Rendering an Avatar from SignWriting Notation for Sign Language Animation

by

Kgatlhego Aretha Moemedi

A thesis submitted in fulfilment of the requirements
for the degree of
MAGISTER SCIENTIAE
in the Department of Computer Science,
University of the Western Cape

Supervisor: James Connan

November 2010

Rendering an Animated Avatar from SignWriting

Kgatlhego Aretha Moemedi

Keywords

Avatar
Blender
H-Anim
Key-frame
SASL
SignWriting
Sign Language
Sign Language Animations
Sign Language Parameters
SWML

Abstract

Sign languages are the first languages of many Deaf people. They are complete natural languages and are communicated in a visual-gestural modality. Several sign language notations have been proposed. Not one of them has been accepted as a standard written form of sign languages. Therefore, it is undesirable for a sign language system to produce a written form of sign language as output. The output must rather be presented in the form of video or virtual signing that uses avatars to perform sign language gestures.

This thesis presents an approach for automatically generating signing animations from a sign language notation. An avatar endowed with expressive gestures, as subtle as changes in facial expression, is used to render the sign language animations. SWML, an XML format of SignWriting is provided as input. It transcribes sign language gestures in a format compatible to virtual signing. Relevant features of sign language gestures are extracted from the SWML. These features are then converted to body animation parameters, which are used to animate the avatar. Using key-frame animation techniques, intermediate key-frames approximate the expected sign language gestures. The avatar then renders the corresponding sign language gestures. These gestures are realistic and aesthetically acceptable and can be recognized and understood by Deaf people.

November 2010

Declaration

I declare that *Rendering an Avatar from SignWriting Notation for Sign Language Animation* is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Full name: Kgatlhego Aretha Moemedi Date: November 2010

Signed:________________

Acknowledgments

I am grateful for all that I have, and I would like to thank God, the Almighty Father for all the things He has done for me. I would like to thank my supervisor, James Connan for the outstanding, committed supervision and the confidence he has in me. Many thanks for the patience and the guidance you offered me. I will forever be grateful, and have learnt a lot from you. The Department of Computer Science, thank you for the support. Prof Venter, many thanks for the wise words of encouragement. Mrs Connan, thank you for taking good care and making sure all was in order. Many thanks for the motivation. Thanks to the SASL group, the best research group to work with, and to the outstanding research and achievements. Special thanks to Mehrdad Ghaziasgar, for the good advice and help with the thesis. Jameel Adams, thank you for the support and for your kindness. Imran Achmed thank you, for a competent and very successful journey. Thank you for keeping me on my toes.

I would also like to thank SignWriting and Blender who have assisted in this research, particularly Valerie Sutton whose contribution greatly assisted my work.

To the Davids family, thank you for making this place a home away from home. Saarah and Amirah thank you for the sisterly love you have shown me.

To my friend, Dintle Selao, thank you for listening and for the laughter. You believed in me, thank you for the aspiration. To all my friends, many thanks for motivating me to work hard and for the encouragement.

To Mom and Dad, I have grown to be ambitious and believe in myself, and it all thanks to you. I can never thank you enough. To my brother Katlego, you are the best brother ever. Thank you for your support and words of wisdom.

Lastly many thanks to the Centre for Excellence at the University of the Western Cape sponsored by Telkom, Cisco and THRIP for the financial assistance.

Contents

Glossary

3D	– Three Dimensional
API	– Application Programming Interface
ASL	– American Sign Language
Auslan	– Australian Sign Language
BAP	– Body Animation Parameter
BDP	– Body Definition Parameter
BSL	– British Sign Language
CSL	– Chinese Sign Language
DGS	– German Sign Language
DOF	– Degree of Freedom
DTD	– Document Type Definition
FAP	– Facial Animation Parameter
FDP	– Facial Definition Parameter
FK	– Forward Kinematics
GSL	– Greek Sign Language
HamNoSys	– Hamburg Notation System
H-Anim	– Humanoid Animation Working Group
HTML	– HyperText Markup Language
IK	– Inverse Kinematics
ISL	– Irish Sign Language
ISWA	– International SignWriting Alphabet
LoA	– Level of Articulation
MPEG	– Moving Picture Experts Group
MPEG-4 SNHC	– MPEG-4 Synthetic Natural Hybrid Coding
MT	– Machine Translation
SASL	– South African Sign Language
SAX	– Simple API for XML
SEE	– Signed Exact English
SIL	– International Sign Language
SSL	– Slovene Sign Language
STEP	– Scripting Technology for Embodied Persona
SWML	– Sign Writing Markup Language
VRML	– Virtual Reality Markup Language
XML	– Extensible Markup Language

