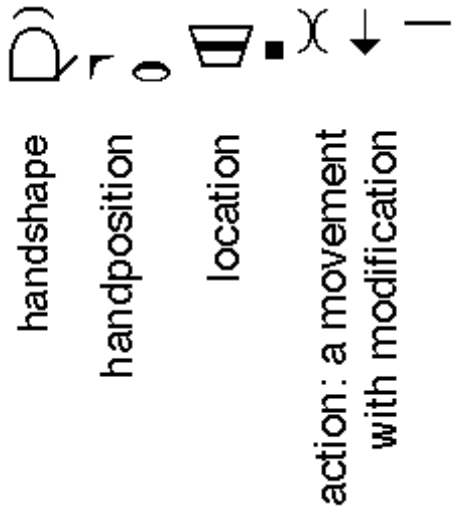


Figure 13: HamNoSys example of the sign "difficult" of the American Sign Language. ***Each phoneme used in HamNoSys is represented by a symbol.***

3.4 Summary

Firstly this chapter outlines general background knowledge in SL with the focus on the ISL. SL is a different way of communicating and has to be understood before the language can be recognized and each sign classified. Section 3.1.2 shows several problems for ISL recognition. One is that the ISL cannot be converted word by word into a spoken language or written form. Another difficulty is that there exist male and female versions of signs which have to be considered. Using single signs, not connected in sentence and creating for each sign an individual model addressed many of the mentioned difficulties.

Section 3.2 gives an overview on the SL linguistic features. The signing space is a very important feature. First, the signing space locates the signer in the image. On account of this the position of the signer in the image can be detected and the signing space can be adjusted on each signer. Second, the size of the data can be reduced before processing because only the region which is included in the signing space will be considered. More details about the signing space detection can be found in Chapter 5.

NMF are very important in SL but as mentioned before recognizing the NMF would be a part of facial expression recognition and not gesture recognition. However, this should definitely be considered for future work.

Section 3.2.5 describes one of the main problems in ISL recognition. Each signer uses his dominant hand for the main movement in the sign. To avoid this problem in the system, only the right hand used as dominant hand is considered.

Section 3.3 which explain the notation system is very important. This section describes different ways to write the SL in symbols. Most of the notation system use different phonemes to symbolise a sign. In this thesis, ISL does not need to be written but each sign has to be identified uniquely. Therefore a classification system was developed. This classification system does not need any symbols to identify a sign but rules are needed which can be applied to separate each sign individually. The classification system is explained in Chapter 5.

Chapter 4 Data Generation

The sign language recognition system which I have developed, needs two different types of data: training data and testing data. The training data is composed of examples which include ideal examples and realistic variations such as light interference, different camera position and variations in the movements. The testing data is always random examples of all the existing data.

The generation of the data repository consists of real-time and synthetic videos. First in section 4.1, I created a database of ISL real-time video which contains sequences of a live signer. This database is used to train and test the system. Secondly in section 4.2, synthetic data is created in a 3D animation package. This allows me to make small changes in the videos such as changes in the movement in the sign or environment influence such as different light conditions or camera positions. Both databases consist of decontextualised signs. The following subsections will explain the different kind of data generated to train, analyse and test the recognition system which is explained in Chapter 5.

4.1 Real-Time Signs

A non-Deaf sign language tutor was recorded with a video camera signing 1400 different signs about eight times, totalling 11,200 videos sequences. There are two different kinds of categories: situation-dependent and alphabetic.

The type of situation-dependent category include everyday situations, such as travel, health, education, numbers, clothes, colours, family, food, seasons, weather, sports, technology, home, jobs, days of the week and special days. Some of these groups contain a few sentences which are mostly used in those situations. A list of all these categories including the names of the signs can be found on the attached CD. In the recognition system only a few signs are used. Those signs do not belong to the situation-dependent categories but to the alphabetic categories. However, these signs were created to be used in future work.

The alphabetic category classifies the signs by their hand shapes and is divided into 23 sections. O'Baoill and Matthews (2000) defined 66 different hand shapes in ISL. All these hand shapes look very similar to the letters of the alphabetic hand shapes. These categories consist of the alphabetic hand shapes but a few hand shapes such as *m* and *p* or *i* and *j* or *n* and *u* look in some signs very similar and for that reason they are in the same categories. Therefore the categories are reduced to 23 groups as shown in Table 1.

Categories	Number of signs, existing in our database
Hand shape a	44
Hand shape b	28
Hand shape c	34
Hand shape d	29
Hand shape e	19
Hand shape f	25
Hand shape g	19
Hand shape h	27
Hand shape i and j	22 and 2
Hand shape k	14
Hand shape l	48
Hand shape m and p	23 and 35
Hand shape n and u	29 and 17
Hand shape o	24
Hand shape q	11
Hand shape r	61
Hand shape s	31
Hand shape t	34
Hand shape v	23
Hand shape w	36
Hand shape x	1
Hand shape y	4
Hand shape z	1

Table 1 : Alphabetic Categories of the real-time video. ***This table indicates the division of the 23 categories and the amount of signs in ech category.***

For the recorded real-time videos, a few criteria were introduced. The SL tutor has to be positioned in the middle of the image and always at the same distance of the camera. Another criterion was that the tutor has to wear a black shirt and a black background was used. These criteria simplify the recognition in the way that the system is focusing only on the signs without being influenced by environmental changes. The signer was recorded in normal day light using a simple camera.

Figures 14 gives an example of one of the real-time videos shown frame by frame. The person in the video signs the sign “adult”. This is one part of a video used in the system which can be found on the CD. The signer always starts in the position where the hands are resting on the legs. In this case the frames stop when the sign has been

signed. The videos used in the system, the signer always moves the hands back to the start position. The recognition system which is described in Chapter 5 recognizes only single signs and not full sentences as shown here.

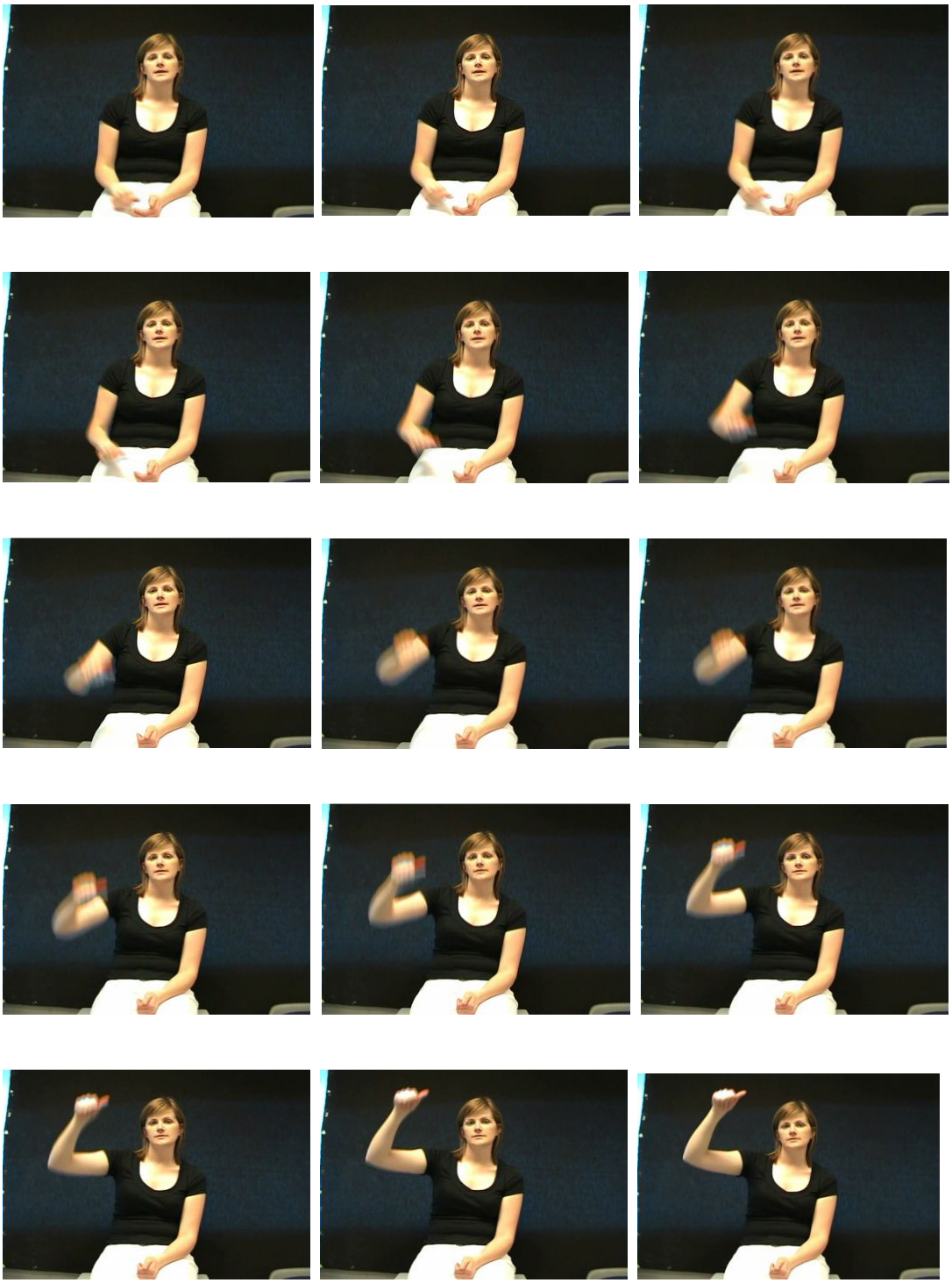


Figure 14: Sign "adult" in real-time video. ***This images show each step of the signers, starting in the start position and ending in the last position of the sign.***

4.2 Animated Gesture Creation

Recording people with a camera takes a lot of time, especially to get all the samples needed. Therefore, these days the computer is able to artificially generate human-like characters in a three dimensional environment which is similar to a real human being and can be used for analysis.

Poser[14] is an animation tool which allows the user to work in a three-dimensional animated environment. Learning how to use Poser is very simple and with a few simple steps the user can create and modify their animated videos. A few human and animal characters are built into Poser. For more tools, characters, animals or environments, the internet provides users with large libraries.

Poser can be used for animation in two different ways: visually or by implementing scripts. With a few clicks and drags in Posers' interface, it allows the user to change the environment and move the characters around or put the character in a specific pose. All changes can also be done by implementing scripts with the programming language Python. Python allows the implementation of detailed movement with all the necessary coordinates. Poser provides function and method for the implementation with Python.

Figure 15 shows the working interface of Poser. A library which provides predefined characters can be found on the right side of the interface. The left side provides the user with tools such as camera positions controller and light controller. On the bottom the user can play the movie or move to particular frames. Above the play

[14] Information found on the official website of Poser – http://poser8.smithmicro.com/dr/index.html

menu, the user can find the tools to move the character or body parts in different positions.

Figure 15: Poser interface. ***This interface was used to generate several signs and influence the signs with different light conditions and camera positions.***

The advantage of Poser is that it allows the user to learn how to work with the program easily and it has a simple working environment, for instance, changes can be made quickly with a few mouse clicks. The videos and the images created in Poser can be exported, and used and changed in other programs as well. The advantages of using Python in Poser are that the movement can be implemented very precisely and can be used repetitively in different videos or with different characters, for instance in signing. Both ways allow the user to create a lot of data for testing in a shorter period of time to real-time videos and to make small changes in the environment and on the character. Figure 16 gives an example of various characters

and poses which are predefined by Poser. I used the character of the man to generate the synthetic videos.

Figure 16: Poser characters in various poses. ***The character in the top left is used to generate the animated videos. The other two could be used for future work.***

I use this tool to model realistic human motion in order to analyse the recognition approach using animated videos of the same sign and include motion and environment variations in the sign. Generating the movement interactively is a fast solution and only useful if the action with that character is only needed once. A repeated action should be implemented with Python. In this case, the actions are repeated using small movement variations or different light influences or different camera positions.

4.3 Gesture Animation Scripting

Python[15] is an object oriented programming language and is used in a wide range of application areas such as internet development, desktop GUI and three dimensional graphics. Python implementation is open source software and is integrated in Poser. As mentioned before, Python scripts in Poser give the user the opportunity to implement repeated actions with variations of certain features. In this sign language recognition system, I use Python scripts to implement animated environment changes on each sign such as changing light interference and camera position or implementing variations in the movement of the signers. These examples are not used for testing the system but they are used to analyse the effect of those influences on the system. Considering that the recognition system focuses only on the movements in the signs, animated videos result in very smooth motions. In the real-time videos the motions include a lot of noise. Therefore the synthetic videos are only used for analysing various environment or movements changes.

Figures 17 gives examples of the sign “allow” created in Poser using Python. Only the region of interest in the image is shown. The character begins with the start position of the sign and not as in the real-time video. The movement in the animated video consists of moving the hand down while keeping the same hand shape. Figure 16 shows frames of the sign “allow” generated as synthetic video, only every fourth frame of the video is given. The generated videos can be found on the CD.

[15] Information has be found on the official website of Python – www.python.org

Figure 17: Animated video with the sign "allow". ***Only every fourth frame of the video is given. The character starts in the start position of the sign and stop in the end position of the sign.***

4.3.1 Implementation Structure

This section gives a small introduction in Poser Python, the difference between Python and other languages and explains some specific characteristics of Poser Python, I used to generate the videos. Those who have experience in other programming languages are used to using curly braces ({}) or semi-colons (;) to

define the structure in the source code. This is not used in Python and for small, short scripts, classes are not used either.

To define the lines, which belong for example to a loop, indentation is used. The first line which is not indented anymore shows that the loop stops at that position and that line will be executed after the loop has finished. This is valid for every kind of loop such as *for*, *while*, *if*, *else*, etc. The following example demonstrates the different levels in the source code. Figure 18 is an extract of one of my scripts it only shows the different levels in the source code. The explanation and description of the meaning of this source code can be found in the next subsection.

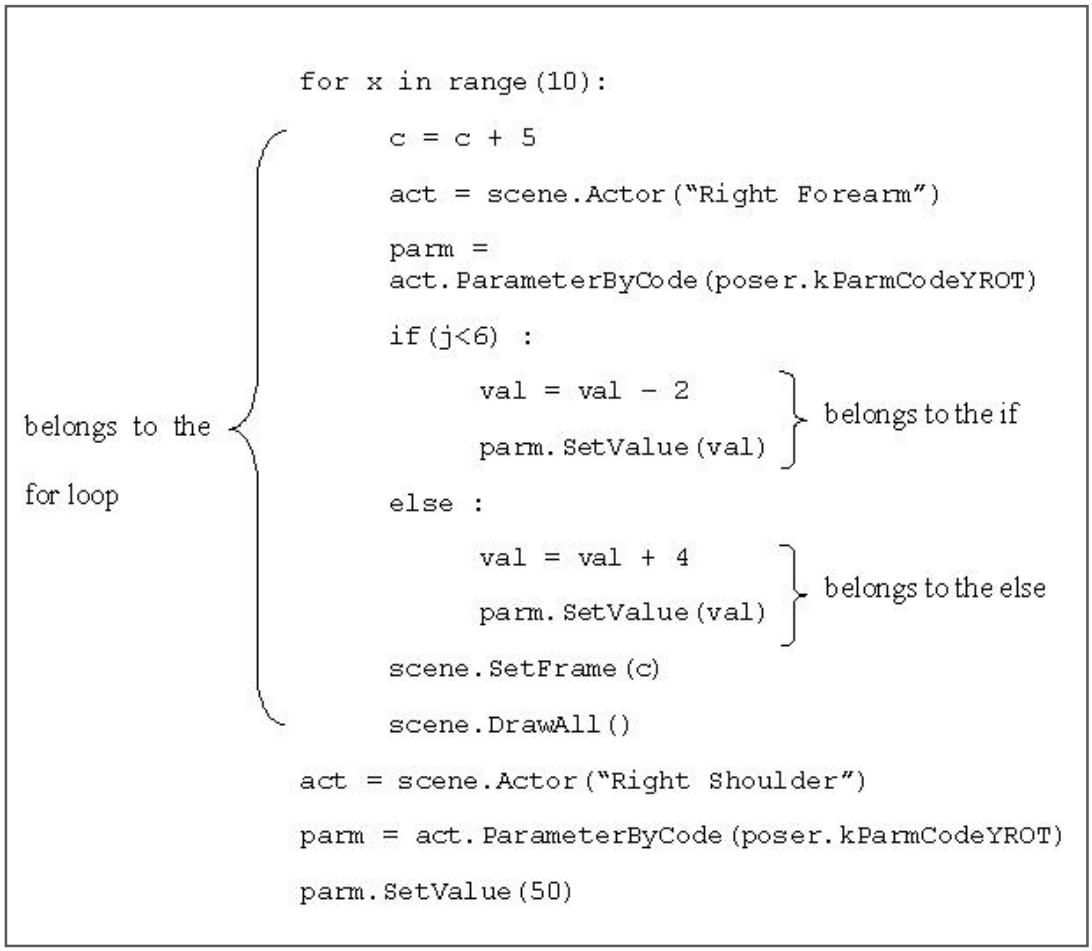


Figure 18: Example of a gesture animation script. ***This script interates a loop to control the Y rotation of the right forearm of a character by setting keyframes within the loop. The following code then rotates the right shoulder part and set the value on 50.***

Another important point in Poser Python is the difference between types and codes. Type is a category of data. The most common data types are numbers or strings. Poser Python includes those data types but adds some particular for Poser to the standard Python types. In my case the most used Poser type is the ActorType which represents an actor within a scene. The term actor refers to any individual item of a scene, which includes every body part, camera, light, etc. One important feature of an actor is that it has to be an item which is able to move, so a shoulder or hair is an actor but a body is a figure.

On the other hand, I have codes, which are representations of given parameters such as coordinates and attributes. In Java, codes represent the reference to an attribute of another class which is imported. These codes make it easier to refer to the internal data of Poser and for the user it is easier to know which parameter is called or set. To show the Poser Python interpreter that they are predefined Poser Python variables, they are used with the prefix “poser.”.

4.3.2 Python Method to Move an Actor

This subsection gives a short introduction to implement a Python script in Poser. This overview explains step by step the main methods I used in my Python scripts. A lot of the time, the Python script needs to import other packages. This has to be done initially using a script. The source code will look like *import math*. This imports the package math which includes the mathematical formulas. At the end of the line there is no ‘;’ as would be expected in other programming languages.

To set a variable is very easy and looks the same or very similar to other languages,

e.g. c = c + 5

My Python scripts include a variable *c* that defines the current frame number. A variable does not need to be defined to a certain type; the information which will be set in the variable will define the variable itself.

To change something in a scene, first the user needs a reference to the actual scene in Poser. This can be done with the code *poser.Scene()* and has to be saved in a variable which can be found in my script with the name *scene*.

The next step will be to define which frame should be changed. *Frame()* returns the number of the current frame. All frame numbers in Poser Python are relative to a starting frame of 0. This means that the frame number in Python is 1 less than the same frame referenced to the Poser GUI. The method *SetFrame()* allows to set the frame number. In this case, starting with the first frame, the current frame number will be set to 0.

To get access to the parameters of an actor, the actor can be defined and given by the call *Actor()* which include the actor name as parameter. The name is the given name of Poser and can be found in the GUI pull down menus. The call

act = scene.Actor("Right Forearm")

gives the possibility to change the parameter of the right forearm. The parameter of an actor can only be changed after it has defined which parameter will be changed and those are stored in a variable. This can be done with the following call

parm = act.ParameterByCode(poser.kParmCodeYROT).

This gives access to the Y rotation parameter of the right forearm and can be change while a new value is set with the method *parm.SetValue()*.

Finally every change has to be drawn in the frame which can be done with the method *scene.DrawAll()*.

Important: In my Python script which creates any gesture, the lines which deal with the frames are written as comments. Those lines look like *#c = c + 5* and *scene.SetFrame(curFrame)* has to be modified. Otherwise the user will see the changes while the script runs but they will not be saved in the frames. The lines have to be changed and should look like the following code: *c = c + 5* and *scene.SetFrame(c)*.

A Poser Python introduction and how to use Python in Poser can be found in appendices. More animated videos and the source code can be found on the attached CD.

4.3.3 Parameter used in Poser Python

Table 2 gives a list of parameters and explains their meaning. These parameters are used in my scripts for various situations and movement.

	Parameter	meaning
general		
	kParmCodeXROT	rotation about the X-axis
	kParmCodeYROT	rotation about the Y-axis
	kParmCodeZROT	rotation about the Z-axis
	kParmCodeSCALE	Amount of the scale in each direction
	kParmCodeXSCALE	amount of scale along the X-axis
	kParmCodeYSCALE	amount of scale along the Y-axis
	kParmCodeZSCALE	amount of scale along the Z-axis
	kParmCodePOINTAT	degree to which an actor set to point at something will actually point at it
	kParmCodeValue	Placeholder for a value. Usually, these values are used functionally to control other things such as full-body morphs
specific		
Camera		
	kParmCodeFOCAL	Camera focal length parameter
	kParmCodeFOCUSDISTANCE	Camera focus distance parameter (affects depth of field effect)
	kParmCodeYON	Camera parameter specifying a far clip plane distance
Light	These codes are used to set the light types.	They are typically used in conjunction with the actor.SetLightType()
	kLightCodeSPOT	Spotlight
	KLightCodesPOINT	Point light
	kLightCodeLOCAL	Local light
	kLightCodeINFINITE	Infinite light

Table 2 : Parameters used in Poser Python. ***These are the main parameters used to move an actor.***

4.4 Summary

This chapter introduce the generation of the data used in the developed recognition system described in section 5.2. This data is composed of real-time videos and synthetic videos.