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Communication by means of spoken language is a natural custom in human interaction. However, communication barriers can be caused by language variations between the participants in a conversation. This is often the case in cross-modal languages such as spoken and sign languages (SLs). Spoken languages are oral-aural while SLs are visual-gestural. The visual modality of SLs gives them a completely different morphology to spoken languages. Hence a communication barrier often exists between deaf and hearing communities. The term 'deaf' refers to audiological impairment. The term 'Deaf' refers to people who use SL as their first language and who are members of the Deaf Community.

Research shows that most Deaf children are born to hearing parents [30]. Although many Schools for the Deaf teach through oralism children pick up SL elsewhere [1]. As a results hearing parents may find it difficult to communicate with their children. Providing Deaf people access to information and technology in their native language is a significant step towards breaking the communication barrier between Deaf and hearing communities. In order for Deaf children to obtain full cognitive development, they require full access to education, medical and social environments. Full access means access in their native language.

Machine translation (MT) system can bridge the communication barrier between deaf and hearing communities. The South African Sign Language (SASL) Project at the University of the Western Cape translates English to SASL and vice versa [47]. The primary goal is to develop technologies that facilitate communication between Deaf and hearing communities. The intention is to provide a natural means of communication. It develops technologies that will allow for the creation of English and SASL MT and educational tools.

1.2 South African Sign language

The Central Statistical Service SA Yearbook 1998, estimates that in 1995 there were at least 4 million South Africans that were deaf or hard of hearing [49]. This forms a minority group which is often misunderstood by the hearing community. Deaf people are not a homogeneous group and are found in all walks of life. Most of the deaf people, up to 90%, are born to hearing families [30]. Deaf children cannot hear the spoken language around them. They do not acquire language as easily or naturally as hearing children in hearing families. Thus, most deaf children become economically disadvantaged. On average Deaf students finish high school with an English reading level equivalent to grade four hearing students [26]. The 2001 South African Census reported that 70% of the disabled people in South Africa have hearing loss [49]. In the late 1990s only an estimated 15 Deaf people in South Africa had university degrees [57]. Unemployment and illiteracy are high in the Deaf community. A breakthrough to these scenarios can be achieved by providing the Deaf people access to information in their native language. Access to SASL enables Deaf people to communicate.

The South African constitution has advanced to recognise SASL as the official language of the Deaf [13]. Yet, SASL is not a school subject [13]. In Deaf schools only 14% of teachers are fluent in SASL [13]. Grade 12 is offered in twelve schools for the Deaf. These schools are based in only 3 provinces [13]. An estimate of 600 000 Deaf people use SASL [13]. However there are no public services available in this language, so Deaf people use the services of SL interpreters. Unfortunately, there are not enough professional SASL interpreters. Table 1.2 gives a summary of a ratio of interpreters to SASL users using a low estimate of 500 000 and a high estimate of 1.5 million SASL users [57].

1.2.1 Variation SASL

Several types of variation occur in all human languages. Variation can be due to racial background or regional variation, among others. Since 1994, research has confirmed that there is one SASL [25]. There may be dialectal variation at the level of vocabulary. Different groups whether by age, school, spoken language or other variables often have different signs. Yet the grammar remains the same for all Deaf people [36].

Category	Number	Ratio of interpreters to SASL users 500 000	Ratio of interpreters to SASL users 1.5 million
Professional on the basis of accreditation of the South African Translators Institute	4	1: 125 000	1: 375 000
Professional on the basis of SASL fluency, experience and competence	20	1: 25 000	1: 75 000
Grassroots interpreters include Children of Deaf parents and teachers of the Deaf	40	1: 12 500	1: 37 500

Table 1.1: SASL interpreters in South Africa, by category, number and by ratio of interpreters to SASL users [57].

Regional Variation

There are some discrepancies in the way people from different regions sign. This could be due to the fact that SL has never been formally taught in schools. Schools for the Deaf tend to be regional. Thus the children will go to the school in their region or nearest region. There are at least four different signs that are used for *mother*, but the same grammatical signals on the face are used across cultural or racial groups [36].

Racial Variation

Another variation that exists in SASL is due racial factors. In general, Black Deaf community tends to use culturally different forms of signs. Oftentimes, these signs differ from those of the White Deaf community. An example is the sign for *basket* in SASL. White signers indicate a basket carried by hand. Rural Black signers indicate a basket being carried on the head [52].

1.3 Motivation

To date, the SASL Project has among others built a realistic and physically plausible avatar [56], which is used to render the SL gestures. The project has implemented gesture recognition using feature vectors [39], which is used to recognize the SL gestures, which will be translated into English. It uses a mobile phone as a service-delivery device [21] that sends the SL gestures and English text or speech to the Deaf or hearing person respectively. The SASL Project supports Deaf people by presenting SL gestures through an avatar. Thus, a system that can animate the avatar to render SL gestures is needed.

1.4 Research question

From this, a research question is formulated.

Can sign language animations be rendered by an avatar from a sign language notation?

Given the hypothesis that such a system can be implemented:

Are the sign language gestures performed by the system intelligible?

Are the sign language gestures performed by the system an accurate reflection of the input notation?

The important factor of visualising SL animations is how the avatar is animated. The manner in which the avatar is animated affects the realism of the SL gestures rendered by the avatar.

1.5 Thesis Outline

The following is a brief outline of this thesis.

Chapter 2 discusses the basic concepts, structure and the writing systems of SLs. First it defines SLs. It discusses the history of SLs and its discovery as real and independent languages. Then it investigates the common misconceptions that hearing people have about SLs. Finally, it gives an outline of the linguistic structure of SLs.

Chapter 3 discusses the issues relating to SL, such as the availability and generation of corpora. It also gives a brief overview of the linguistic aspects of SLs.

Chapter 4 discusses how SLs can be visualised. It discusses the use of videos and avatars to visualise SLs. It shows that despite the high quality of videos, virtual signing is more advantageous for constructing a general visualisation system. The advantages and disadvantages of visualisation approaches are examined.

Chapter 5 addresses the design of the system. It discusses the standards and technologies used. It outlines the structure of the avatar. It explains the technologies used to achieve a physically plausible avatar. Then it discusses the software used for the modelling and animation of the avatar. It discusses the functionalities and capabilities of the technology used for animation. Finally, it discusses the interface of the SL notation used to describe the gestures.

Chapter 6 discusses the implementation of the system. It discusses the parser implementation, which is divided into two main classes. Then it discusses the animation techniques used to achieve smooth and realistic animations.

Chapter 7 describes how the system was evaluated. It gives an overview of the criteria used for judging the recognition and the accuracy of the signs rendered by the system, how the evaluators for the system were selected, how the signs used to evaluate the system were selected and how the evaluation of the criteria was conducted.

Chapter 8 presents the results of the research based on the signs produced by the system. It evaluates the intelligibility of the SASL signs rendered by the system from the input notation. It investigates how accurately the SASL animations match the input notation. Then it discusses the factors that influence the quality of sign animation.

Chapter 9 provides a conclusion on the work presented in this thesis and it provides recommendation on future work.

Chapter 2

Sign Languages

SLs are the first languages of the Deaf. They are complete natural languages and are communicated in a visual-gestural modality. This chapter discusses the basic concepts, structure and SL notations. First, it defines SLs. It discusses the history of SLs and their discovery as real and independent languages. Secondly, it investigates common misconceptions regarding SLs and explores how these misconceptions may have come about. Reasons why these misconceptions are untrue are given. Thirdly, the chapter outlines the linguistic structure of SLs. Last, an outline of SL notations that may be used to record SLs, is given. The methodologies they adopted are examined with their respective advantage and disadvantages.