As mentioned in the first contribution in section 1.4, the real-time videos are necessary to see how the system reacts in real-time situations. Only one sample of each sign is not enough to test the recognition system. This database includes 1400 individual signs. Only a small amount, specifically 12 signs, was used to test the recognition system because only a small part of the classification system, which is described in section 5.1, is realised in the recognition system. The recognition system focuses at this stage only on hand movements in the signs and therefore the important factor in the selection of the data was to test the system on different movement.

According to the second contribution in section 1.4, the synthetic data is very useful in analysing the system on situation influence by various side effects such as light conditions, camera positions or simply variations in the signers' movement. Generating this data in an animation tool and not in real-time videos is a very positive decision. Firstly, the data is generated much faster and more precisely because the videos can be created using coordinates. Secondly, the synthetic videos do not include any noise. In real-time videos the signer will never be able to make the movements in the sign as smooth as the character in the animated videos. Therefore, the results of the synthetic data are very useful for analysing different situations which will be discuss in the next chapter.

Chapter 5 Recognition System

The previous chapter described the generation of the database in real-time and synthetic videos which will be used in the two developed systems. These two parts solve the first two statements of my contributions in section 1.4.

This Chapter presents the two systems I developed. Therefore it is divided in two main sections. Section 5.1 introduces a classification system which I developed to categorise each sign uniquely. Creating this classification system is the third statement in my contributions in section 1.4. This system is based on the ISL information gained from Chapter 3. The structure of ISL is needed to understand the construction of the signs. Additionally, based on the ideas gained from section 3.3, I was able to realise a classification system for ISL which is based on some of the phonemes. This classification system is able to categorise every sign stored in the database of the real-time video. The generation of these videos is explained in section 4.1.

The second part, section 5.2, describes a prototype for a recognition system which is based on the recognition of the hand movement of ISL signs. Developing the gesture recognition system is the fifth statement in my contributions in section 1.4. There is a big difference between classification and recognition and certainly most times both are used in the same problems. Classification is a process in which individual objects are divided into groups based on their recognised characteristic features. In general, recognition involves identifying objects through their characteristic individual features that have been defined before. Unfortunately until now, there does not exist

a full functional gesture recognition system, which could be used to control a computer or any computer device.

Two fundamental strategies are considered for gesture recognition: feature extraction and recognition. Instead of using motion capture details in the video, Principal Component Analysis (PCA) was used to represent movements in a three-dimensional space. This representation of movement using PCA is the fourth statement in my contribution in section 1.4. PCA is explained in section 2.2. Based on this representation, for each gesture a small Hidden Markov Model (HMM) presented in section 5.2.3, was developed and trained. HMMs constitute an algorithm developed for speech recognition and can be used to represent the temporal variation within a gesture. All of the HMMs are connected to one large system, trained and tested with real-time videos from a database. The source code for the recognition system can be found on the CD.

5.1 The Developed Classification System

5.1.1 Why is a New Classification System needed?

In the area of gesture recognition, a notation system for classification is needed. After the recognition process has identified the individual features, the items can be classified into groups based on those characteristic features. The existing notation systems, described in section 3.3, are based on writing the sign language in symbols but they are not always capable of identifying a sign uniquely. Therefore a classification system had to be built which allows us to implement a gesture recognition system.

Both systems, recognition and classification, are based on ISL. The classification system, I developed, is based on the identification of each individual sign. After recording real-time videos of ISL signs, a classification system was developed focusing on different phonemes: difference between left and right hand, hand shape, movement, location in the signing space in relation to the signer and hand orientation in the space. This classification system is used to categorise the signs using the features which were identified during the recognition process.

Figure 19 gives an overview of the structure of the classification system. These steps split a sign into different parts. The system first differentiates the left and right hand. Thereafter, for each hand certain phonemes will be considered. The phonemes consist of hand shape, hand movement, hand position in relation to the body and hand orientation in the space. The combination of the result of each part identifies the sign. The final model, to recognize ISL signs is based on these steps. Each step is explained in detail in the following subsections.

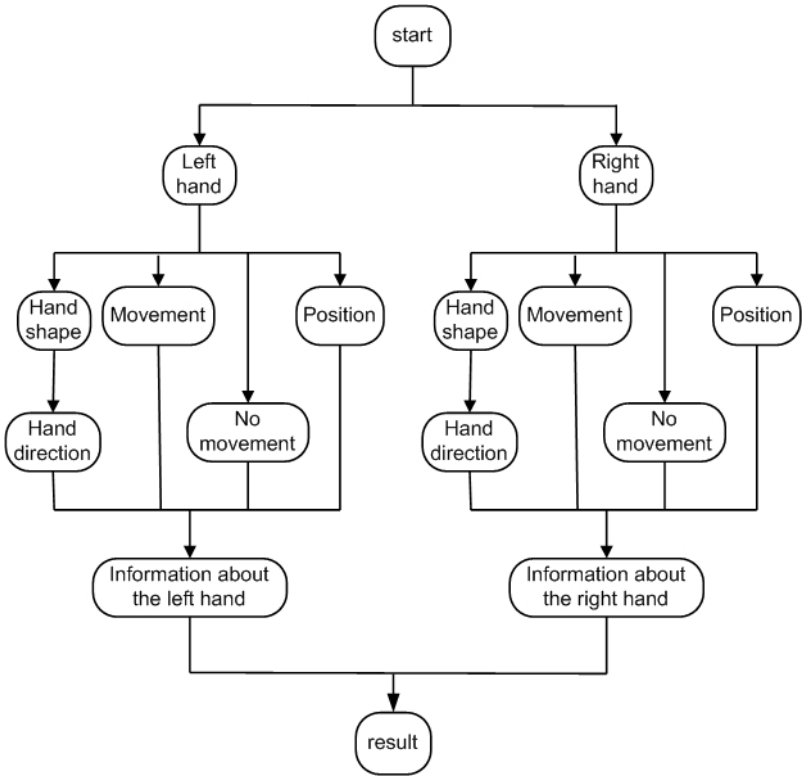


Figure 19: Structure of the classification model. ***This model split a sign into phonemes which are used ot identify each sign uniquely.***

5.1.2 Hand Shapes

The signs on which this notation system was tested are a part of a database which is explained in section 4.1. After recording the signs, they are stored as follows: the main and most used signs are categorized by their hand shapes, the other signs are stored in categories used in everyday situations. The signs which are used in the system are all classified through their hand shape in alphabetic groups. This means a sign is classified in the group “hand shape a” if the dominant hand forms mainly the hand shape ‘a’ during signing that sign. The storage and generation of the databases are explained in the Chapter 4. A list of the individual signs in each category can be found on the attached CD.

The first step in identifying a sign is to focus on the hand shape. Both hands do not always use the same hand shape. Therefore both hand shapes have to be considered separately. However the main focus lies on the dominant hand which is in my case the right hand. In the recognition system, explained in section 5.2, the hands will be considered together at this stage of the prototype.

5.1.3 Movement

The second step of the classification system focuses on the signers hand movements. I defined 17 individual movements which I was able to recognize in real-time videos. The real-time videos which are used for this recognition approach are two dimensional, but the individual movements occur in a three dimensional environment. The 3D perception is important due to the fact that for instance, rotations in movement or those signs moving away from the body can be taken into

account. Figure 20 shows the different movements into which the sign will be classified.

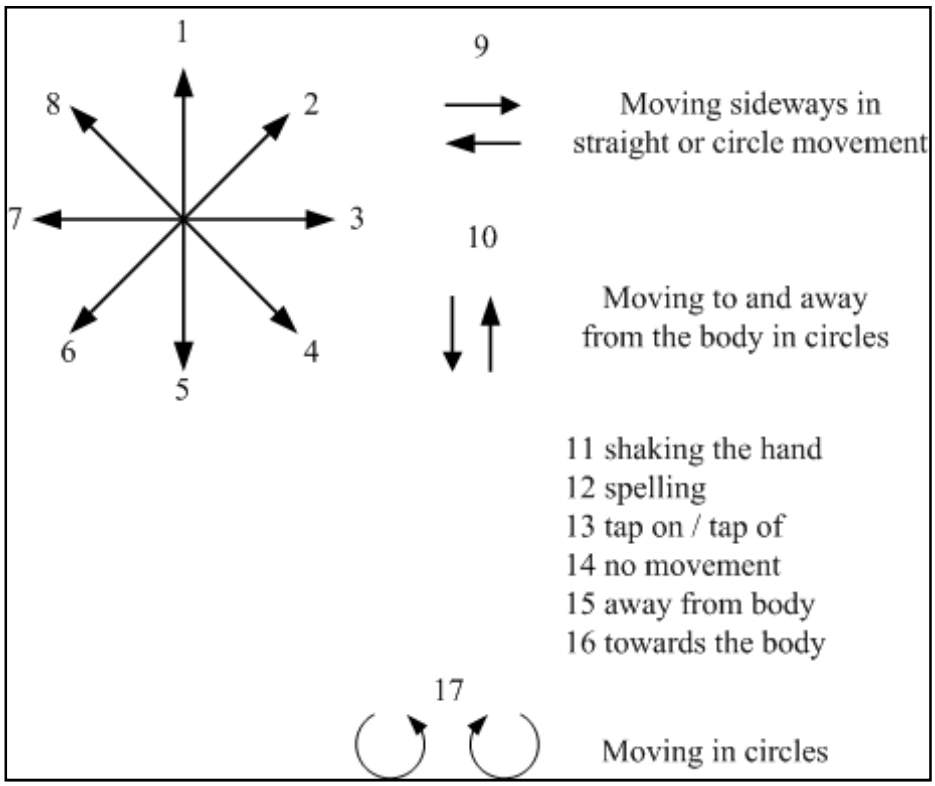


Figure 20: Movements of the classification model. ***These 17 described movements describe the different motions used in sign language.***

My developed gesture recognition system, introduced in section 5.2, focuses only on this step of my classification system. The other parts are not yet been realised. This is possible because the signs chosen for the recognition system have very clear differentiation in the movement. For this reason I was capable to test the recognition system without having to take the hand shape into consideration.

5.1.4 Position

The third step of the classification system is the position of the hand in the signing space. The signing space is divided into 9 different parts as shown in the Figure 21. This signing space does not include the signs above the head or below the waist and these will not be considered at the moment. A solution for this problem could be to create an area 10 which would be above and below the signing space. As mentioned before the size of the signing space can vary for each individual and therefore the size of these dividers can change as well. My developed algorithm for the generation of the singing space is explained in section 5.2.1, however the algorithm is only detecting the signing space and is not used for dividing this space into parts.

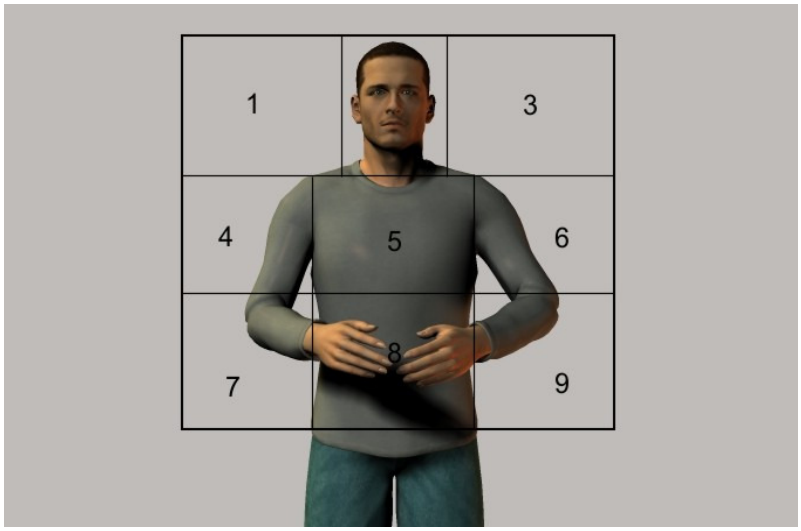


Figure 21: Signing space divided in 9 sections. ***Defining in which part of the signing space the sign is been signed, helps identifying each sign.***

5.1.5 Hand Orientation in the Space

The first three steps in the classification system can classify most of the signs, but a few signs are very similar and therefore this last step has to be taken to be able to identify all of the signs. This last part defines the direction in which the hands themselves are oriented in the space. This rule is used to separate signs such as “meet” and “accident”. In these signs, both hands use the hand shape ‘a’ and come together with a horizontal movement in front of the body. The only difference in these two signs is the hand orientation in the space, meaning the direction in which the hands are turned in the space. This step is left out for the moment because the description of the hand orientation in the space has not been defined yet. For example, taking into consideration the direction in which the thumb is pointing at is not enough. Additionally the directions in which all the fingers are pointing at have to be taken into account as well.

5.1.6 Classification Tree

The classification can be shown in a decision tree. Figure 22 outlines the full decision tree. At the point where the data is classified through the third step, the hand position in front of the body, the sign should be classified. For only a few cases such as the signs “meet” and “accident”, the orientation of the hand in the space is needed, which has not been added.

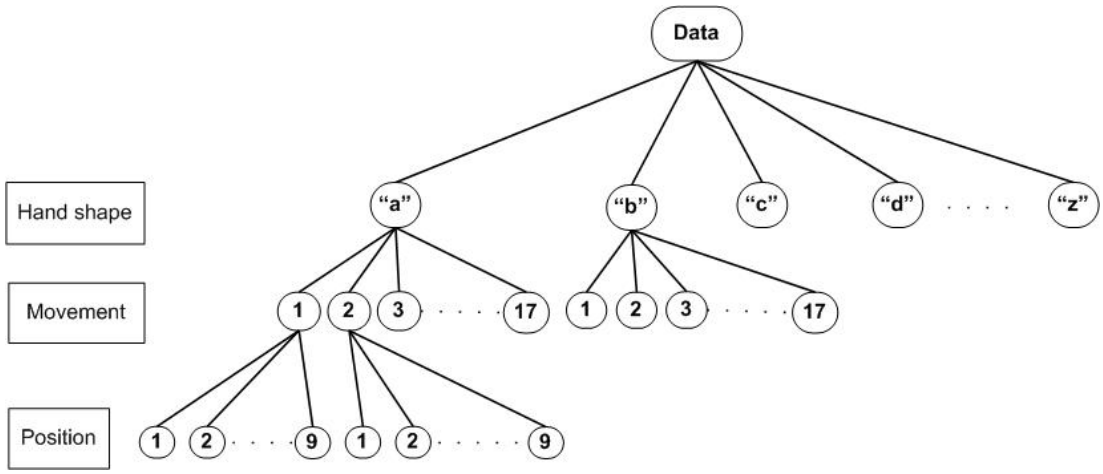


Figure 22 : Classification Decision Tree. ***Each sign can be categorised after applying all the steps of this classification tree.***

Table 3 shows the classification of some signs. All the signs in this table belong to the category of the hand shape "a". The first three columns show which movement has been applied to the hand (right or left). The last column explains how to solve the cases where hand shape and movement alone cannot differentiate the signs. None of these examples require the position of the hand in front of the body to classify the selected set of signs. Another example of hand shape "b" can be found in Appendix A.

Movement (using signs of hand shape "a")			
Number of the movement	**Left hand** (can be any hand shape)	**Right hand** (hand shape "a")	Other features needed to tell the underlined signs apart
1	Advice	Adult, Advice, Age, Ago	left hand shape
2			
3		Meet, Accident, Across	Orientation of the hand
4		Fail, Absent	
5		Nothing, Account	
6			
7	Meet, Accident	Address	
8			
9	About	About, Wash	
10	Act	Yesterday, Act	
11		Numbers	
12			
13		Knock, My, Again	
14	Able, Fail, Knock, My, Nothing, Numbers, Sorry, Wash, Yesterday, Your, Above2, Absent, Account, Across, Address, Adult, Affectionate, Again, Age, Ago	Affectionate	
15		Able, Your	Orientation of the hand
16			
17		Sorry, Above2	

Table 3 : Classification of the signs of the hand shape "a". ***This table shows an example that the classification system is capable of identifying each sign.***

As mentioned before, only the second step of the movement is developed in the recognition system. This is possible because the signs chosen for this recognition system have very clear differentiation in the movement.

5.2 Recognition System

In the previous section 5.1, I described step by step my developed classification system. I now present a prototype of a gesture recognition system based on the classification system. The gesture recognition system is the fifth statement of my contributions. As mentioned before, at this stage of the recognition system, the system focuses only on one phoneme, the different movements of the signers' hands.

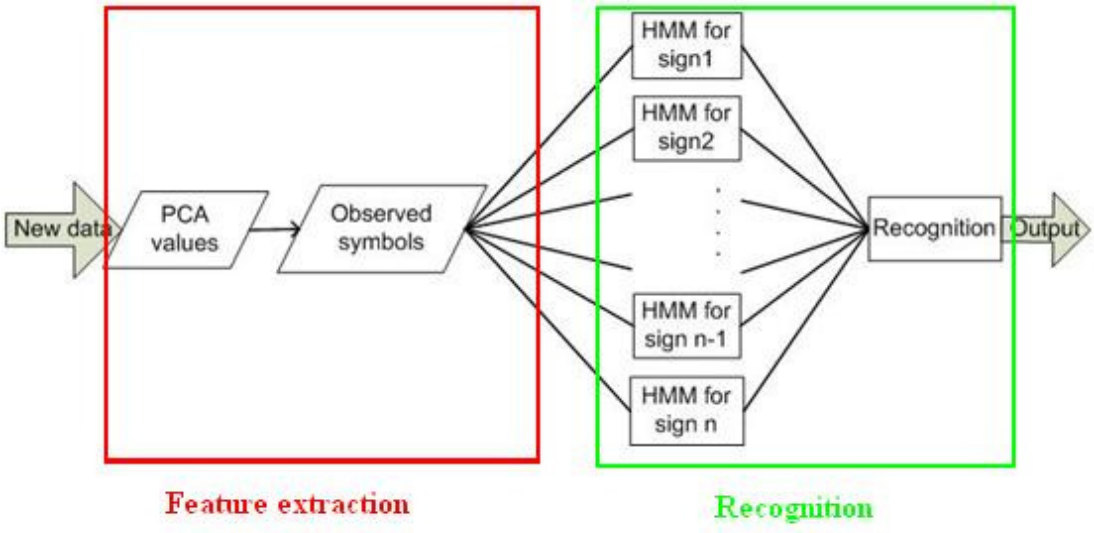


Figure 23: Recognition System Overview. ***This system contains feature extraction which is composed of calculating the PCA values and the observed symbols and the recognition which is composed of an HMM model for each sign individually and the recognition part.***

Figure 23 gives an overview of the structure of the system. Before any data can be put into the system, the video has to be converted into a frame sequence. The new data, which consists of single frames, is then input into the system. These frames will be converted to three dimensional coordinates using PCA. Out of those PCA values, the observed symbols will be calculated for each stored eigenspace. Each sign has its own calculated eigenspace and its own HMM model. The observed symbols of each

eigenspace will be used as input for the specific HMM, created for that sign in that eigenspace. After every HMM has computed the likelihood (Korb & Nicholson, 2003), the system will classify the sign by comparing these results. The likelihood is a probability which shows how likely the new data fits to the compared training data.

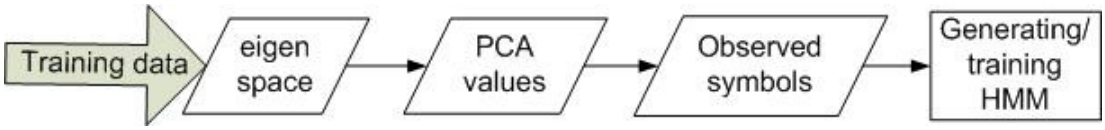


Figure 24: Training of the system for a new sign. ***First claculating of the eigenspace, PCA values and observed symbols, then a HMM model is generated and trained.***

Figure 24 shows the training process of a new sign. First multiple samples of the new sign are used to calculate the eigenspace and its PCA values. Afterwards, the observed symbols are created which will be used to generate the HMM model. The HMM is generated during the training which creates the hidden states and all the probabilities. The finished trained HMM will then be connected to all the other generated HMMs for the recognition of new unknown data.

The HMMs was tested with 12 signs of the real-time videos. Of each sign, 6 variations were put into the training set. Each sign was tested with four samples, two of them were in the training data and two of them were new data. The signs taken from the training set are used to prove that the HMMs work well because the HMM is generating itself during the training. The two unknown samples are used to see how well the system recognises new samples of the trained signs. The signs which were tested in the system are: “Adult”, “Allow”, “Begin”, “Black”, “Body”, “Easter”, “End”, “Evening”, “Lady”, “Man”, “Morning” and “Tomorrow”. The videos and the frame sequences of these signs can be found on the CD.

I use animated gestures to illustrate the effect of changing the user's pose, the camera position or lighting conditions. Implementing animated videos can be very useful because it is possible to keep all parameters constant and change only the parameter of interest. Thus I can see the effect of changed parameter without interference from anything else. It is very difficult to do this with real gestures because it is impossible for a human signer to make the same gesture in exactly the same way twice. This method is used to analyse how much the system is influence by small changes. These changes can be different light conditions, various camera positions and slight variations in the signer movements. The animated videos are represented in the space without any noise which allows me to analyse the different interference without any influences.