2.1 Background

SLs are visual-gestural languages made up of manual and non-manual features. Manual features refer to hand configuration, its movement, location and orientation in the signing space. Non-manual features refer to facial expressions and body movements. SLs use a combination of hand and finer movements, lips, facial expression and body language to communicate and convey information to others. These languages developed due to the difficulties experienced by Deaf people in communicating with other people, both Deaf and hearing. SLs are as complex and rich as spoken language [54], even though in certain countries there is no official recognition for the languages. They are complete natural

languages, with their own morphology [1], phonology[2] and syntax[3] [54].
In 1960, the work of Stokoe showed that SLs are real and independent languages [54]. He argued that SLs are the native languages of Deaf people. Prior to his work, SLs were not considered to be real languages. They were considered a set of meaningless gestures. As a result SL could not be used in the education of Deaf children. Other methods were used which include oralism, total communication and bilingual cultural method.

Oralism

Oralism is the practice of communication by lip-reading and speech alone [4]. There is no SL used. This method requires Deaf people to practice speech sounds. Most Deaf people who have been forced into oralism later go on to learn SL [4].

Total communication

Total communication involves the use of any form of communication: SL, finger-spelling, lip-reading and pictures [5]. Total communication has been criticized for the poor quality of the SL [5]. The method was regarded ineffective because both signing and speech produced poor SL. The SL used in total communication is more closely related to English.

Bilingual cultural method

The bilingual cultural method involves the teaching of SL as a first language and English as a second language. In this method, SLs are used to teach spoken languages. Students are taught SL and spoken language in parallel.

2.2 Myths and misconceptions

The previous section showed that SLs are natural languages. This section studies the common misconceptions of SLs. Since Stokoe's discovery, SLs have been acknowledged to possess the grammatical and expressive complexity of a true language [54]. Schools are teaching SLs and SL research is being carried out in many countries. However many misconceptions still exist, even within the Deaf community itself. The misconceptions are discussed in the following subsections.

[1]Morphology refers to the manner in which parts of signs are arranged to make a sentence.

[2]Phonology refers to the manner in which the phonemes are combined.

[3]Syntax is a rule that governs how signs are arranged to form a sentence.

2.2.1 Myth: Sign languages are universal

SLs are not universal. If this were true, the communication barrier between Deaf communities would be absent. There are many SLs that have developed independently in different parts of the world. There are at least 25 SLs in Africa [42]. In addition, the Ethnologue of world languages lists 121 SLs [14]. Most of these SLs are distinct. Due to diversity of culture and background, SLs vary from country to country. Hence there is Chinese Sign Language (CSL), American Sign Language (ASL), Irish Sign Language (ISL), Danish Sign Language (DSL), SASL, etc. The signs can be iconic, but are different. An example is the sign *tree*. In CSL, the hands in *open* X[4] hand-shape are moved upwards to outline the trunk [33]. In DSL, the hands portray the round shape of the top of the tree then moves downward along the trunk. In ASL the forearm in a *five* hand-shape is held upright and twisted, which resemble the tree trunk and its branches. The different signs for *tree* are shown in Figure 2.1. Furthermore within a country SL might still have some variation. In South Africa for example, the lexical variation differs across regions, but the grammatical structure is the same.

Figure 2.1: The different signs for *tree* in CSL, DSL and ASL [33].

Although SLs are not universal, there is an International Sign Language (SIL)[5]. SIL is an artificially formulated SL used to facilitate communication among Deaf people at international level. It is composed of loan words and onomatopoeic words that Deaf people conventionally agreed upon. While SIL is significant, it does not have a concrete

[4] Open X hand shape in CSL has the index finger and thumb extended

[5] SIL is formerly known as Gestuno [48]. The name Gestuno is an Italian word that references gestures and oneness. Hence Gestuno means *"the unity of sign languages"*

grammar. The lexicon is not permanently conventional; it keeps on changing. Moreover, there are just about 1500 signs in SIL [29].

2.2.2 Myth: Sign languages are based on spoken languages

SLs are not dependent on spoken language. They have rich complex grammars [54]. They are not a spelling of spoken words. In SL fingerspelling is mostly used for names of people, geographical places and scientific words borrowed from spoken language. Furthermore some signs are based on the hand-shape of the first letter of the English word. The sign for *aunt* and *uncle* in SASL are made with a hand-shape of letter *A* and *U* respectively. SLs take advantage of unique features of the visual medium, thus are visual-gestural. In contrast spoken languages are oral-aural and linear. In spoken language, most words and the referent objects have an arbitrary relationship. For example the size, pronunciation or spelling of the word *car*, does not relate to the referent object. In contrast, SLs are highly iconic. Many signs resemble the referent object. For example the sign *tree* in Figure 2.1 shows different resemblances of the referent object. The complex spatial grammar of SLs differs from spoken languages. For example, in English the verb comes before the object: *I am reading a book*. In SL, the verb comes after the object: *I a book reading*. However there are SLs that may be considered visual representation of spoken languages. An example is Signed Exact English (SEE). It visualizes SL using English grammar and morphology [23]. These signs are used in the exact English word order.

TODAY WE HAD MUCH RAIN. *(correct English grammar)*	English
Today we had much rain. *(there is a sign for each word in the sentence in the correct order)*	SEE

2.2.3 Myth: Sign languages are always iconic

Section 2.2.2 indicated that SLs are iconic. Iconicity does not refer to pointing at objects (e.g pointing a car nearby) or acting out the actual events (e.g. jumping up and down). Iconicity is a resemblance between a linguistic term and its meaning. Iconicity is present in both spoken and SLs. However, it is more common in SLs than in spoken languages. This could be due to the fact that in the real world, there are more visual images than sounds that can relate clearly to a concept. For example a *table* can be characterised by its shape. The visual-gestural modality of SLs provides a number of possible ways to make visual imagery more readily recognizable. Iconicity is not always present in SLs – some signs do not have any iconicity to what they represent. The formation of signs in SLs is not determined only by their resemblance to an object or action. Some iconic signs can develop into arbitrary symbols over time [31]. For example the Australian Sign Language (Auslan) sign for *library* originally meant *hairclip*. *Hairclip* was the name-sign of the school librarian [31]. This sign may be arbitrary to some signers. In addition, even when signs are iconic, the resemblance can differ as shown in Figure 2.1.

2.2.4 Myth: Sign languages cannot be written

SLs were not regarded as real languages because they were thought to lack a written form. There is currently no universally accepted written form of SL, but there are ways to transcribe anything a person signs. A number of SL notations have been proposed. Some have become widely used by SL researchers. Others are being used as educational tools. SLs can be written by using glossing or notations. Glossing uses spoken language meaning to translate signs. In some cases special symbols are used to indicate facial expressions. Notations use special symbols to describe the physical parameters of SLs. Examples of SL notations are Stokoe, Hamburg Notation System (HamNoSys) and SignWriting. These notations are discussed in Section 2.4.

2.3 The structure of SLs

This section studies the structure of SLs as well as the phonemes of SLs. In spoken languages, a phoneme is the smallest building block of speech that distinguishes meaning. Similarly, a sign in SL is composed of a phoneme. The phonemes of SL are divided into five components: hand-shape, orientation, location, movement and non-manual features [54]. They occur simultaneously and distinguish signs from each other as phonemes in spoken languages distinguish words from each other. Each of the phonemes of SL is discussed the next subsections.

2.3.1 Hand-shape

The hand-shape is probably the most apparent parameter of a sign. Hand-shape includes the hand configuration formed by the fingers and thumb. For example in SASL, the hand configurations roughly correspond to numbers and letters of the English alphabet. The alphabet is extended by modifications of the hand configurations such as: *bent B*[6] or *flat O*[7]. As previously discussed in subsection 2.2.2 some signs are based on the first letter of an English word. This shows the significant role of the hand-shape in distinguishing signs from one another. Furthermore some signs can be similar and differ only in hand-shapes. The importance of hand-shape can be seen in fingerspelling where a sign is unknown and the letters of the sign are spelt instead. Given the significance of the hand-shape in a sign, it is vital that the hand-shape is correctly rendered.