The following subsections describe each process of the recognition system. The system is implemented in a programming tool called Matlab[16]. The first subsection explains the generation of the signing space. Another part of the feature extraction is the calculation of the PCA values and eigenspace. The explanation of the PCA can be found in section 2.2 and is not further explained in this chapter. Section 5.2.2 explains the representation and interpretation of the PCA values in a 3D space and the generation of the observed symbols. Additionally, this section includes representations of the impacts arising from different influences such as different light conditions and camera position. Finally this section ends with a few exceptions of movement which cannot be added to the system yet.

Following the pre-processing steps, section 5.2.3 introduces the recognition system. The recognition system is implemented using HMMs. This section includes the structure, the training and results of the approach. Section 5.2.4 introduces a distance

[16] Official website of Matlab – www.mathworks.com

metric to improve the recognition of the movement. Finally section 5.3 presents and discusses the results of the recognition system.

5.2.1 Signing Space

Before PCA can be applied, the images have to be pre-processed to reduce the computation cost to the system. This includes finding the region of interest of the images which reduce the dimensionality of the data in advance. To define the signing space, primarily the head and the hand/arms have to be detected. As mentioned in the literature review, there are many ways to detect the arms and face.

One method would be to use skin detection. Skin detection is a very complex task. The main problem is that the skin consists of a wide range of colours. These colours do not just change for each person but they change as well in different lighting conditions. The database which is used for this system includes only one signer but in different lighting conditions. Therefore the complex task of skin detection did not work for every video. In the videos, the signer sits on a desk and sometimes the light changes the skin colour so that it is very close to the colour of the desk in which case the skin detection would include the desk as a skin object.

To avoid all these problems, the conditions for the signer were to sit in front of a black or very dark background and to wear a black shirt. Therefore the easiest way is to convert the images into black-and-white binary images. The head and the neck will only be considered as one labelled component if they are connected to each other. Very often the neck and the face are not connected, therefore the images are converted using a threshold.

Figure 25 shows the results using different thresholds on the same frame. In some cases, using a too small threshold could include objects of the environment such as a chair or desk as a skin object. The best result was reached with a threshold of 0.4.

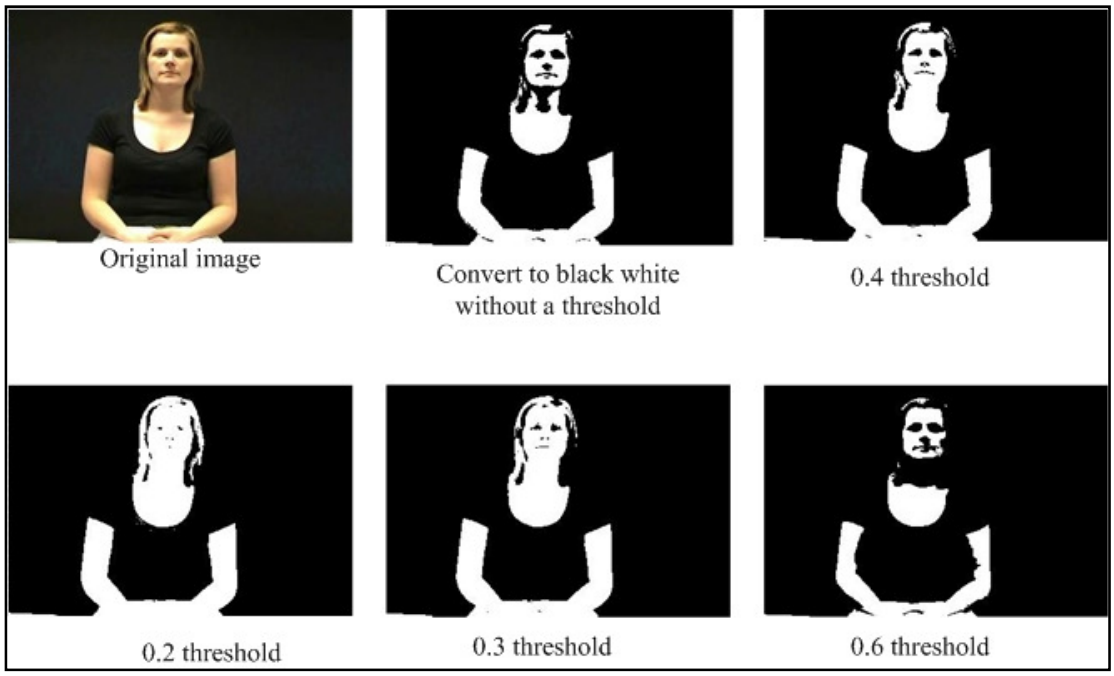


Figure 25: Converted image to black-white using thresholds. ***This process starts with the original image (top left) which is converted into a binary image (top middle). The other four images show the binary images using different thresholds.***

After converting the image into black-white, the detected areas will be labelled as separate regions. Thereafter the created label matrix will be converted into an RGB colour image for the purpose of visualizing the detected regions as shown in the Figure 21 below. The two largest labelled objects, the arms together as one and the head, are filtered out and surrounded by two boxes. These two boxes are then connected together. The width of the signing space was defined as the width of the arm box added to the width of the face box on each side, which adjusts the signing space to each individual signer.

Figure 26 shows the last two steps. On the left, the labelled and coloured regions, and right, the signing space represented by the green box.

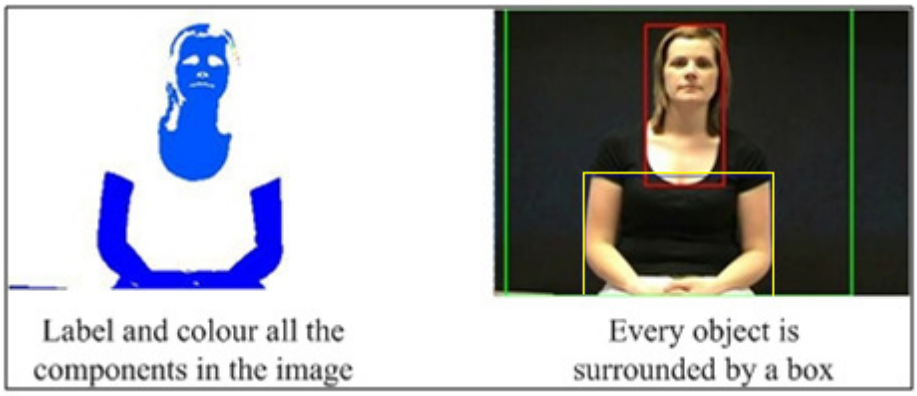
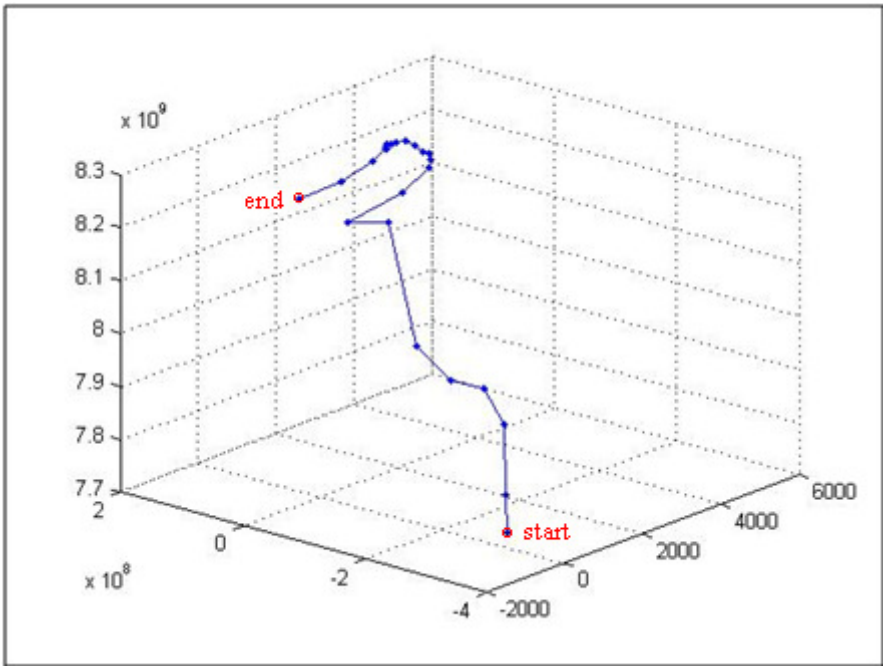


Figure 26 : Detection of the signing space. ***The head and the hands in the image are labeled (left) and thereafter boxes are drawn around these components to create the signing space.***

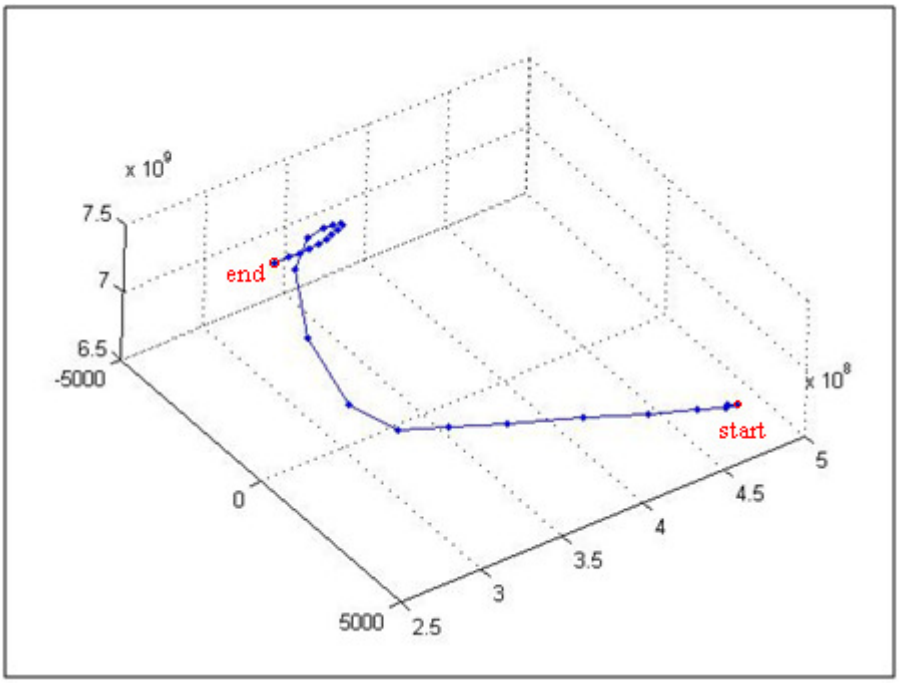
Every unimportant space around the signing space will be cropped out in every image to reduce the processing time of the approach. At this stage in the development of this approach, it is important that the signing space is always the same size. This is not a big problem because the data which is used always contains the same person in the same location, therefore there is no need to change the signing space. That is why the cropped size was defined for each frame to 352x240 pixels.

5.2.2 Gesture Representation

For this recognition approach, the same signing space is used for each gesture and is defined by coordinates in the image. This is only possible because the signer is in the same location of the image in every video. Therefore if the signer would be in a different location, the motion would not be fully represented using PCA. In our case, the PCA (which is only used on the cropped image), reduces each frame of the video to a point in a three-dimensional eigenspace. Graphs, 1 and 2, give an example of two gestures which are processed by PCA and represented in a three-dimensional space.



Graph 1: Motion representation in three-dimensional eigenspace for the sign "adult". ***The shape of the signs motion is represented with its start and end point.***



Graph 2: Motion representation in three-dimensional eigenspace for the sign "man". ***The shape of the signs motion is shown including its start and end point.***

Each graph shows at which point the sign starts and where it ends. The areas where the points are close together demonstrate slow movement and fast movement is expressed with larger distances between each point.

To find out the start and end point of a sign in the eigenspace, I created an interface shown in Figure 27 which visualises each step of the procedure. The interface shows on the left the representation of the principal component (PC) values in the eigenspace and on the right, the current frame of the signer is represented. All the frames are represented as blue points in the graph and the current frame is surrounded with a red circle of its corresponding point in the PC space. The red circle will not be deleted in the next step which allows the user to follow each step of the video. The user is able to observe the entire process of one video by clicking on any key of the keyboard.

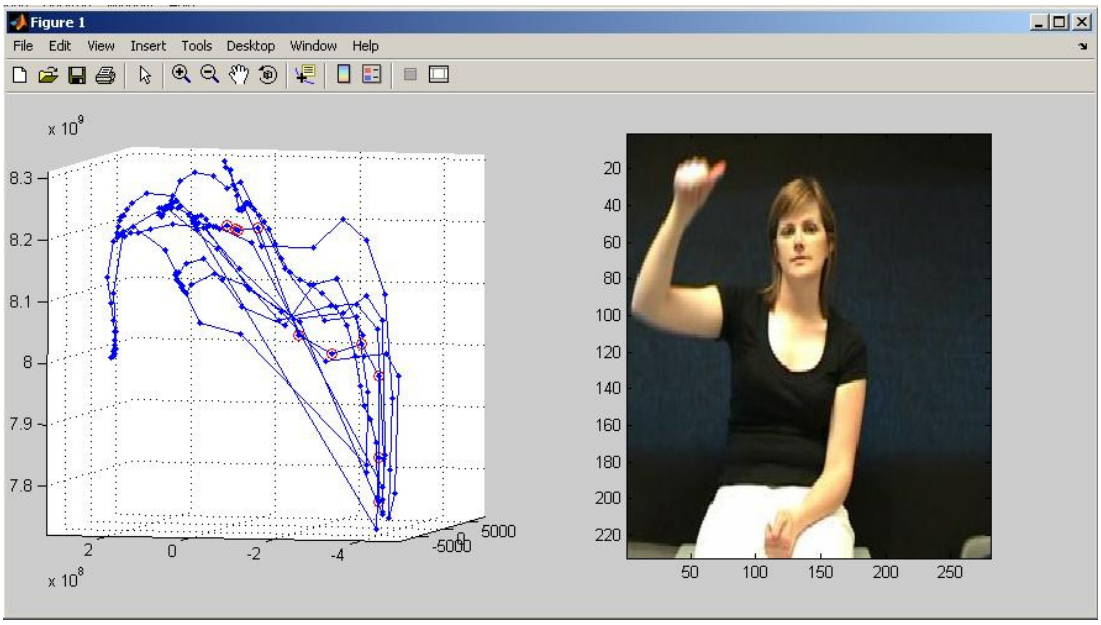
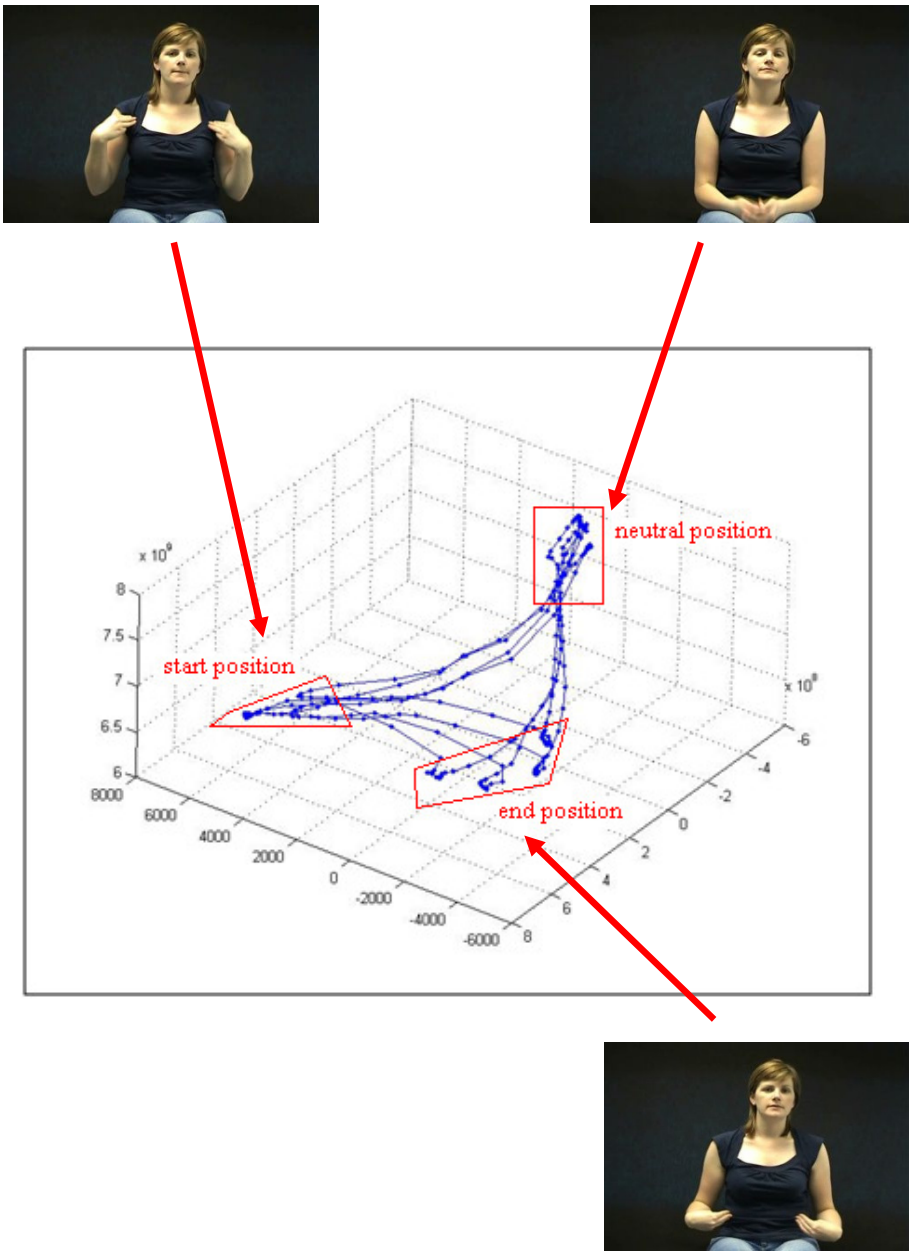
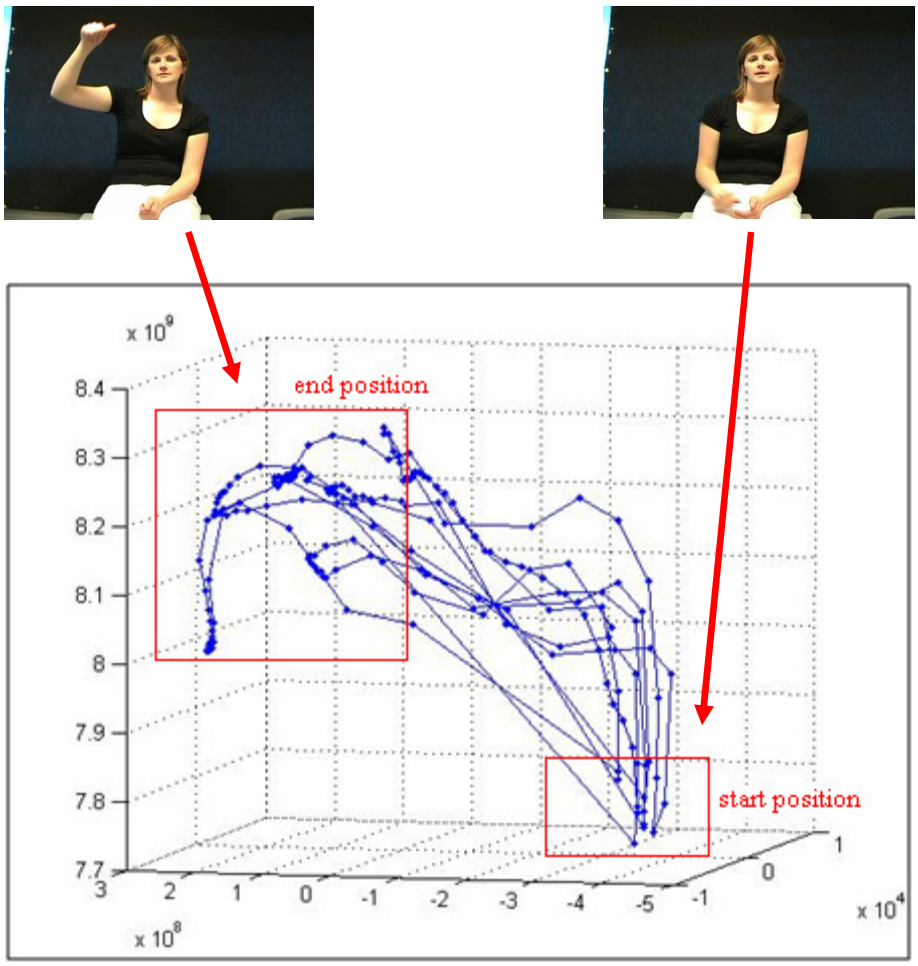


Figure 27: Implemented interface of Matlab. ***This interface is used to detect each point in relation with its corresponding frame visually, so the start and end position of the signs can be defined.***

We try to identify each sign uniquely using their shape and coordinates in the graphs. Therefore a single example of a sign is not enough for identification. Every sign will be calculated and displayed in its own eigenspace and this eigenspace will be used for every variation of that gesture. Each point in these graphs represents a frame of the video. Points from the same instance of a sign will be connected together. These graphs show how dissimilar each gesture is. However, this is not always the case. The more the gestures look alike, the more their trajectory in the graph will resemble each other. The best way to analyse gestures is to represent each gesture with different variations in the same eigenspace. This is shown in the following two Graphs, 3 and 4. The graphs show the sign “man” and “adult” with their start and end positions. The neutral position means that the signer has her hands resting on her legs before starting the next sign.



Graph 3: Sign "man" including movement variations. ***The main positions of the sign are marked including frame samples of that position.***



Graph 428: Sign "adult" including movement variations. ***The main positions of the sign are marked including frame samples of that position.***

In Graph 4, there is no neutral position identified because the sign starts from the neutral position. In this case the start position and the neutral position are the same.

In this manner, the proposed system will use animated gestures to observe and train variance of gestural parameters into the classification system. This provides the following advantage over existing methods:

- Control of signing space- by placing the animated character in static position, full control of signer movement can be obtained

- Control of lighting conditions- lighting parameters can be controlled at the rendering stage and thus are not subject to the large variance found in natural environments.

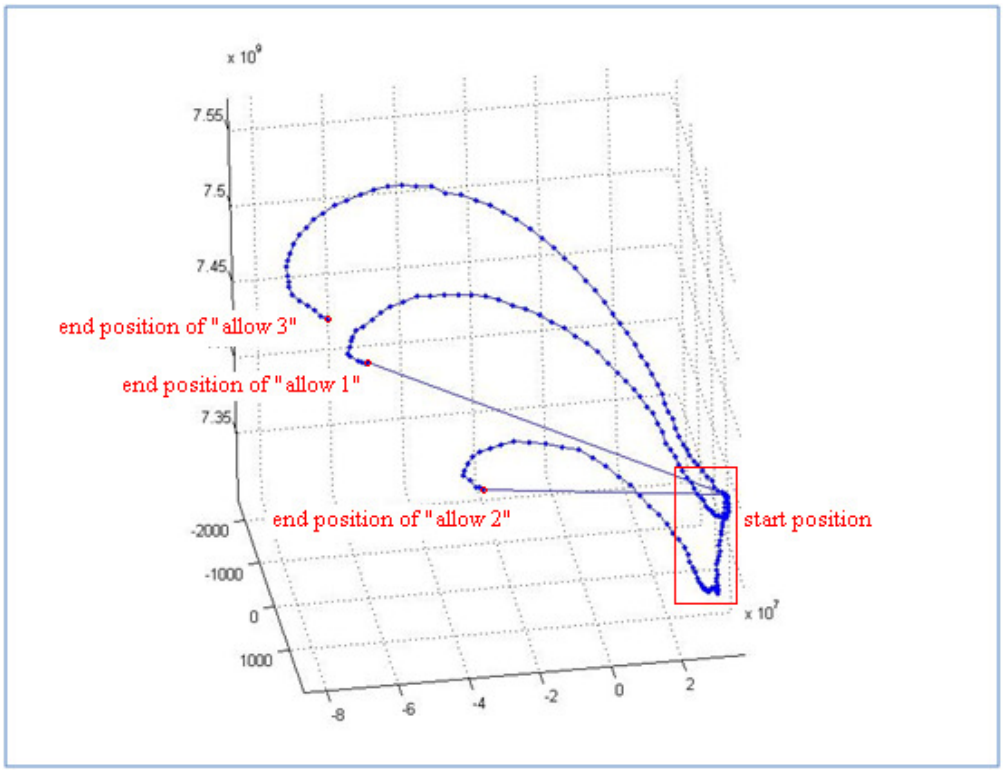
- Control of signer movement- by using coordinates in the implementation the animated character can be moved very precisely.

- Control of camera position- by placing the camera in different position the signer movement can be analysed using different point of view of the signs.

5.2.3 Impacts of the Representation of the Movement in the Graphs

At the moment the real videos and synthetic videos are not commensurate. This is because the synthetic avatar has a very different appearance from the real signer and the lighting conditions and camera parameters are different. Therefore it is not possible to compare real and synthetic videos in the same eigenspace. It is my intention in future works to try to model the real situation accurately in Poser so that the two sets of videos may be directly compared to each other.

The representation of the movements can easily be influenced by small changes in the environment or by signer themselves. Graph 5 shows three examples of the sign “allow” created in Poser as described in section 4.2 using different angles of the arm during the movement.



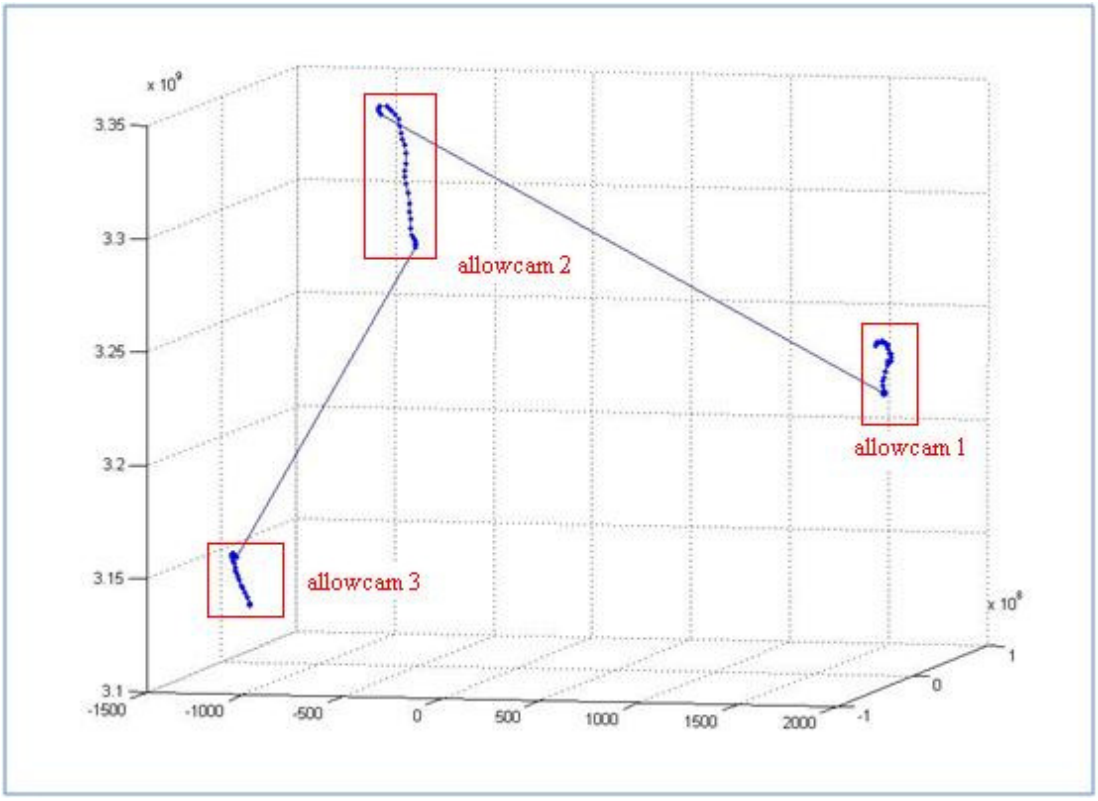
Graph 529: Sign "allow" generated in Poser using different angles in the arm movement. ***Small changes in the arm motion do not have big effects on the shape but the position of the shape in the space.***

In “allow 1”, the signer moves the arm down in a straight line to the camera. In “allow 2”, the signer moves the same way as in “allow 1” but the end position is to the left of “allow 1”. In “allow 3”, the signer moves again in the same way as “allow 1” but this time the end position is to the right of “allow 1”. As seen in the graph, those small changes in the movement can have big changes in the graphs.

On the other hand, using different camera angles can have a serious effect on the recognition. Figure 28 shows the different camera positions which were used. The first frame of each video is shown. First the front and normal view is represented. The second picture shows that the camera moved to the left and in the last picture, the camera moved to the right.

Figure 28: Animated character using Poser. ***Left image shows"allowcam1", middle image shows "allowcam2" and right image shows "allowcam3".***

Graph 6 shows the same gesture using different camera positions. The main difference is that each time the camera angle was changed the gesture occurs in a different part of the eigenspace. Additionally the gesture has a slightly different shape which is not very clear in this graph. The different shape results from the movement variances which arise from the different camera perspective.

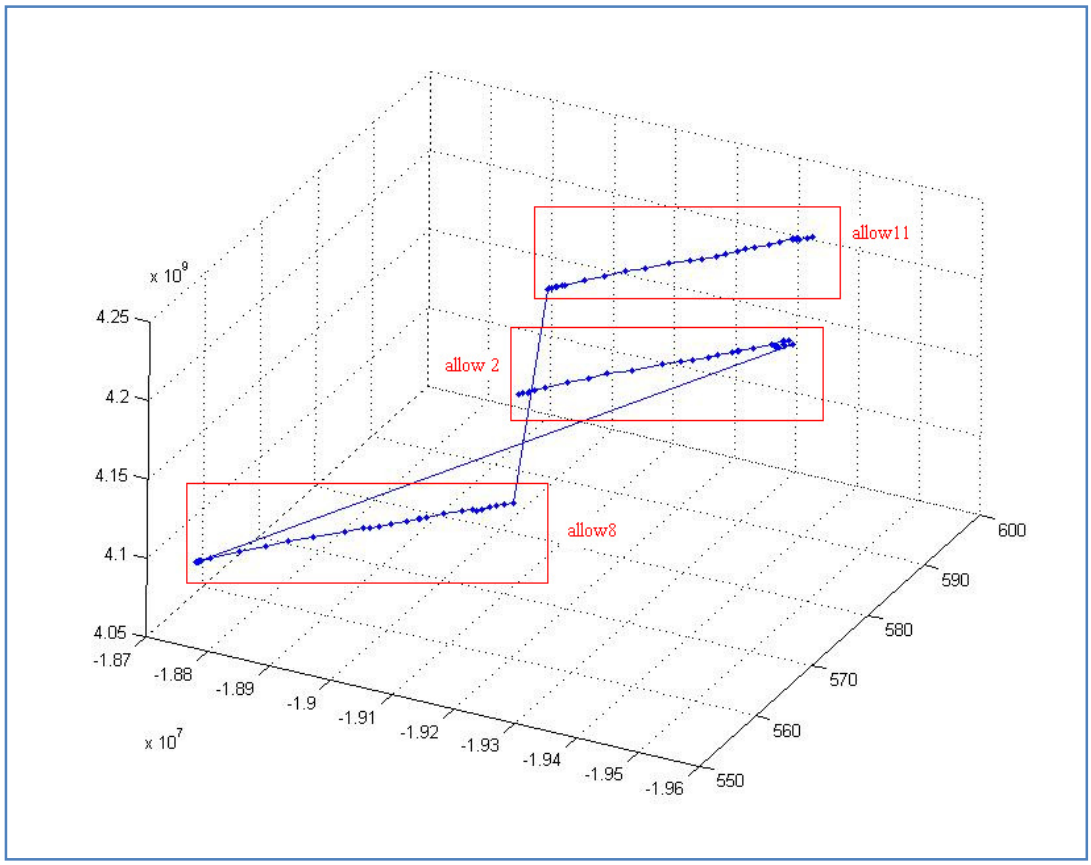


Graph 6: Sign "allow" using different camera positions. ***The different camera positions show the shape of each sample in a different position in the space.***

It is not only the camera position that can affect the motion results in the eigenspace. The lighting condition and the skin colour can also have a significant impact on the recognition. Figure 29 shows three frames of three different movies and each video uses different lighting conditions. The effects the various light conditions have on the graphs can be seen in Graph 7.

Figure 2930: Animated character with different light conditions. ***Left image shows "allow2", the middle image shows "allow8" and right image shows "allow11".***

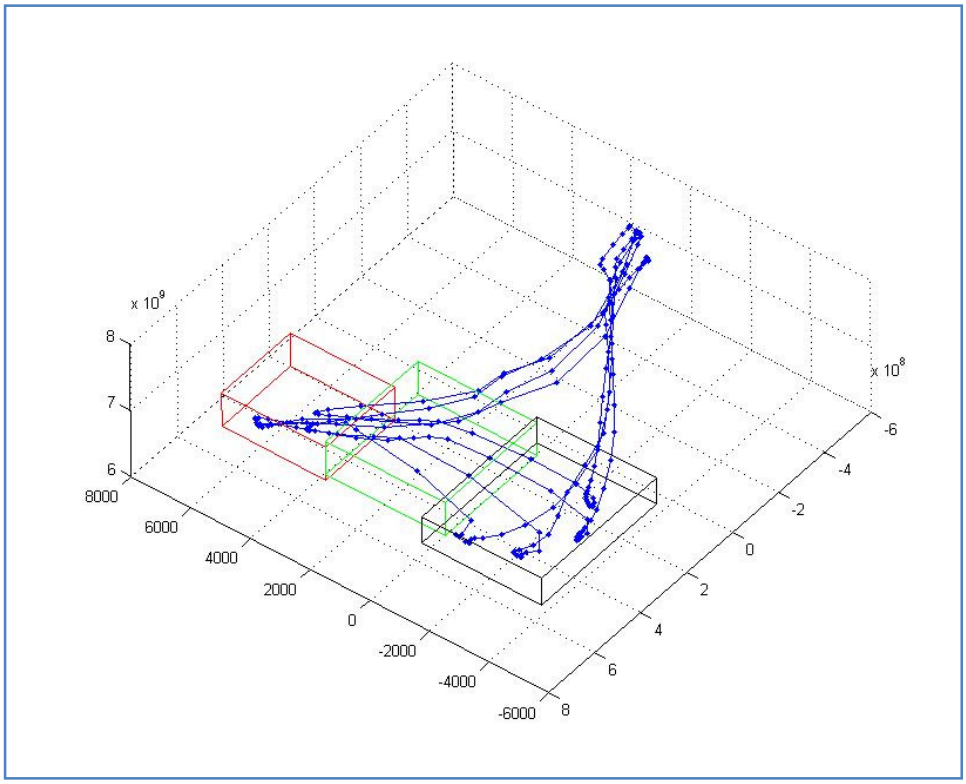
Graph 7 shows the shifting of the movements in the eigenspace using different light conditions. In our recognition system these changes in the eigenspace would have serious impact on the recognition. The next subsection explains the identification of the movement shapes in the graphs is solved.



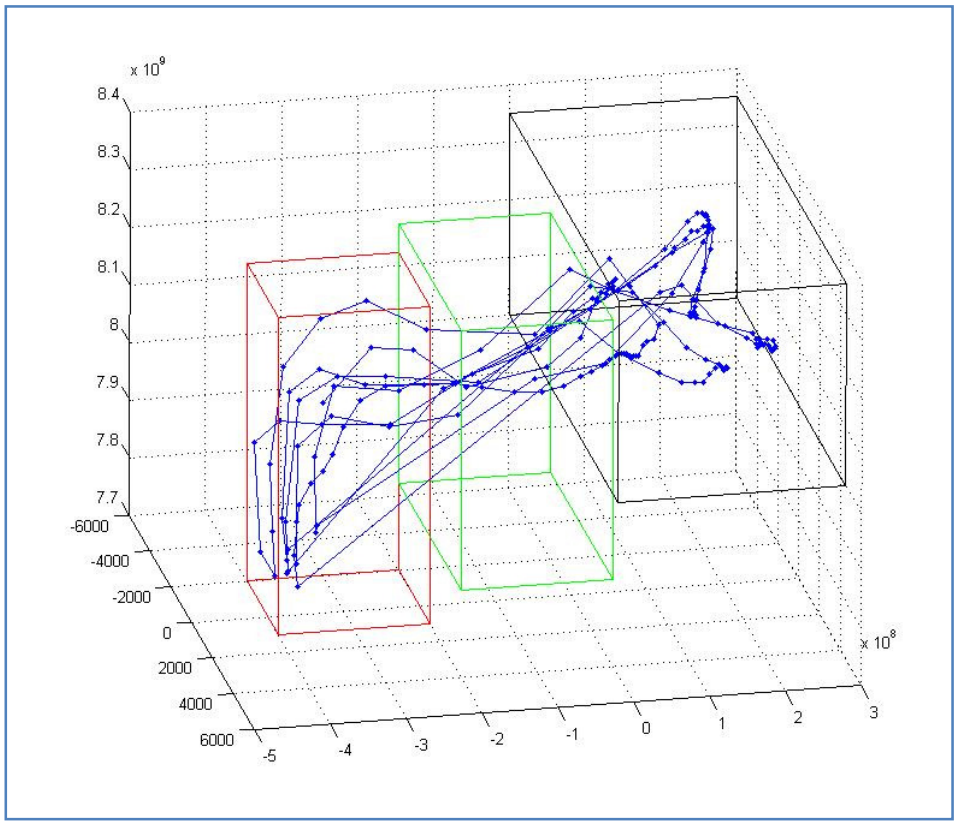
Graph 731: Sign "allow" using different light conditions. ***Different light conditions show each sign in the same shape but in a different position in the space.***

5.2.4 Observed Symbols

One step of the recognition is to identify uniquely the trajectory of the gestures which are represented in these graphs. An HMM will be used for the recognition of those trajectories. Therefore, one of the pre-processing steps of an HMM is to define how many symbols are needed to recognize the object. These symbols are called observed symbols. Each trajectory will be segmented into sections which visually form those symbols. The eigenspace coordinates of each segmented section of that specific gesture allow the observed symbols to be computed. The Graphs, 8 and 9, give examples of the observed symbols, drawn as boxes, which are used to identify the trajectory of a gesture. Only those parts which are included in the boxes belong to the sign, the other parts belong to the neutral position as mentioned before.



Graph 8: Sign "man" including the observed symbols. ***The sign starts in the area of the red box and ends in the area of the black box. Each box represent an observed symbol.***



Graph 9: Sign "adult" including the observed symbols. ***The sign starts in the area of the red box and ends in the area of the black box. Each box represent an observed symbol.***

The red box is always the start symbol and the black box is the end symbol of the sign. The number of symbols can change for each sign but it should be kept small. If the shape of the sign is divided into too many symbols, the recognition of new samples could be difficult in the case that new data includes a lot of noise. The boxes should not overlap so that each frame can never belong to more than one symbol.

A small program was implemented to convert the coordinates of each frame into observed symbols. The result is a list of numbers which represent the observed symbols. If the coordinates of a frame lie within one of the boxes the number of the symbol will be displayed, otherwise if it does not fit at all the number '5' appears

which means it does not fit in any of these defined symbols. Therefore the boxes should not overlap or have space between them.

The following sequence is an example of the observed symbols calculated for one gesture.

[1 5 5 5 5 5 5 2 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 5 5

5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5]

This list was calculated for one example of the sign “man” represented in its own eigenspace. The frames which do not belong to the actual sign will be expressed as symbol “5”. Every PC value of the new data, projected in the eigenspace, fits into any of the observed symbols is indicated with the number of the observed state. The red box, marked in Graphs, 8 and 9, labels the starting position of the sign. Therefore, if a value lies in this box, the observed symbol is defined as ‘1’. As soon as the values lie in the green box which is the second step of the sign, the observed symbols are marked with a ‘2’. The third step is marked with a ‘3’. Different signs can have a different number of observed symbols. In my case, I identified signs with two or three observed symbols. This sequence is used afterwards as input to the HMM.

When a new unknown gesture arrives, the observed symbols will be calculated from the new data in each eigenspace. The more different the new gesture is, the more “5”s will be displayed in the list. The more alike the gestures are, the more alike will be the observed symbols. Therefore we need the HMM to differentiate the gestures which are very alike. The next subsection demonstrates that there are still some movement which cannot be identified through this method.

5.2.5 Exceptions

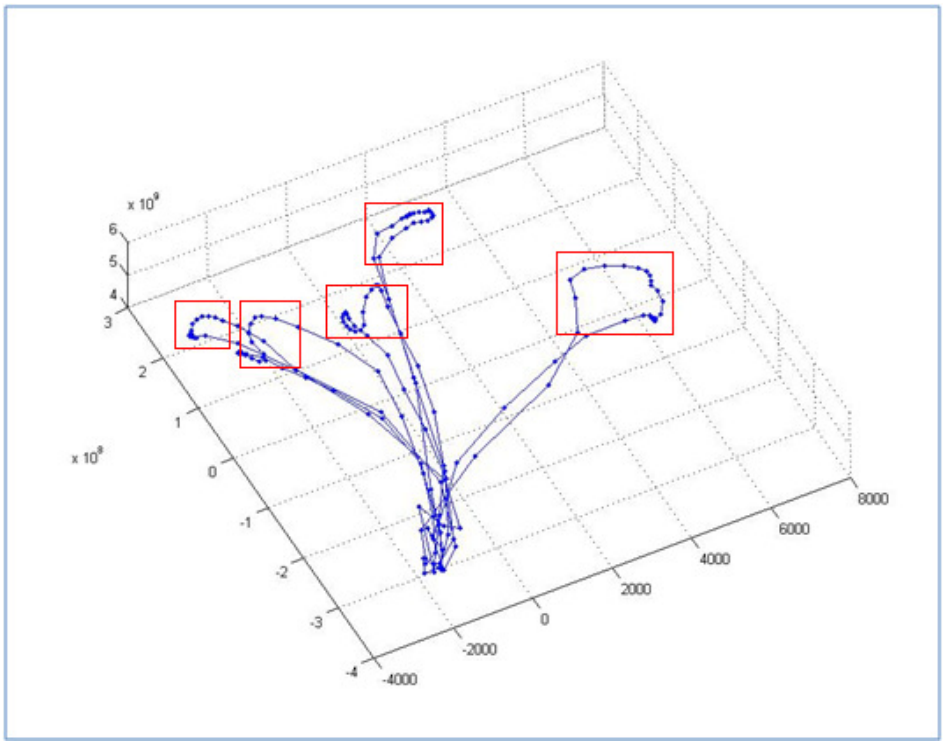
In some cases it was not possible to identify a sign by its trajectory using this method. The following two graphs show two examples in which it was not possible to identify the sign with the current method we use.

a. No movement

The following three frames in Figure 30 are taken from three different examples of the sign “home” including the variations. The only difference of the sign is the position of the hands in front of the body. This example can be found on the CD.