2.3.2 Orientation

The hand can be orientated in two orthogonal directions or plane, i.e horizontal or vertical. In each plane the signer can either see the front, back or the side of the hand. Thus there are only six possible directions: front, side, back, up, down and side. In SASL the signs *live* and *now* are similar but differ in orientation. They use hand-shape Y and show it horizontally with the hands facing up for *now* and down for *live*. This acknowledges the importance of rendering the correct orientation in a sign.

2.3.3 Location

Ranges of locations in SL are limited by the reachable workspace of the hand. Its location is almost always above the waist and in the area in front of the signer and on both side of the head. In most cases the location of the sign contributes to its meaning. For example the sign *doctor* in SASL is performed on the left and the right side of the chest, which refers to the stethoscope. Furthermore the sign *ear* and *eye* are performed by touching the ear and eye respectively.

2.3.4 Movement

Hand movement is a key parameter of SLs. Many meanings of signs may be described through movement. The movement may be used to resemble the doer or the action. In SASL the sign *help me* and *help you* are similar but differ in movement. In the sign *help me*, the movement moves towards the signer, indicating that the signer is offered help. In the sign *help you*, the movement move away from the signer. Size of a referent can be described by the size of the movement. Similarly the speed and rhythm of the referent

[6] The fingers are extended and bent at the base

[7] Four finger are bent straight at the base with the index and middle finger touching the thumb

can be indicated by movement. For a slow action the movement will be slow, and fast for faster action. Rate of occurrence or plurality can be indicated by a repetitive movement.

2.3.5 Non-manual features

Non-manual features refer to facial expressions and upper body posture. Facial expressions include among others raising eyebrows, wrinkled forehead and eyes blinking. In spoken languages, emphasis is done through the tone of voice. In SL emphasis is done through non-manual features. Non-manual features in SL play a significant role to convey meaning. The non-manual feature can be performed on its own, or together with a sign to add emphasis to its meaning. Figure 2.2 shows a sign for how big an object is. This is determined by how much you puff out your cheeks. Facial expressions reflect emotions and can be used in questions. For example, yes/no questions require raising eyebrows and tilting head forward. *Wh*-questions (*who?, what?, when?, where?, why?*) use quizzical facial expressions therefore furrowed brows are required. A change or absence of facial expression can express an entirely different meaning.

Figure 2.2: Puffy cheeks indicating how *fat* an object is [50].

2.4 Sign Language notations

It is known from Section 2.2.4 that SL can be written using SL notations. SL notations are writing systems used to record SLs. SL notations discussed in this section are Stokoe, HamNoSys and SignWriting. These SL notations are important since the SL gestures performed by an avatar are dependent on them. Given the SL structure and importance of phonemes in Section 2.3, the following subsection study how each SL notation exploit these phonemes for SL generation.

2.4.1 Stokoe

Stokoe Notation was the first phonemic script in the history of ASL. It was developed by a linguist, W.C Stokoe [54]. He originally developed 55 symbols and showed that each sign operated on the parameters of Tabula (tab), Designator (dez) and Signification (sig). Tab refers to the location. Dez refers to the hand-shape. Sig refers to the movement. An example is shown in Figure 2.3. It uses the Latin alphabet and numbers to represent hand-shapes. Glyph icons are used to represent hand positions and movements. It is written linearly from left to right. Symbols are arranged in a standard order: location, hand-shape, orientation and movement. It can be written on a computer.

Figure 2.3: The ASL phrase *don't know* in Stokoe notation.

The sign in Figure 2.3 is a one-handed sign. The first symbol indicates the place of articulation which is on the forehead. The letter *B* indicates a *B* hand-shape. The letter *T* indicates the orientation of the hand. The palm of the hand faces towards the signer. The letter x indicates contact. The inverted letter a indicates that the palm is turned downwards. The upside-down *T* indicates that the palm of the hand faces away from the signer. The illustration of the sign in shown in Figure 2.4.

Figure 2.4: The ASL illustration of *don't know* [17].