Figure 30: Sign "home". ***The sign does not contain any movement, these frames show different samples of this sign.***

Graph 10 shows the sign “home” from Figure 30, is represented in the same eigenspace. The parts which are marked in red are the actual signs and the rest are transitions to and from the signs. In this case, the sign is not only represented in different areas of the eigenspace but forms each time a different trajectory as well. This is the result of a sign which does not include movements in the gesture but the sign is shown using a hand position only.



Graph 10: Sign "home". ***The shapes are in different position in the space and each shape is different.***

b. Alternate hand movement

Figure 31 shows four frames representing an instance of one example of the sign "balance". In this sign both hands move in opposite direction.

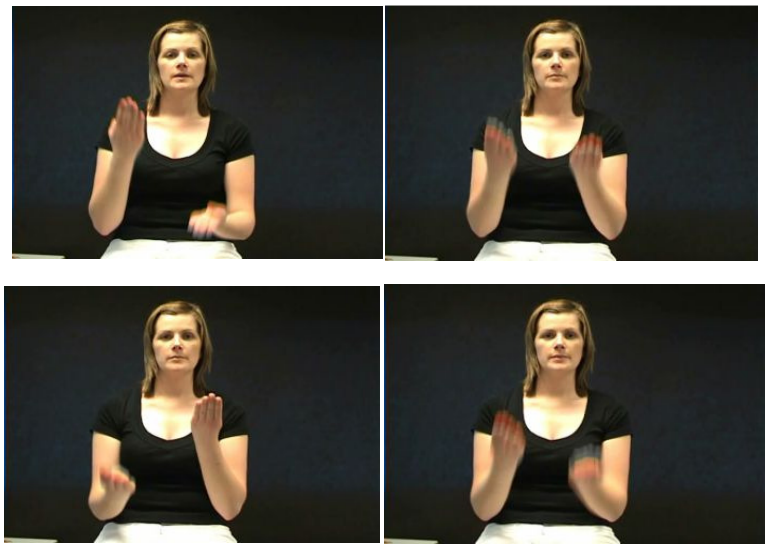
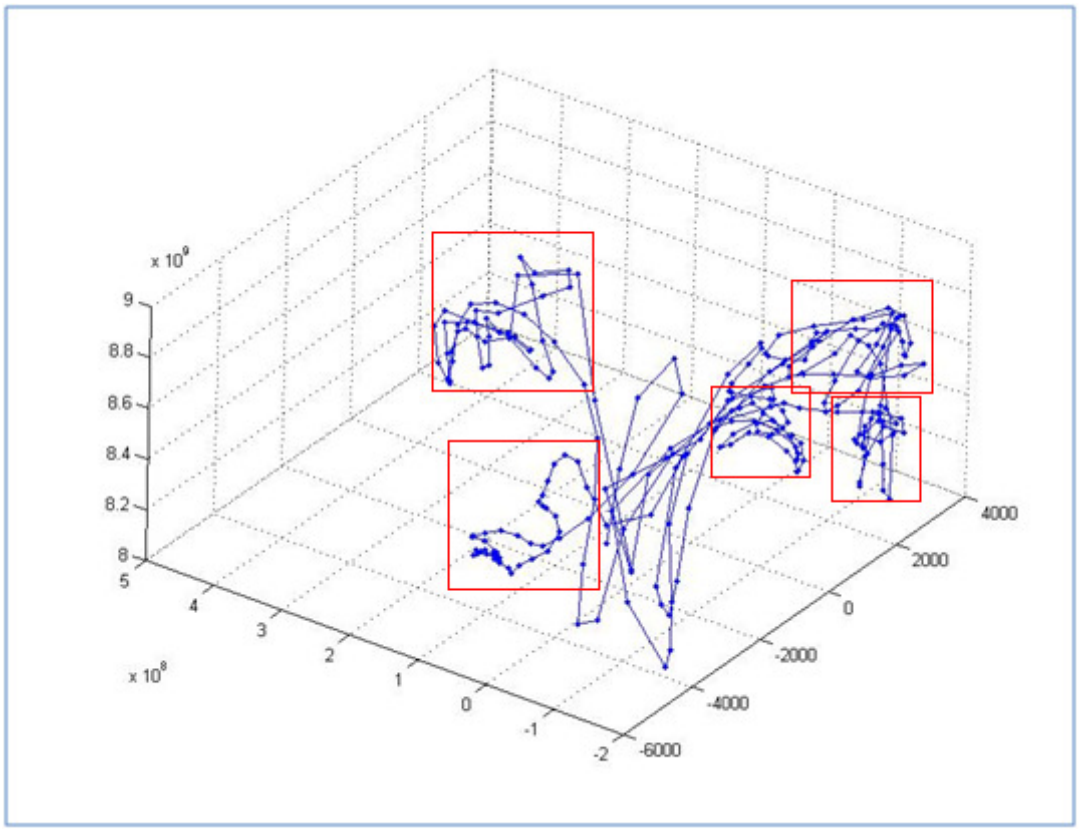


Figure 31: Sign "balance" of real-time video. ***The four frames are taken from one sample of the sign to show that both hands moving in opposite direction.***

In Graph 11 the sign “balance” which is shown in Figure 31, is represented. In this example each red box marked one example of the sign. In this case, the sign is represented in different areas of the eigenspace and forms a different trajectory each time. Therefore the sign cannot be identified through the method of creating boxes. This effect appears if the two hands make the same movement but in the opposite direction to each other. A possible solution for this problem could be found using a combination of HMM and DBN. Additionally, it has to be considered how often this problem appears after using all the other features of the classification system in consideration.



Graph 11: Sign "balance". ***In the sign, both hands moving in oposite directions, therefore the shapes of each sign is different to the shape of the other samples.***

5.2.6 Hidden Markov Model – Bayesian Network

The recognition in this system is processed using HMMs which are explained in section 2.3.3. HMMs are a very successful recognition method and are used in many areas such as speech, handwriting, gesture or sign language recognition. In general, an HMM is a tool for representing the probability distributions of observed sequences. Different types of HMM exist, which are differentiated from each other mostly by their input and output features. In this case we use a simple discrete HMM. This means that the observed symbols will be represented using discrete numbers.

A very important point is that all the Markov models are stochastic processes and use the Markov property which implies that every state depends on its previous state.

5.2.6.1 Hidden Markov Model

An HMM is a collection of states, a combination of observed symbols and hidden states, connected by transitions. An HMM can be considered as the simplest Dynamic Bayesian Network.

An HMM is characterized by a number N of hidden states S_N and a number M of observed symbols O_M which are all connected with two different type of transitions: the state transition probability and the emission probability. The system is in one of the hidden states at any given time. If the system is in state i it may move to another state j with transition probability $a(i,j)$. The emission probabilities contain the probabilities of a particular observed symbol being emitted by a given hidden state.

An HMM has the ability to calculate the likelihood of a new sequence of observed data. The *likelihood* is a term used in statistics and defined as the "probability of the

data given the model". Therefore I created a small HMM for each gesture I want to recognize. The results of the HMM for all the gestures in the vocabulary are then compared and the HMM with the largest likelihood is defined as the recognized sign. Figure 32 gives an overview of the system. For each sign an individual HMM is created and trained. All the HMMs are connected to each other.

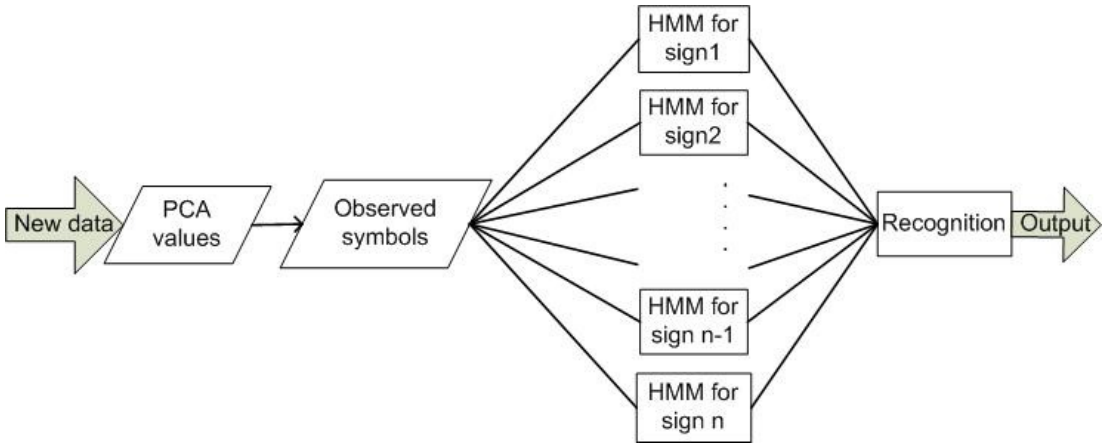


Figure 32: Recognition System Overview. ***Each step of the recognition process is shown, starting by calculating the PCA values and the observed symbols and followed by puting these symbols in each HMM model to recognize them.***

During the training, the HMMs create their connections and calculate the transition and emission probabilities between the states. Table 4 shows two examples of the transition and emission probabilities of two signs: "adult" and "man". These matrices were created during training by the HMM itself. Each row has to the sum to 1.

sign "Adult"			
transition matrix			
0.7741	0.2073	0.0000	0.0186
0.0000	0.0287	0.0000	0.9713
0.0414	0.0000	0.9586	0.0000
0.0000	0.0368	0.1223	0.8409
emission matrix			
0.0000	0.0000	0.0000	1.000
0.9989	0.0000	0.0000	0.0011
0.0000	0.2367	0.7633	0.0000
1.000	0.0000	0.0000	0.0000

sign "Man"			
transition matrix			
0.7705	0.1148	0.0950	0.0197
0.0579	0.9421	0.0000	0.0000
0.0719	0.0000	0.3931	0.5350
0.1034	0.0000	0.8729	0.0237
emission matrix			
0.0000	0.4265	0.0000	0.5735
0.0000	0.0000	1.000	0.0000
1.000	0.0000	0.0000	0.0000
1.000	0.0000	0.0000	0.0000

Table 4: Probabilities of the sign "adult" and "man". ***The transition matrix represents the probablities between the connections of the hidden states and the emission matrix represents the probabilities between the hidden states and the observed states.***

Figures, 33 and 34, are a visual representation of the matrices above. This gives an overview on the hidden states and their transition probabilities and the connection to the observed symbol.

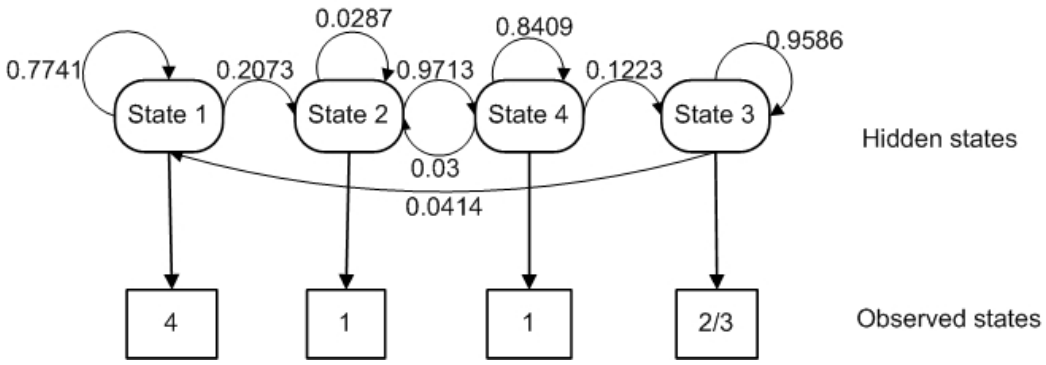


Figure 33: HMM structure of the sign "adult". ***The HMM is represented with four hidden states and four observed states including the connections between the hidden states and their probabilities.***

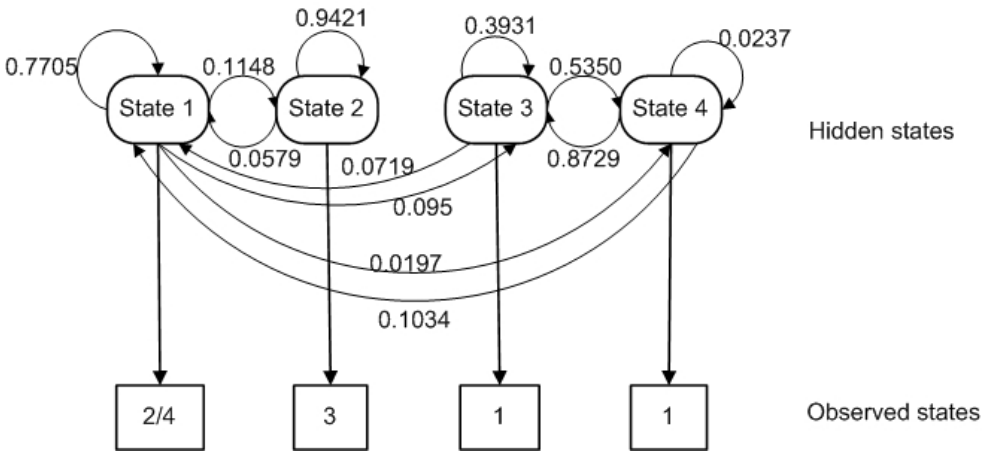


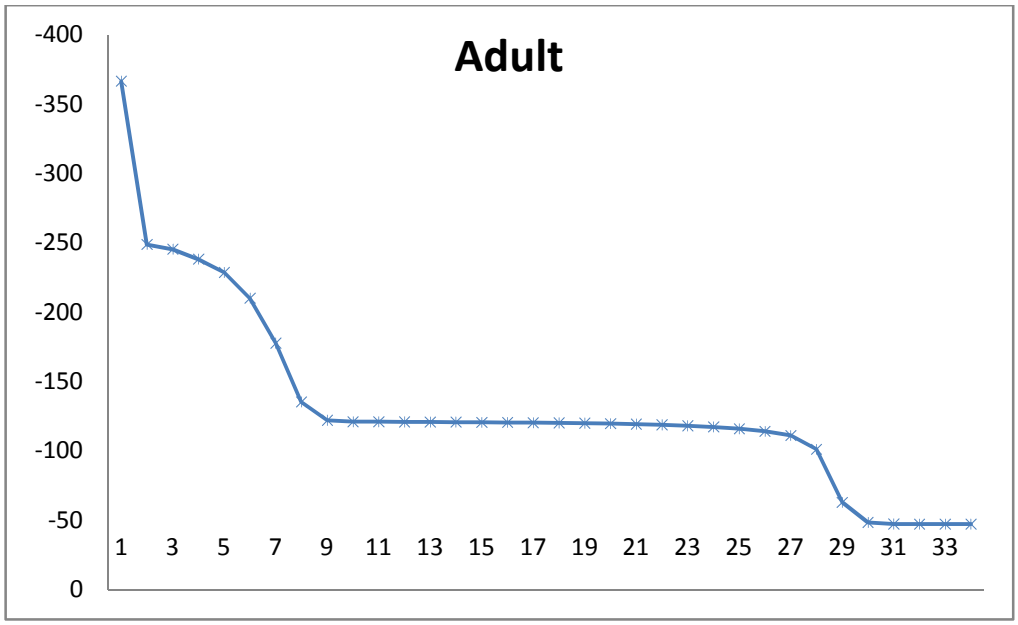
Figure 34: HMM structure of the sign "man". ***The HMM is represented with four hidden states and four observed states including the connections between the hidden states and their probabilities.***

5.2.6.2 Training

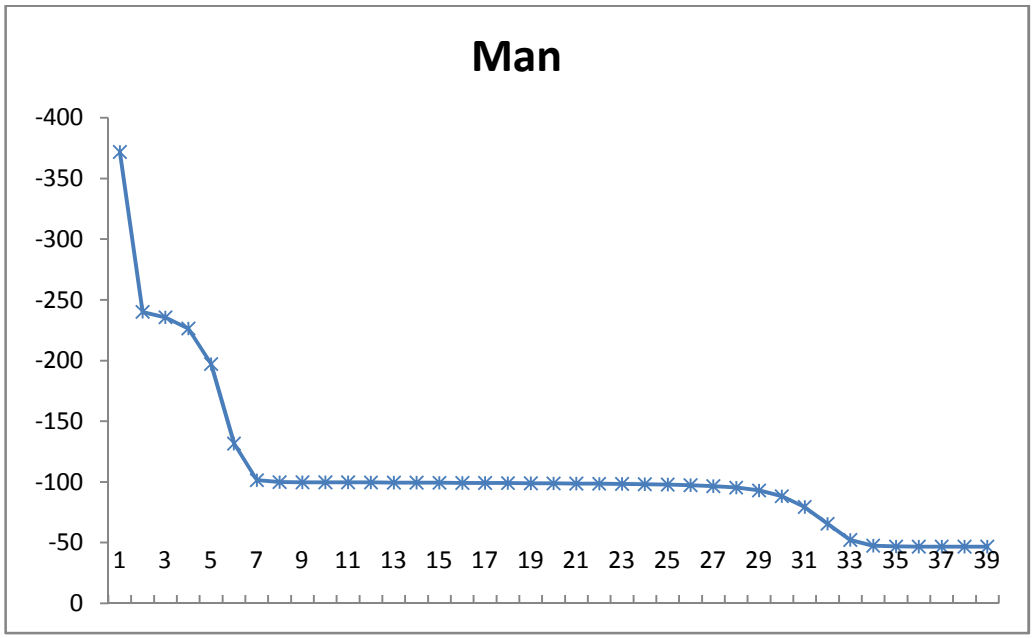
Training is the key part of the HMM. During training, the structure of the states connected with the probabilities will be generated. Each training process forms a new structure of the states and their probabilities. Therefore if the system is well trained and the recognition rate is good, the training information will be stored and applied to each recognition process. Each sign has six samples in the training set.

As explained in section 2.3.3 the HMM can be trained using the Baum-Welch algorithm. The output of the training is the log-likelihood which is in Matlab called *loglik*.

Graphs, 12 and 13, represent the loglik which is calculated during the training. The horizontal axis gives the number of iterations of the Baum-Welch algorithm. The closer the loglik gets to 0 the better the HMM is trained.



Graph 12: Training output of sign "adult". ***The training iterations are shown, the closer to 0 the better the HMM is trained.***



Graph 13: Training output of sign "man". ***The training interactions are shown and stops around the value 50 which means this HMM cannot be trained better.***

5.2.6.3 Results

The HMM was tested with 12 signs of the real-time videos and each sign had four examples which included variations in the gestures, totalling 48 examples. Of these examples the HMM could recognize 41 signs correctly, therefore the system has a recognition rate of 85.41%. This results, that some signs, if they are projected in other eigenspace, they have a very similar shape as the trained data. There were no synthetic videos used for training and recognition because these do not have natural human motion. To improve the recognition rate, a distance metric is introduced in the next section.

5.2.7 Distance Metric

Recognising movements using PCA and HMM is very efficient for a few signs with big difference in the movements. When it comes to a large number, hundreds of different signs, another method has to be introduced and combined with HMM because the difference between some signs is too small for the HMM. Therefore I introduced a distance metric. The idea to use this method arises from the work of Coogan (2007). He used this on static hand shape recognition. In my case, the images of the data are represented in the original full-dimensional space and the perpendicular distance of each full-dimensional point to a given eigenspace will be calculated.

The perpendicular distance (D_p) will be calculated as followed:

$$D_p^2 = D_e^2 - \sum_{i=1} [(p - o) * u_i]^2$$

With D_e = Euclidean distance between point p and the origin o of the eigenspace E, and the u_i are the eigenvectors of the eigenspace.

Perpendicular distance (D_p) to a point p to a given eigenspace E is shown in Figure 35 below.

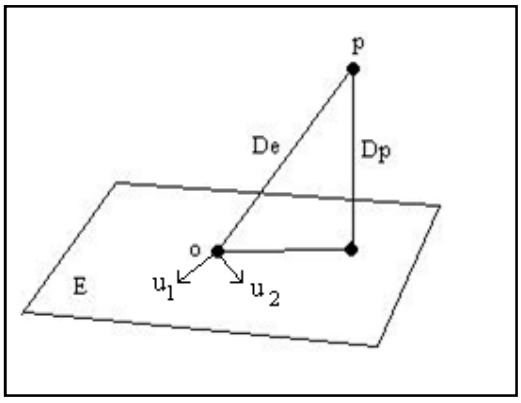


Figure 35: Distance to the eigenspace. The *perpendicular distance from a point 'p' to a eigenspace 'E' is shown.*

The distance is calculated for each eigenspace stored in the system. I use this technique to improve the recognition system.

Table 5 shows the distance between each sign and the eigenspaces of all the stored signs. The results in the table 5 show that distances which are larger than 10,000 do not need to be considered. I made this conclusion out of the results shown in the following table because if the distance has a higher value than 10,000 the eigenspace is too far away from the new data which makes it improbable that the new data and the compared trained sign would be similar to each other.

from \ to	Adult	Allow	Begin	Black	Body	Easter	End	Lady	Man	Morning	Tomorrow
Adult	2811	11443	15038	16391	15618	14427	14931	17428	16473	15743	15911
Allow	10469	2952	13230	17164	14379	16323	15835	16843	17147	16690	16988
Begin	15408	13570	3699	15984	11975	15799	15323	15223	18328	18046	15604
Black	14880	15047	13459	4402	13847	11390	11753	12663	14158	13673	10847
Body	14862	14342	10948	15125	2593	15255	14904	14140	16365	16047	14566
Easter	12925	13631	12466	9940	13453	2703	6271	12941	13008	12458	11276
End	12526	13335	12679	11931	13619	8278	3635	12890	12846	12548	12058
Lady	13508	13410	11633	10953	10995	12275	11860	2527	12518	12222	11817
Man	14156	15068	14846	14071	14978	12533	12634	13826	2529	6415	14555
Morning	13947	14978	14712	14531	14547	13789	13549	13473	6241	2843	14422
Tomorrow	14407	15247	13437	11051	14041	12586	12813	13600	15592	15300	2795

Table 5 : Perpendicular distance results. ***The more the signs are different form each other the bigger the perpendicualr distace between the sign and the eigenspace***

Comparing the sign “adult” and “allow”, the two signs are very different from each other. The only similarity is the hand shape, both signs use the hand shape ‘a’. The sign “adult” moves the hand from down up, unlike the sign “allow” which move the hand from the face away from the body. Calculating the perpendicular distance, from the sign “adult” to the eigenspace of the sign “allow”, results in 11443 units. This shows that the two signs are very different from each other.

Comparing the signs “man” and “morning”, both signs use the same hand shape and the same movement at the same position in front of the body. The only difference is that the motions of the two signs are made in the opposite direction. While signing “man”, the hands are moving downward and while signing “morning”, the hands are moving upwards. For this reason, the two eigenspace lie very close to each other in the global space. Therefore if I calculate the perpendicular distance between one of these signs and the other sign eigenspace, the distance is below 10,000.

Only for those data where the distance is lower than 10,000 will the data be projected onto those eigenspaces and the trajectory will be recognized using the HMM.

5.3 Results

Table 6 illustrates the result for the first example of the sign “Adult” and the third example of the sign “Easter”. These examples where taken from those mentioned in section 5.2.3.3. First, the distance between the new data in the original full-dimensional space and every stored eigenspace was calculated. Afterwards, in general, the loglik is been calculated from only those where the distance was less than 10,000. To give a better overview, all the loglik (Korb & Nicholson, 2003) were

calculated. As mentioned before, the smaller the distance the closer the new data is in the global space to the trained data eigenspace. Additionally, the closer the loglik is to 0, the more similar the two compared data are.

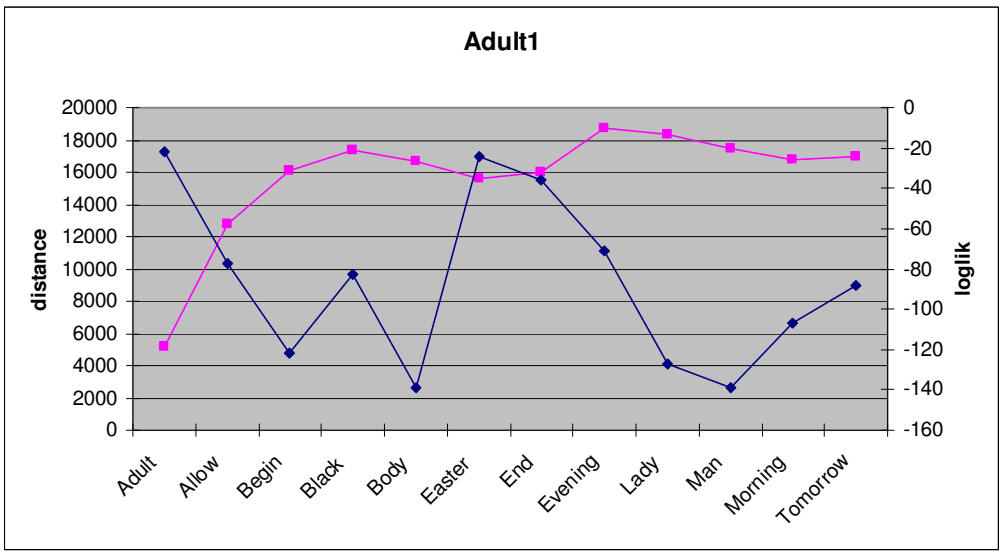
As shown in the left table, only the distance of the sign “Adult” is below 10,000 and in that case there would be only one loglik calculated. In the right table the distance of “Easter” and “End” are smaller than 10000. In this case the loglik of “Easter” and “End” would be calculated. This case is a perfect example where the HMM does not work correctly because the signs looks very similar to each other. Therefore the perpendicular distance is in two cases less than 10,000: in the case of “Easter” and “End”. The HMM values are closer to 0 in comparison with the sign “End”, therefore the recognition in this case would not be correct.

Adult1	distance	Loglik
Adult	5152.75	-22
Allow	12744.58	-77
Begin	16116.99	-122
Black	17398.95	-83
Body	16680.45	-139
Easter	15572.08	-24
End	16033.24	-36
Evening	18768.73	-71
Lady	18364.54	-127
Man	17476.43	-139
Morning	16759.51	-107
Tomorrow	16942.2	-88

Easter3	distance	loglik
Adult	14797.85	-200
Allow	15676.2	-94
Begin	15054.37	-151
Black	12274.4	-102
Body	15558.91	-171
Easter	6503.982	-84
End	9886.999	-75
Evening	14281.51	-116
Lady	15055.63	-157
Man	14975.31	-171
Morning	14563.35	-131
Tomorrow	13485.56	-108

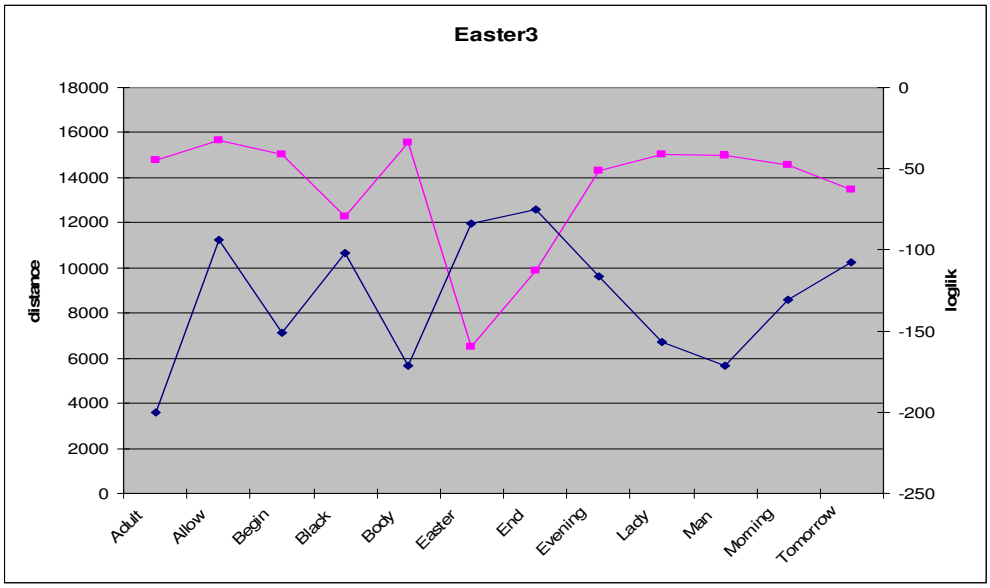
Table 6 : System results of the sign "adult" and "easter". ***The distance and the loglik are shown to demonstrate all the recognition results.The left table shows the results of the sign “adult” which is perfectly recognized. The right table shows the results of the sign “easter” which is the only sample which could not be recognized correctly.***

These two Graphs, 14 and 15, are a visual representation of the results above. The pink line represents the distance and the blue line the loglik. The first Graph 14 is a good example on how well the combination of the HMM and distance metric works. In Graph 14 of “Adult”, the loglik of two signs could match but the distance metric exclude the second loglik before it would be calculated and results in only one match.



Graph 14: System results of the sign "adult". ***The blue line represents the loglik and the pink line represents the distance results.***

Graph 15 shows that the system is not 100% efficient. The result should be the sign "Easter" and with the two methods (distance and HMM) combined, it results in the sign "End". The results of other tested signs can be found in Appendix C.



Graph 15 : System results of the sign "easter". ***The blue line represents the loglik and the pink line represents the distance results.***

The system was tested with 12 individual signs and each sign appeared four times with variation. As mentioned before, testing the data with only HMM the recognition rate is 85.41%. However, testing the data on the full system, out of those 48 test data only one sign was not correctly identified which is shown above as an example. In total the system has now a recognition rate of 97.91%. The system is more accurate using the combination of those two methods. Taken into consideration, the recognition is based on one phoneme of the classification system. However, this recognition system proves that the different movements can be recognized using HMM and distance metric.

5.4 Summary

This chapter introduced two developed systems. In the first section a description of a classification system is provided. This classification system which is addressed in the third statement of my contributions in section 1.4 is based on phonemes: hand shape, movement, hand position in the signing space and the hand orientation in the space. The classification system is very accurate because it can identify uniquely every sign of the created real-time video.

In the second section a gesture recognition system is described. It is very efficient to use PCA to reduce the dimension of the data and to represent the data in a 3D space which is addressed in the fourth statement in my contributions. The real-time videos were recorded applying two main conditions. The background in the video has to be black and the signer has to wear a black shirt. Using these criteria, the only changes from one frame to the next one are the movement in the sign. Therefore I was able to represent the movements of each sign in its own eigenspace. Additionally to the real-time video I generated synthetic videos. At the moment the real videos and synthetic videos are not commensurate. This is because the synthetic avatar has a very different appearance from the real signer and the lighting conditions and camera parameters are different. Therefore it is not possible to compare real and synthetic videos in the same eigenspace. However, I was able to use the synthetic videos to analyse several conditions such as different camera positions and light conditions. The synthetic videos perform the shapes smoother in the graphs, which enables better analysis of variations.

Using this method I was able to create observed symbols of the shape in each eigenspace. These observed symbols are used as input data in each small HMM. Out

of the results I conclude that HMM turned out well. The recognition results can be improved using a distance metric method. Calculating the distance of the new data in the global space to each eigenspace improves the recognition rate. Developing the recognition system using HMM is addressed in the fifth statement of my contributions in section 1.4.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

The aim of this thesis is to generate a gesture recognition system which can recognize several signs of ISL. Chapter 1 starts with an overview of the problems and listed the difficulties in developing a gesture recognition system for ISL and ends with explaining the main contributions of this thesis. Most of these difficulties are addressed and solved. Those which could not be solved, involve areas such as NMFs which were not investigated in this thesis.

Chapter 2 introduced previous work including developed techniques used in gesture recognition. As shown in this thesis, the recognition is always built up of first, feature extraction and second, the recognition. Several methods were pointed out for feature extraction such as skin detection, motion capturing using animated videos, and PCA. I made the decision that I will not use skin detection because it is too complex at this stage of the system but I use hand and face detection by extracting objects to detect the signing space. Additionally, PCA was proven as a very robust and useful method to reduce high-dimensional data into low-dimension and represent the data visually in a new coordination system. Also, using HMM for gesture recognition was described as very efficient. However, using complex gestures and a large vocabulary such as 1000 signs, the HMM is not sufficient. To solve this problem Suk *et al*. (2008) suggests a combination of HMM and DBN.

The linguistic background in Chapter 3, provides background knowledge on ISL. It is important to understand the basis of a language which is analysed and recognized. ISL is very different in its structure to spoken language. Different features such as NMF have to be involved in identifying signs. This chapter describes difficulties which can arise or which the developer has to be aware of, for the generation of a recognition system. Additionally it explains previously developed notation system used in ASL. These systems were developed to write the sign language in symbols. The understanding of these systems helped me to develop my own classification system.

Chapter 4 described the data which is used in the system. For this system, two different kinds of data were created: synthetic and real-time videos. Initially the data to analyse the recognition system, is generated in a 3D animation package in order to simplify the creation of variation of the signer movement and his environment. The implementation in the animation environment allows the generation of different versions of the same gesture with slight variations in the parameters of the motion. Using this animation tool to explore variation within the gesture is a very effective way to model translation, rotation, illumination etc. As soon as the gestures and each variation are implemented, a lot of new data can be generated with only a few clicks. Using synthetic videos is addressed as the second statement of my contributions in section 1.4.

The synthetic data is very useful in analysing various impacts on the system because it does not contain noise. This means the shapes in the graphs, shown in section

5.2.2, have the ideal shape for that situation and the conclusions can therefore be used on real-time videos.

Another component of the training data is a real-time video database which addresses the first statement of my contributions. I produced a database of ISL in real-time video. The database contains 1400 different signs, including each sign in several variations, such as variations in the signers' movement. This is very helpful because the impact of variations in the movement can be analysed and the system can be trained on them. One negative aspect is that only one signer is used in the database. It would benefit the approach, to expand the database of real-time video with more signs which includes the variations and using different signers for analysing and testing the system. Different signers have an impact on the system, using a database with different signers would benefit in the way that the influence of several signers could be analysed.

Chapter 5 describes two developed systems. The first part, focused on the development of a classification system, upon which the recognition system is built. This classification system is based on the linguistic background knowledge of ISL and is able to identify each sign of the real-time video database. One big disadvantage of this approach is that the NMFs are not included and therefore not every existing sign in ISL can be classified uniquely. This classification is addressed as the third statement in my contribution in Chapter 1.

Further in section 5.2, a method for an automatic detection of the signing space was developed. Up to now, this basic method cannot yet be applied because the system is too elementary. This means that currently the region of interest in the images has to

be the same size. This fixed size affects the approach in real-life situations with different possible positions of the signer in front of the camera.

Until now, only one step of the classification model, the movement, has been implemented as an HMM and used to recognize a limited set of signs. PCA, provides an economical way of representing a set of gestures and using PCA to represent the motions in this approach is very effective. PCA allows me to represent movement of a 2D video in a 3D graph. These representations enable me to analyse the impact of the system which is addressed as the fourth statement in my contributions. An advantage of PCA is that it presents every small change in an image sequence. This is a benefit if the sign involves only small movements which in this case can still be represented. On the other hand, it can also be a negative effect because every other small changes such as in the environment will influence the representation.

Additionally, the observed symbols, described in section 5.2.2, are defined by the trajectories in the PCA spaces. Until now, the observed symbols are defined as rectangular boxes. This method is acceptable for the amount of signs I am using at the moment. Using more signs, this method will not be able to be applied on more complex trajectories.

Based on this representation, I use a discrete HMM, for recognising each sign which is addressed as the fifth statement of my contributions. The HMM was trained using the Baum-Welch algorithm, and I use the HMM to represent the temporal variation within a gesture. All of them are connected to one large system, trained and tested with real-time videos of our database. For the small set of signs which can be recognized until now, this was a very good choice but for a larger set of signs, the

discrete HMM is not good enough. This problem could be solved using a mixture of HMM and Bayesian Networks.

Additional to the HMM, I used a distance metric which had previously been used for static signing. This method works by calculating the perpendicular distance between the new data in the global space and each defined eigenspace. This combination of distance calculation and HMM works very efficiently and enabled us to recognize more signs with a high recognition rate 97.91%. 47 out of 48 signs were recognized correctly. Half of the 48 signs were taken from the training set to prove that the HMM models were generated correctly and the other half is new data to see how well the system recognize unknown samples. The third sample of the sign “easter” was not recognized correctly. This is possible because the sign “easter” and the sign “end” are very similar. Both have the same hand shape and the same movement. This sign was chosen to show that a sign cannot only be recognized by its hand shape and movement.

There is a lot of work still to be done, such as developing the other parts of the classification system and including the NMF, for a full ISL recognition system but this work can be seen as a good foundation on which future work can be built on.

6.2 Future Work

The aim of sign language recognition is the provision of an interactive gesture recognition system. This project focuses on the development of a *proof of concept* ISL recognition system which incorporates the core principles and challenges inherent in a generic gesture recognition system. Future work on this project can be to develop a more complex gesture recognition system which can recognise ISL

fully. This project has built a foundation which is a simple classification system upon which my recognition system is based.

Future work could initially focus on researching new methods in skin and hand detection, variation of signing space and depth detection. The support for skin and hand detection will facilitate advanced motion tracking on a variety of human subjects under various lighting conditions. Furthermore, there is a requirement to investigate the provision of depth detection to enable background filtering so as to isolate and focus on the person alone. The additional distance information may be exploited by our recognition algorithm allowing objects to be tracked, identified and separated according to distance from the camera. The ability to slice a 3D environment into depth-wise “slices” would permit improved gesture recognition against arbitrary backgrounds which are the norm in real world environments.

An important component of this research project involves system analysis with both synthetic and real-time video data. More initial data has to be created in a 3D animation package which allows generating synthetic videos including variation parameters in the environment such as camera positions, light angles and background textures. This will be supplemented with large volumes of real-time video data. Similarly, the real-time videos have to be adjusted as well. Therefore more real-time video have to be created which include more samples of the existing signs and using different signers. This enables a better analysing and testing of the recognition system.

A full gesture recognition system could be adjusted with small changes for various situations. A significant contribution to the availability of sign language recognition software on the mobile phone would greatly enhance the quality of life for the Deaf

community because they could communicate with Hearing people using the camera and the converter program. Additionally, the ability to interact via gestures in the more general sense could aid each person in their interactions with computers, mobile devices and ultimately with their environment. It would also support the creation of gesture-based user interfaces for mobile phones, which will be easier to use than the current keypads. For example, older people can have problems using the small keys on a mobile phone, it could be easier for them to use simple gestures to interact with the phone. This should enable the use of virtual reality immersive applications. For the realisation of a mobile phone application, the program has to be small which means not using too much capacity from the mobile phone that would slow them down and it has to perform gesture recognition efficiently.

It is envisaged to continue this work and support the system with the ability to recognise a large vocabulary of signs, including a robust recognition of gesture variations (i.e. change in angles, rotation of movements, and scaling) and thus ensuring that variations to the camera position will have minimal impact on the classification. A primary design goal for all sign language recognition will be to ensure the program works on a standard personal computer with a normal web cam or on a mobile device and that it can adapt to a new user after a short training session.

Bibliography

Argyros, A.A. and Lourakis, M.I.A. (2004). Real Time Tracking of Multiple Skin-Coloured Objects with Possible Moving Cameras. *In Proceedings of the European Conference on computer vision.* pp.368-379. Prague, Czech Republic.

Awad, G. (2007). *A Framework for Sign Language Recognition using Support Vector Machines and Active Learning for Skin Segmentation and Boosted Temporal Sub-units*. PhD Thesis, Dublin City University. Dublin, Ireland.

Boyes, C. (2005). *Body Language: The Secret Language of Gestures and Postures Revealed*. London: HarperCollins.

Boyle, R.D. *Hidden Markov Model*. [online]. [Accessed 25 October 2008]. Available from World Wide Web: http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html

Brown, D., Craw, I., and Lewthwaite, J. (2001). A som based approach to skin detection with application in real time systems. *In Proceedings of the British Machine Vision Conference*.

Bunke, H. and Caelli, T. (2001). *Hidden Markov Models, Applications in computer vision*. World Scientific Publishing.

Chen, C. and Wang P.S. (1993). *Handbook of pattern recognition and computer vision*. World Scientific Publishing.

Coogan, T.A. (2007). *Dynamic Gesture recognition using transformation invariant hand shape recognition*. MSc. Thesis, Dublin City University. Dublin, Ireland.

Cox, S. (2009). *Lip-reading computer picks out your language*. [online]. [Accessed 29 April 2009]. Available from World Wide Web: http://www.newscientist.com/article/mg20227055.800-lipreading-computer-picks-out-your-language.html

Foran, S.J. (1996). *Irish Sign Language*.

Gemmar, P., Gronz, O., Heinrichs, T. and F. Hertel, F. (2008). Advanced Methods for Target Navigation using Microelectrode Recordings in Stereotactic Deep Brain Stimulation. *In Proceedings of 21st Institute of Electrical and Electronics Engineers International Symposium on Computer-Based Medical Systems*. pp.99-104. Jyväskylä, Finland.

Grobel, K. and Assan, M. (1997). Isolated Sign Language Recognition using Hidden Markov Model. *In Proceedings of the International Conferences on Systems, Man and Cybernetics.* pp.162-167. Orlando, Florida.

Hipp, C. and Gumhold, M.. (2002). *Motion Capture*. Ulm, Germany.

Huang, T.S. and Pavolic, V.I. (1995). Hand Gesture Modeling, Analysis and Synthesis. *In Proceedings of the International Workshop on Automatic Face and Gesture Recognition*., pp.73-79. Zurich, Switzerland

Huang, T.S. and Wu, Y. (1999). *Vision-Based Gesture Recognition: A Review in Gesture-Based Communication in Human-Computer Interaction*, Lecture Notes in Computer Science, **1739** pp.103 - 115. London, UK

Hutchins, S., Poizner, H., Mcintire, M. and Newkirk, D. (1987). Implications for Sign Research of Computerized Written Form of ASL. *In Proceedings of the Fourth International Symposium on Sign Language Research*. Hamburg: Signum Verlag, pp.255-268. Lappenranta, Finnland.

Impedovo, S., Ottaviano, L. and Occhinegro, S. (1991). Optical Character Recognition - A Survey. *In Proceedings of the International Journal of Pattern Recognition and Artificial Intelligence*. **5**(1-2), pp.1-24.

Iwai, Y., Watanabe, K., Yagi, Y. and Yachida, M. (1996). Gesture Recognition using Colored Gloves. *In*: *Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96)*. **1-7270** p.662. Washington, DC.

Kadous, M.W. (1996). Machine Recognition of Auslan Signs using PowerGloves: Towards Large-Lexicon of Sign Language. *In Proceedings of the Workshop in Intergration of Gesture in Language and Speech*., pp.165-174. Wilmington, Delaware, USA.

Kang, H., Lee, C.W., and Jung, K. (2004). Recognition-Based Gesture Spotting in Video Games. *Patter Recognition Letter*. **25**(15), pp.1701-1714. New York, USA.

Kelly, D. and McDonald, J. (2008). System for Teaching Sign Language using Live Gesture Feedback. *In Proceedings of the 8th International Conference on Automatic Face and Gesture Recognition*. Amsterdam, Netherlands.

Korb, K.B. and Nicholson, A.E. (2003). *Bayesian Artificial Intelligence*. Chapman & Hall/CRC Press UK.

Luukka, P. (2009). PCA for Fuzzy Data and Similarity Classifier in Building Recognition System for Post-Operative Patient Data. *An International Journal on Expert System with Apllications.* **36**(2), pp.1222-1228. Tarrytown, NY, USA.

Matt, J.. (2002). *Tracking und Modifizieren Menschlicher Oberkörperbewegungen mit Dynamischer Simulation*.

Ó'Baoill, D. and Matthews, P.A. (2000). *The Irish Deaf Community (Volume 2): The Structure of the Irish Sign Language*. The Linguistics Institute of Ireland, Dublin, Ireland.

Park, C.-B., Roh, M.-C. and Lee, S.-W. (2008). Real-Time 3D Pointing Gesture Recognition in Mobile Space. *In Proceedings of the 8th International Conference on Automatic Face and Gesture Recognition*. pp.1-6. Amsterdam, Netherlands.

Park, U., Tong, Y. and Jain, A.K. (2008). Face Recognition with Temporal Invariance: a 3D Aging Model. *In Proceedings of the 8th International Conference on Automatic Face and Gesture Recognition*. Amsterdam, Netherlands.

Prillwitz, S., Leven, R. and Zienert, H. (1987). *International Studies on Sign Language and the Communication of the Deaf*. Hamburg: Signum.

Quek, F. (1994). Toward a Vision-Based Hand Gesture Interface. *In Proceedings of the Conference on Virtual Reality Software and Technology*. World Scientific Publishing. pp.17-31. Singapore.

Sauer, J. (1997). *Neural Network with Java*. [online]. [Accessed 7 July 2009]. Available from World Wide Web: http://fbim.fh-regensburg.de/~saj39122/jfroehl/diplom/eindex.html

Shlens, J. (2005). *A tutorial on Principal Component Analysis*. Unpublished, Version 2

Stokoe, W.C. (1972). *Semiotics and Human Sign Languages*. The Hague: Mouton & Co.

Stokoe, W.C. (1980). Sign Language structure. *In Proceedings of the Annual Review of Anthropology*, pp.365-390. Washington D.C.

Stokoe, W.C. (2005). Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf. *Journal of Deaf Studies and Deaf Education*. **10**(1).

Suk, H.-I., Sin, B.-K. and Lee, S.-W. (2008). Recognizing hand gestures using Dynamic Bayesian Network. *In Proceedings of the 8th International Conference on Automatic Face and Gesture Recognition*. Amsterdam, Netherlands.

Sutton, V. (1995). *Lessons in Sign Writing, Textbook and Workbook (Second Edition)*. The Center for Sutton Movement Writing, Inc. La Jolla, CA.

Sutton, V. (1981). *Sign Writing for Everyday Use*. The Center for Sutton Movement Writing. La Jolla, CA.

Theodoridis, S. and Koutroumbas, K. (2006). *Pattern Recognition*. Elsevier. USA.

Vezhnevets, V., Sazonov, V. and Andreeva, A. (2003). A Survey on Pixel-Based Skin Color Detection Techniques. *In Proceedings of the 13th International Conference on Computer Graphics and Visions.* pp.85-92. Moscow, Russia.

Vogler, C. and Metaxas, D. (1998). ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis. *In Proceedings of the 6th International Conference on Computer Vision*. p.363. Bombay, India.

Whean, P.F. and Molly, D. (2001). *Machine Vision Algorithm in Java, Techniques and Implementation*. Dublin: Vision System Laboratory, School of Electronic Engineering, Dublin City University.

Yang, J., Zhang, D., Frangi, A.F. and Yang, J.-Y. (2004). Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. *In Proceedings of the Conference on Transactions on Pattern Analysis and Machine Intelligence*. **26**(1) pp.131-137. Washington, DC.

Zarit, B.D., Super, B.J. and Quek, F.K.H. (1999). Comparison of Five Color Models in Skin Pixel Classification. *In*: *Proceedings of the International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*. pp.58-63. Corfu, Greece.

Glossary

2DPCA - two-dimensional Principal Component Analysis. Feature extraction and data representation technique in two dimensions.

ASL - American Sign Language. The native language of the American Deaf Community

BSL - British Sign Language. The native language of the British Deaf Community

Classification – a process in which individual objects are divided into groups based on their recognised characteristic features.

DBN - Dynamic Bayesian Network. A probabilistic model which can represent a set of variables and their probabilistic dependencies.

Deaf – People with hearing difficulties who have a strong sense of Deaf identity and are involved in Deaf culture with a sign language as their preferred means of communication.

deaf – People with hearing difficulties who are not necessarily involved in Deaf culture and who may or may not use a sign language as their preferred language.

Feature – the interesting characteristic details of an image

HMM - Hidden Markov Model. A probabilistic model which can represent a set of variables and their probabilistic dependencies.

ISL - Irish Sign Language. The native language of the Irish Deaf Community

Loglik - log-likelihood. Shows the probability on how similar the unknown data to the trained model is

LSF – Langue des Signes Francaise - The native language of the French Deaf Community

NMF - non-manual feature. They refer to non verbal expression that is composed of face expression, body movement and eye gaze.

PCA - Principal Component Analysis. Feature extraction and data representation technique.

Phonemes – The basic units of speech sound. In the case of sign languages, they refer to the basic units that compose a sign articulation, namely the hand shape, palm orientation, location, movement and sometimes non-manual features

Poser – an animation tool which allows the user to work in a three-dimensional animated environment.

Recognition – identifying objects through their characteristic individual features

SL - Sign Language. Visual-gestural languages used by Deaf communities, that are fully expressive and natural communication channels.

Appendices

A. Classification System results

Table 7 shows the signs of the hand shape “b” category classified into the movement. The signs which are underlined cannot be distinguished using only the rule of the movement, therefore the last column give the additional rules which have to be applied to these signs. The last four rules of the movement are shown on the next page.

Movement (using signs of hand shape "b")			
Number of the movement	**Left hand** (can be any hand shape)	**Right hand** (hand shape "b")	Other features needed to tell the underligned signs apart
1	Boots, Build	Back1, Build	
2			
3	By, Big		
4			
5	Bank2, Body, Brave	Bank2, Beside, Between, Body, Boots, Brave	Position
6		Below	
7	Bring	By, Big, Bring	Orientation and position
8			
9	Baby	Baby	
10	Balance, Bread	Balance, Before, Begin, Bread	Orientation
11		Back3	
12			
13			
14	Back1, Back3, Bank1, Before, Begin, Below, Beside, Between, Bold, Bored, Boss, Brain	Behind, Bold, Brain	Position
15		Bank1, Bored, Boss	Orientation and position
16	Behind		
17	Book, Bury	Book, Bury	Orientation

Table 7: Represent the classification of all the signs in the hand shape "b" category. ***This table shows an example that the classification system is capable of identifying each sign.***

B. Python and Poser

1. Running Python

This section will cover step by step how to run and modify a Python script in Poser. Poser simplifies this by giving the user a predefined simple graphical user interface which allows the user to run a Python script by clicking on a button.

1.1 Using Script Palette

The script palette can be found in the menu item Window -> Python Script. A simple graphic user interface will be displayed and allows the user to execute the Python script. A few scripts have been defined and are ready to be used or modified for specific needs.

A button which is not labelled does not contain a script. A box opens with one click on that button and the user can choose a new script to add in the palette.

Figure 36: Python script palette

1.2 Running Python Script

To run a Python script, a simple click on the desired button in the palette is needed to execute the Python script automatically.

1.3 Modifying Python Script

A Python script can be opened in a text editor to modify the script by pressing and holding “ctr” button and clicking on the desired button. The script will open in the system’s default text editor and allows the user to do changes easily. The modified script only needs to be saved and reconfirmed with a click on the specified button in the palette to rerun the updated script.

A Python script can be removed from the palette by pressing and holding “alt” button and clicking on the desired button. The button will not be labelled and a new script can be open on this button. The removed Python script will not be deleted from the computer; it will only be removed from the palette.

A new Python script has to be saved in the format (.py) otherwise Poser will not accept the script as a Python script.

Figure 37: Python script palette with empty buttons

2. Python script

As mentioned before, Python is a language which can be easily learned. The structure of the source code is very simple. The easiest way to get started with Python is to have a look at some finished script. I started with Pierre's scripts called Angle Variations, Hand Opening, Random Variation; and I tried to understand them line by line.

There exist a few tutorials in the internet but basically it takes some research to find information about Poser Python.

2.1 Additional syntax help

Table 8 shows some syntax example used in Python.

Description	**Example**
Loops	for i in range(10):
Arrays	fingerparts = [“rIndex1”, “rIndex2”, “Right Hand”]
Choose option	if(wert == 1): else:
Counter	range(10)
Equation	wert == 1
Set a variable	wert = 1
Swap values	nudge, wink = wink, nudge
Output of information	print act.Name()
Number format of the output	print “value: “ + “%3.2f”% val
Show a entry box	Poser.DialogTextEntry(message = “...”)
Comments	# this comment a text

Table 8 : Additional Syntax Help

2.2 Parameter

	Parameter	**meaning**
general		
	kParmCodeXROT	rotation about the X-axis
	kParmCodeYROT	rotation about the Y-axis
	kParmCodeZROT	rotation about the Z-axis
	kParmCodeSCALE	
	kParmCodeXSCALE	amount of scale along the X-axis
	kParmCodeYSCALE	amount of scale along the Y-axis
	kParmCodeZSCALE	amount of scale along the Z-axis
	kParmCodePOINTAT	degree to which an actor set to point at something will actually point at it
	kParmCodeValue	Placeholder for a value. Usually, these values are used functionally to control other things such as full-body morphs
specific		
Camera		
	kParmCodeFOCAL	Camera focal length parameter
	kParmCodeFOCUSDISTANCE	Camera focus distance parameter (affects depth of field effect)
	kParmCodeYON	Camera parameter specifying a far clip plane distance
Light	These codes are used to set the light types.	They are typically used in conjunction with the actor.SetLightType()
	kLightCodeSPOT	Spotlight
	KLightCodesPOINT	Point light
	kLightCodeLOCAL	Local light
	kLightCodeINFINITE	Infinite light

Table 9 : Parameters used in Poser Python. ***These are the main parameters used to move an actor.***

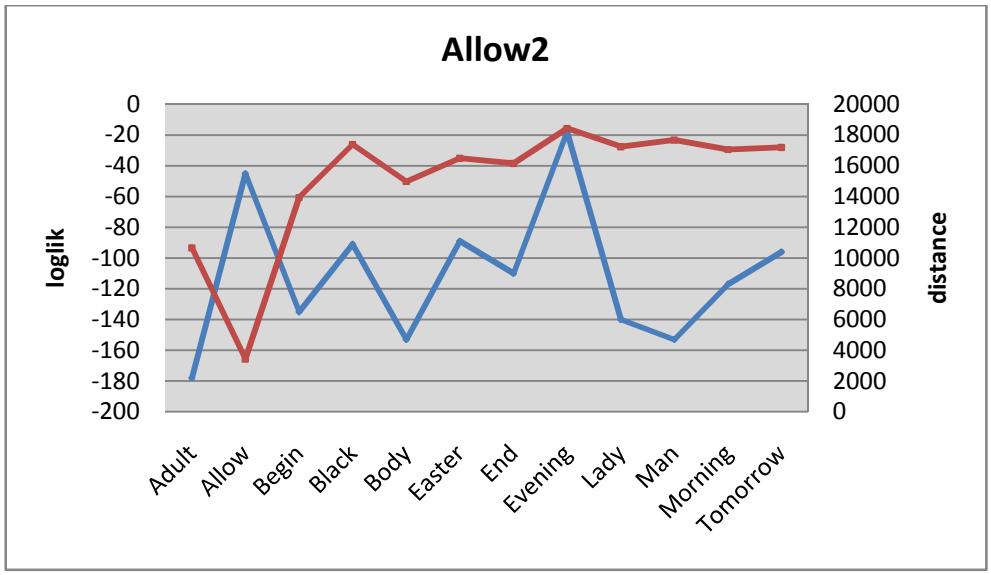
C. System Results

1. Results of the system: using combination of the HMM and the distance metric

In the following pages, some results of the recognition system are shown. These results are represented in a table and a graph for each tested sign. The tables show the perpendicular distance and the log likelihood of the data compared to all the trained data. The graphs represent the values of the tables in a graph. Only the signs where the distance is below 10,000 will be consider and out of those the closest value to 0 of the loglik will define the recognized sign. In the figures, the red line represents the perpendicular distance and the blue line represents the log likelihood.

Allow2	distance	Loglik
Adult	10663.59	-178
Allow	3432.56	-45
Begin	13943.19	-135
Black	17399.19	-91
Body	14982.50	-153
Easter	16489.67	-89
End	16151.40	-110
Evening	18431.26	-18
Lady	17241.00	-140
Man	17681.45	-153
Morning	17063.62	-117
Tomorrow	17196.65	-96

Table 10: Results of the second sample of the sign "allow"



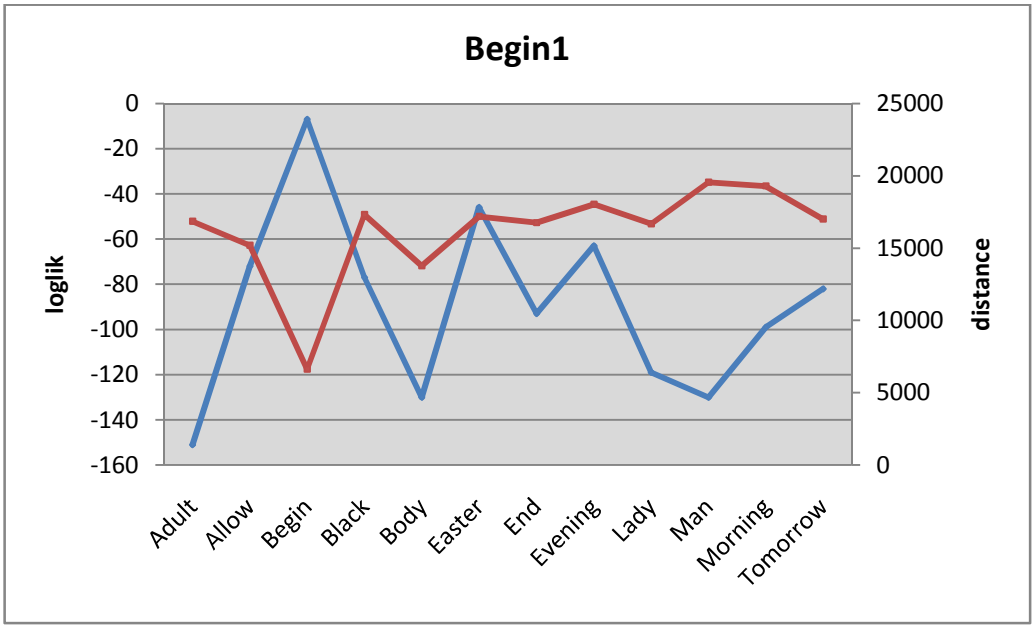
Graph 16: Visualisation of the results values of the sign "allow"

As seen in the Table 8 and Graph 50 the sign “Allow” has only one distance below 10,000. Using a wider range of signs, some of the distance will be closer together. However, this sign is perfectly recognised by the developed system.

The next tested sign is the sign “Begin”. In this case the table shows that only one distance is lower than 10,000. Taking this into account, with a full overview on all the loglik values, the closest value of the likelihood is of the trained sign “Begin”. In this case, the distance would not have been needed.

Begin1	distance	Loglik
Adult	16862.59	-151
Allow	15198.74	-72
Begin	6634.09	-7
Black	17315.05	-77
Body	13775.89	-130
Easter	17199.19	-46
End	16760.62	-93
Evening	18031.09	-63
Lady	16685.83	-119
Man	19559.49	-130
Morning	19301.56	-99
Tomorrow	17018.77	-82

Table 11: Results of the first sample of the sign "begin"

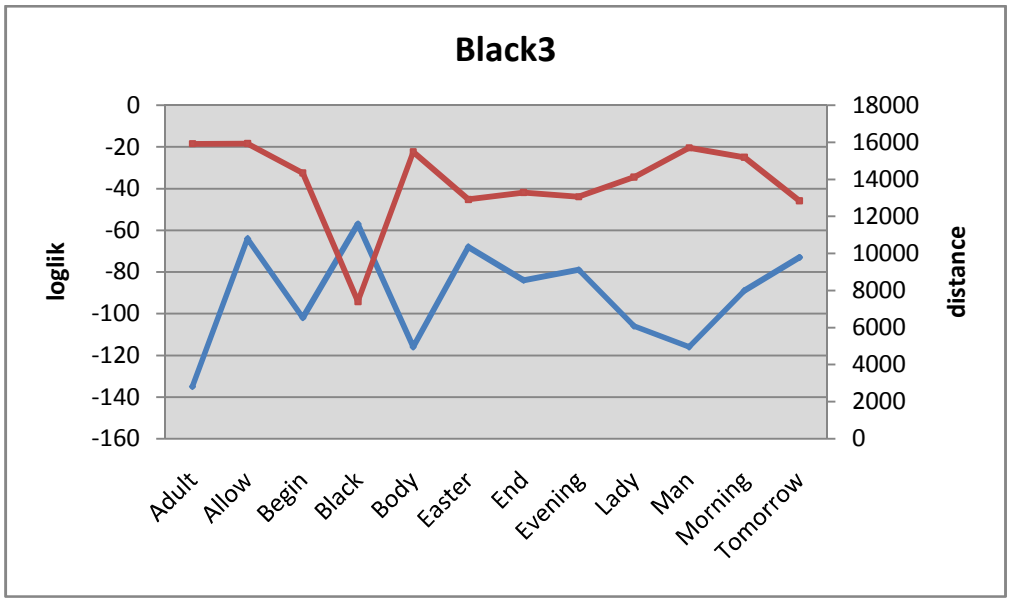


Graph 17: Visualisation of the results values of the sign "begin"

In this case, again only one distance is below 10,000 and the HMM shows with the loglik that the new data looks the most similar to the sign “Black”.

Black3	distance	Loglik
Adult	15918.92	-135
Allow	15931.34	-64
Begin	14341.67	-102
Black	7401.02	-57
Body	15487.02	-116
Easter	12914.39	-68
End	13288.66	-84
Evening	13053.97	-79
Lady	14116.71	-106
Man	15698.00	-116
Morning	15189.48	-89
Tomorrow	12841.15	-73

Table 12: Results of the third sample of the sign "black"

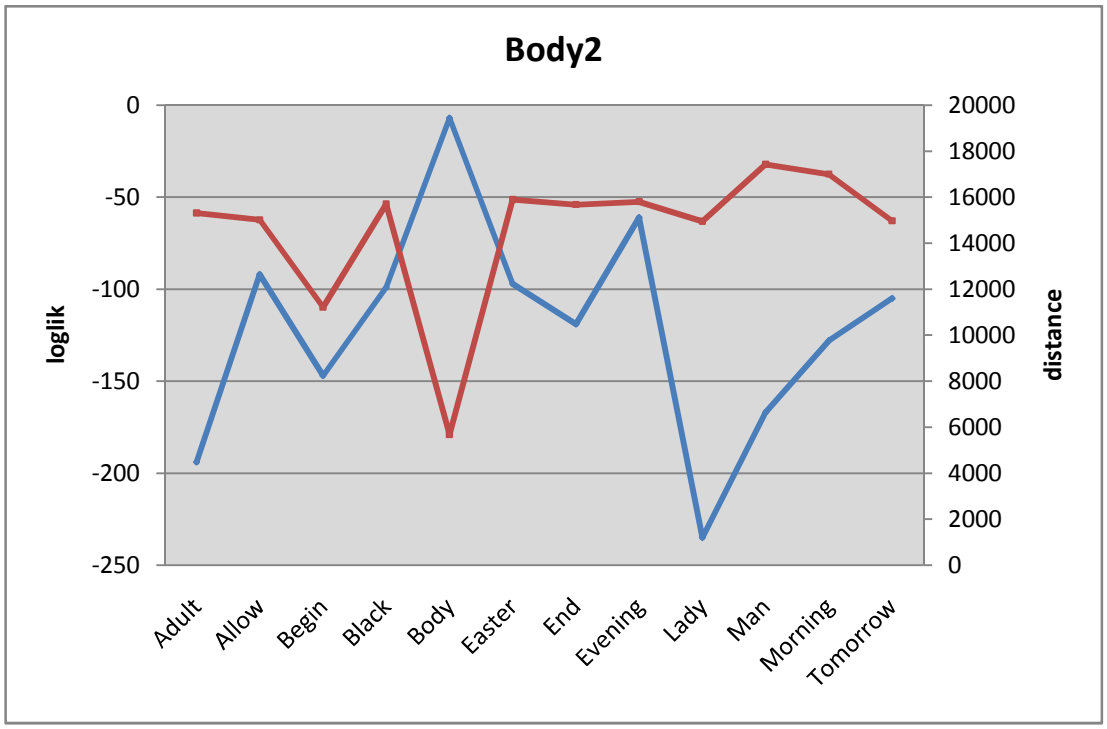


Graph 18: Visualisation of the results values of the sign "black"

The same as the previous sign, there is only one distance which is lower than 10,000 and additionally the loglik is on the same comparison the closest to 0 as well.

Body2	distance	Loglik
Adult	15315.42	-194
Allow	15015.67	-92
Begin	11222.78	-147
Black	15709.69	-99
Body	5682.06	-7
Easter	15901.12	-97
End	15681.85	-119
Evening	15808.09	-61
Lady	14952.33	-235
Man	17430.50	-167
Morning	17001.61	-128
Tomorrow	14973.91	-105

Table 13 : Results of the second sample of the sign "body"

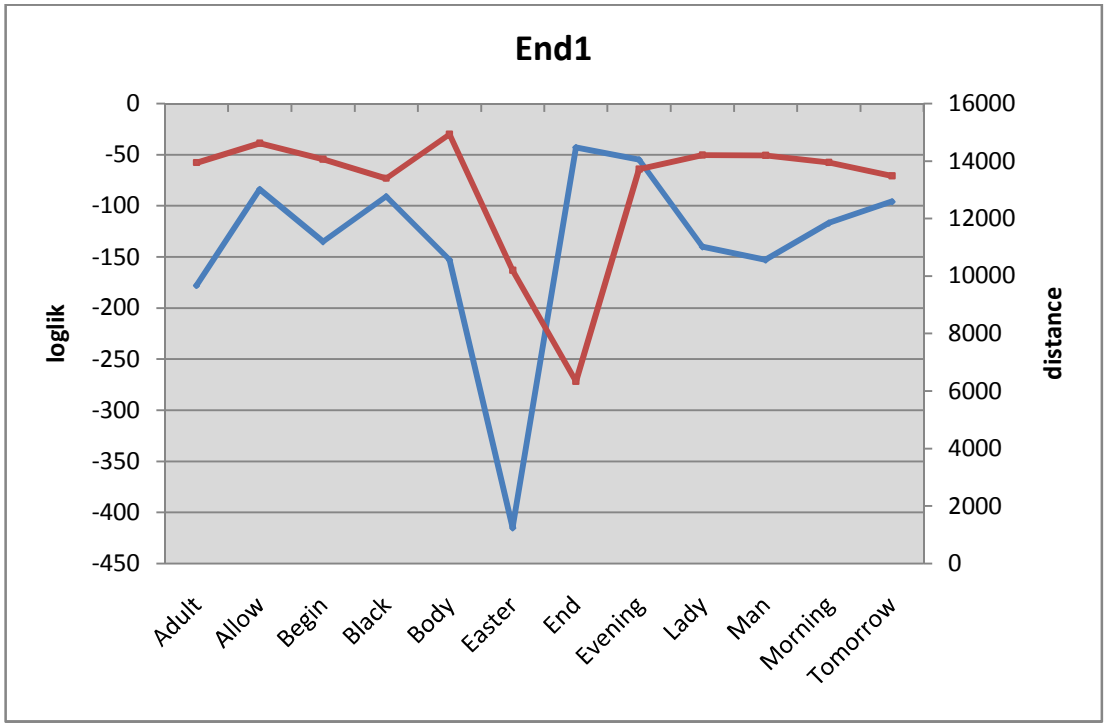


Graph 19: Visualisation of the results values of the sign "body"

This sign is perfectly recognized by its distance and loglik.

End1	distance	Loglik
Adult	13945.60	-178
Allow	14615.30	-84
Begin	14061.44	-135
Black	13399.95	-91
Body	14932.32	-153
Easter	10198.91	-415
End	6336.94	-43
Evening	13714.32	-55
Lady	14204.26	-140
Man	14199.59	-153
Morning	13951.88	-117
Tomorrow	13485.57	-96

Table 14 : Results of the first sample of the sign "end"

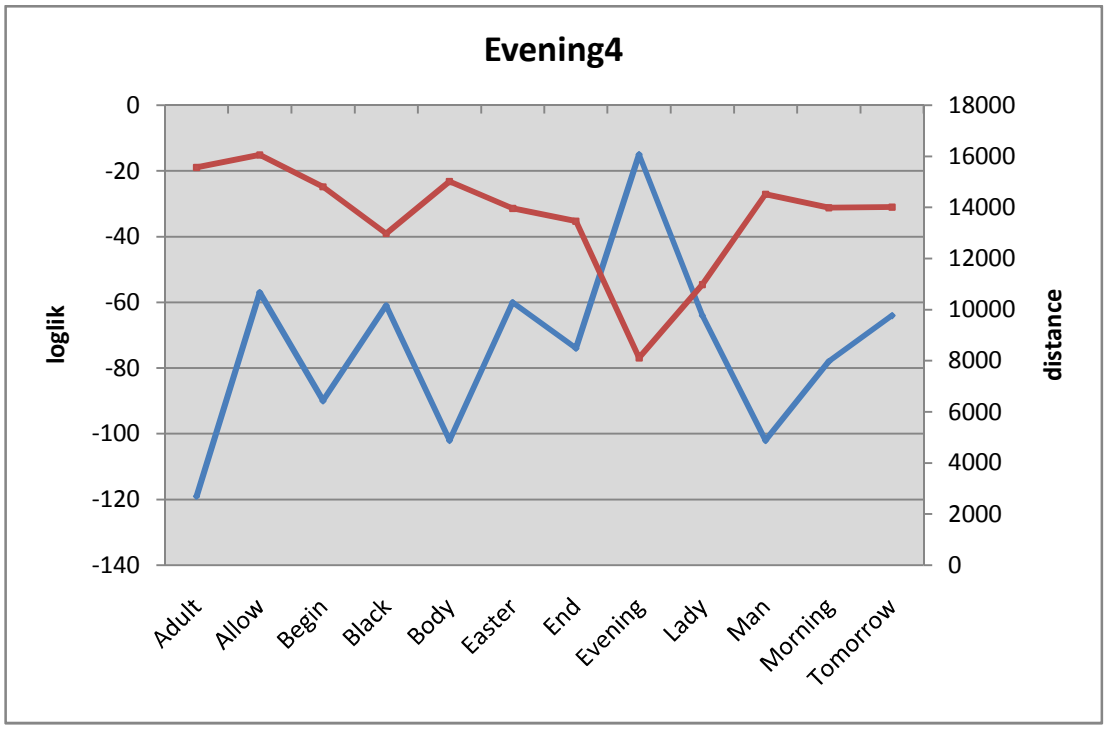


Graph 20: Visualisation of the results values of the sign "end"

This sign is perfectly recognized having only one distance below 10,000 and the loglik value closest to 0 fit as well.

Evening4	distance	Loglik
Adult	15562.66	-119
Allow	16052.16	-57
Begin	14802.62	-90
Black	12974.49	-61
Body	15013.27	-102
Easter	13962.99	-60
End	13458.7	-74
Evening	8110.37	-15
Lady	10981.91	-64
Man	14516.27	-102
Morning	13980.88	-78
Tomorrow	14011.2	-64

Table 15: Results of the fourth sample of the sign "evening"

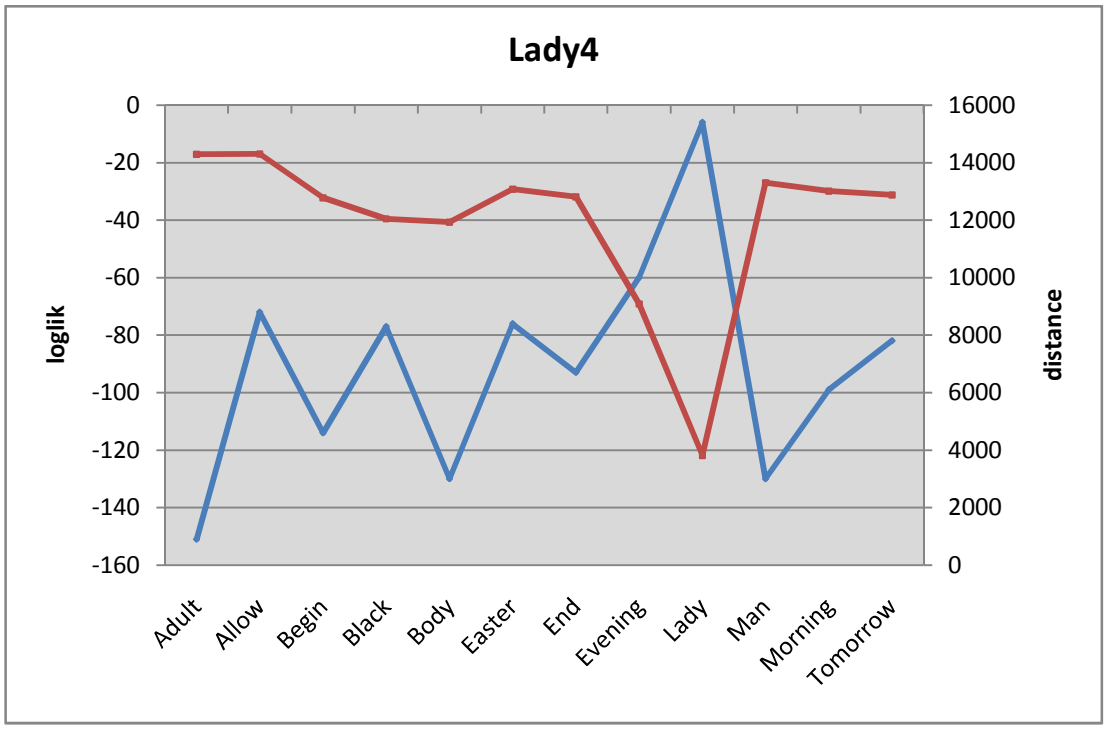


Graph 21: Visualisation of the results values of the sign "evening"

In this case two distances have a value below 10,000 but the loglik makes a clear differentiation.

Lady4	distance	Loglik
Adult	14288.00	-151
Allow	14303.64	-72
Begin	12775.84	-114
Black	12047.60	-77
Body	11932.14	-130
Easter	13076.08	-76
End	12810.36	-93
Evening	9086.58	-60
Lady	3813.80	-6
Man	13305.35	-130
Morning	13012.59	-99
Tomorrow	12877.74	-82

Table 16 : Results of the fourth sample of the sign "lady"

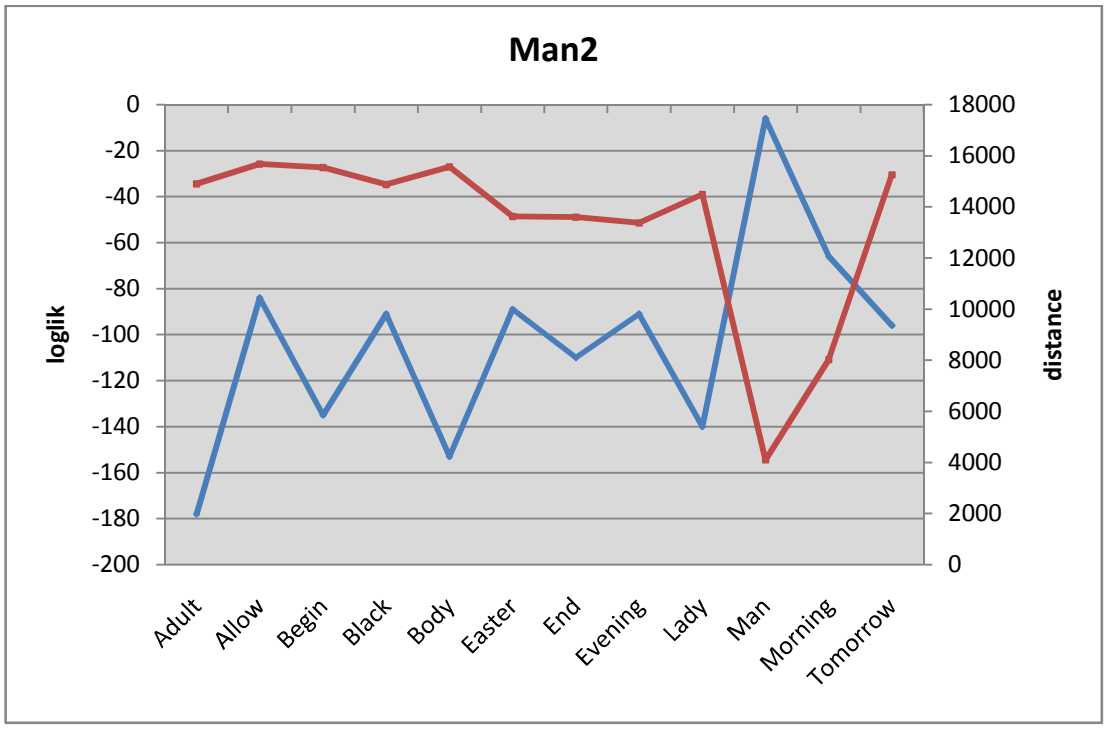


Graph 22: Visualisation of the results values of the sign "lady"

The sign “Man” looks similar as the sign “Morning”, the only different is the direction of the movement: compared to the one of these signs, the other one moves in the opposite direction. The loglik differentiate the two signs.

Man2	distance	Loglik
Adult	14901.14	-178
Allow	15680.16	-84
Begin	15538.47	-135
Black	14875.76	-91
Body	15564.51	-153
Easter	13623.67	-89
End	13601.30	-110
Evening	13375.02	-91
Lady	14494.96	-140
Man	4098.05	-6
Morning	8024.03	-66
Tomorrow	15257.35	-96

Table 17 : Results of the second sample of the sign "man"

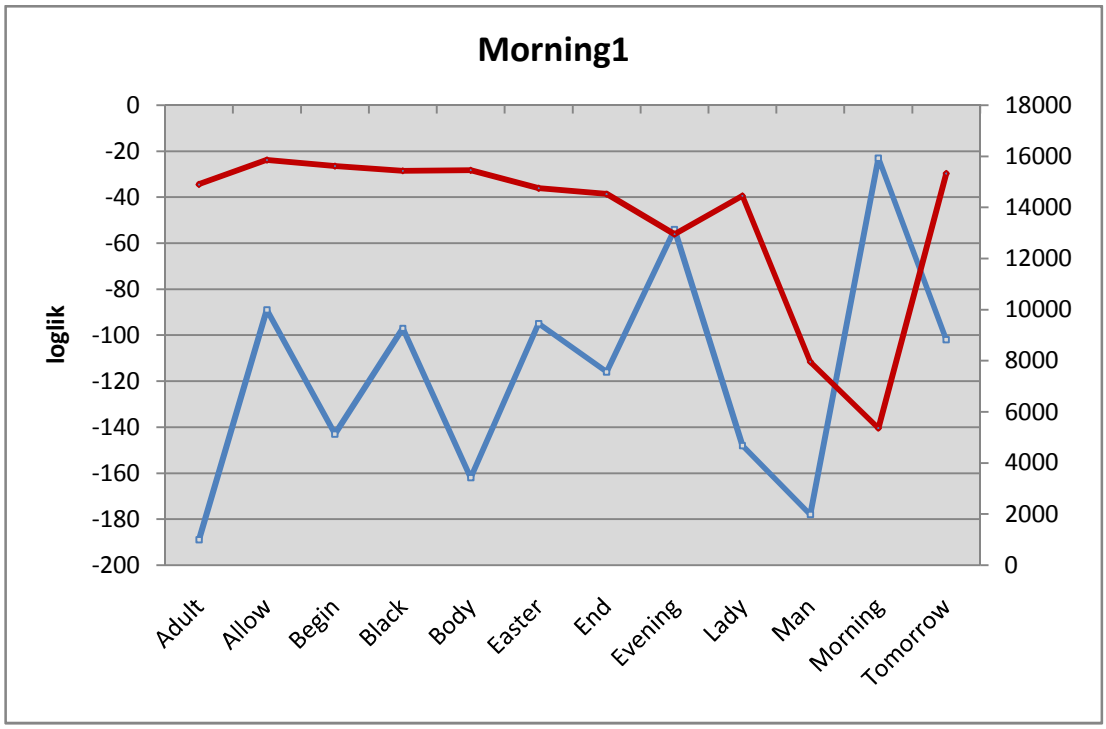


Graph 23: Visualisation of the results values of the sign "man"

As mentioned in the previous example, the sign “Morning” is very similar to the sign “Man”, but the difference is shown in the distance and the loglik.

Morning1	distance	Loglik
Adult	14895,12	-189
Allow	15853,07	-89
Begin	15614,48	-143
Black	15431,07	-97
Body	15452,98	-162
Easter	14747,94	-95
End	14520,72	-116
Evening	12949,08	-54
Lady	14450,03	-148
Man	7955,69	-178
Morning	5359,08	-23
Tomorrow	15323,83	-102

Table 18 : Results of the first sample of the sign "morning"

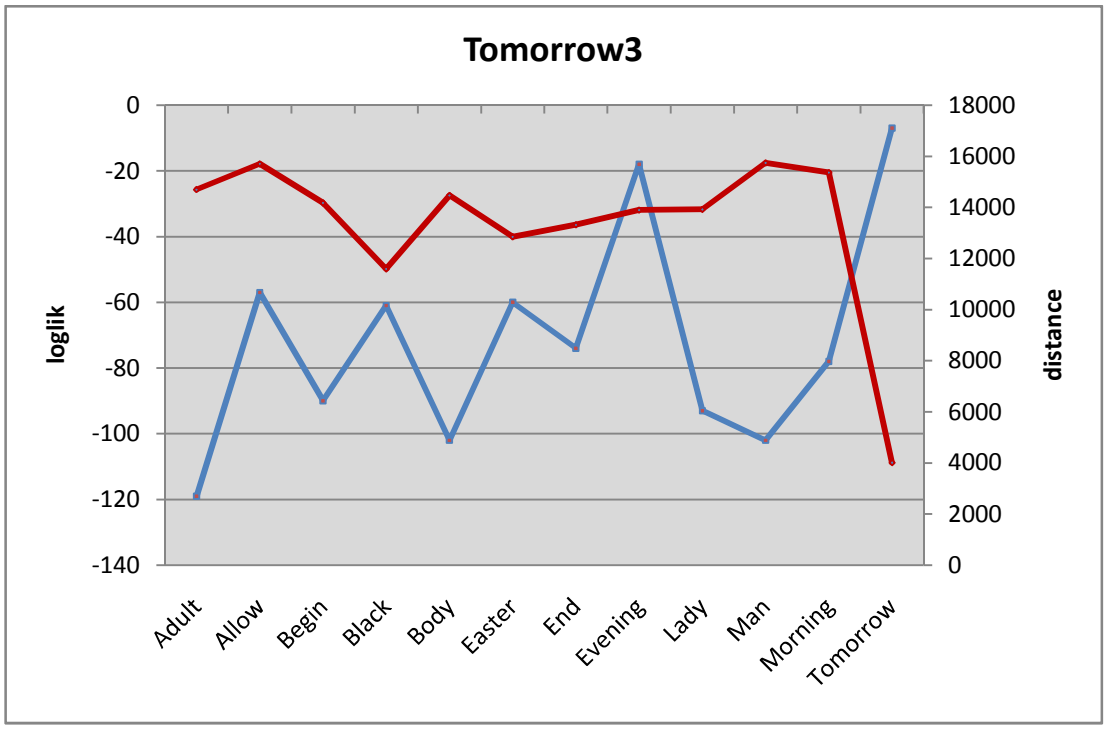


Graph 24: Visualisation of the results values of the sign "morning"

The sign “Tomorrow” is very different as all the other sign which have been chosen for the testing. Therefore the results are very clear.

Tomorrow3	distance	Loglik
Adult	14693,23	-119
Allow	15702,39	-57
Begin	14174,26	-90
Black	11592,72	-61
Body	14466,59	-102
Easter	12846,61	-60
End	13317,6	-74
Evening	13893,89	-18
Lady	13919,02	-93
Man	15738,61	-102
Morning	15361,92	-78
Tomorrow	4016,94	-7

Table 19 : Results of the third sample of the sign "tomorrow"



Graph 25: Visualisation of the results values of the sign "tomorrow"

D. Attached CD