This notation system is not practical for Deaf people to use as a writing system. The innovative intent was to differentiate between minimal pairs of lexical terms. It is well suited for notating singular signs in SL dictionaries. It lacks a means of describing non-manual features. It does not have large corpora available. It does not have a sufficient number of symbols to cover all the hand-shapes and locations used in SLs. It provides limited linguistic information in terms of hand-shape symbols. The hand-shape symbols are based on the ASL fingerspelling hand-shapes rather than a more linguistic descriptive categories. For example, the Irish manual alphabet hand-shape for *G* is called an *F-hand-shape* in Stokoe notation. Thus this SL notation is limited.

2.4.2 HamNoSys

HamNoSys (Hamburg Notation System) was created in 1989 at The University of the Hamburg. It was created for study and research purposes. HamNoSys like Stokoe, writes signs linearly from left to right. However it is far more detailed than Stokoe. HamNoSys has been improved to make the description of signs more accurate. HamNoSys version 1 to HamNoSys version 3 only record manual parameters. HamNoSys version 4 records both manual and non-manual features. It records the hand-shape, its orientation, location, movement and non-manual features. The phonemes of signs are recorded in this order: symmetry operator, non-manual features, hand-shape, location and movement. HamNoSys distinguishes about 200 hand-shapes and a set of 60 locations in space in front of the signer. The hand-shapes are constructed in a completely different way from Stokoe. They are classified in an anatomically consistent approach. The symbols have iconic relationship to their referent. An example of HamNoSys transcription is shown in Figure 2.5. In Figure 2.5, the first symbol on the left indicates a *flat* hand-shape.

Figure 2.5: The SASL phrase *thank you* in HamNoSys [31].

The next symbol with a bat-wing shape indicates that the hand is orientated diagonally upwards. The oval symbol indicates that the palm of the hand faces towards the signer. The symbol that looks like the letter *U* indicates the location, which refers to the chin. The arrow indicates movement away from the signer. An illustration of how the sign is performed is shown in Figure 2.6.
HamNoSys was not intended for everyday communication in SL, such as writing letters or newspapers. It aimed to provide a written medium for researchers on SL to record signs. It is language-independent. It is applicable to any SL. HamNoSys is phonetic rather than phonemic. It does not describe in depth the possible degrees of the thumb extension and flexion [3]. A systematic grouping-of-signs-by-finger selection is not provided. Some hand-shapes differ by the extension of fingers and thumb.

2.4.3 SignWriting

SignWriting was invented by a dancer, Valerie Sutton in 1974. Sutton with a background as a dancer rather than a linguist developed SignWriting with no prior knowledge of any SL. Without any linguistic theoretical assumptions, the system was developed as a writing tool for recording the body movements. The purpose of SignWriting is to create a daily

Figure 2.6: The SASL illustration of *thank you* [17].

writing system for Deaf and hearing signers who use SL as their primary language and to preserve the native SLs around the world. The system includes symbols for writing hand-shapes, movements, facial expressions, spatial relationships and punctuation. It can be used to write any SL and can be written by hand or by computer using for example SignText Editor[8]. SignWriting captures all possible configurations a hand can make. The possible hand configurations are arranged into 10 groups that match the ASL number 1-10. In each group, fingers can be configured in different ways. There is no specific symbol for location. Instead each written sign uses a two dimensional space as a map of the human body. The symbols of hands are placed within that space so that vertical and lateral dimensions of locations are completely iconic. SignWriting places the hand-shape and movement symbols in a spatial configuration with a symbol for the head to encode locative parameter instead of using a linear sequence of phonetic symbols. This approach is intuitive. SignWriting also has examples of systematic motivation. Hand-shape symbols use the principle that the back of the hand is black and the front is white. An example of a sign transcribed in SignWriting is shown in Figure 2.7.
The circle indicates the head. The hand-shape is a *flat* hand configuration. The asterisk indicates contact. The arrow indicates a right hand moving forward. SignWriting contains over 600 symbols and describes all the phonemes of SLs. SignWriting is ideal for learning

[8]SignText Editor developed by Steve Sleivinski is a text-editing program and a web application for writing signs in SignWriting even sends them in email.

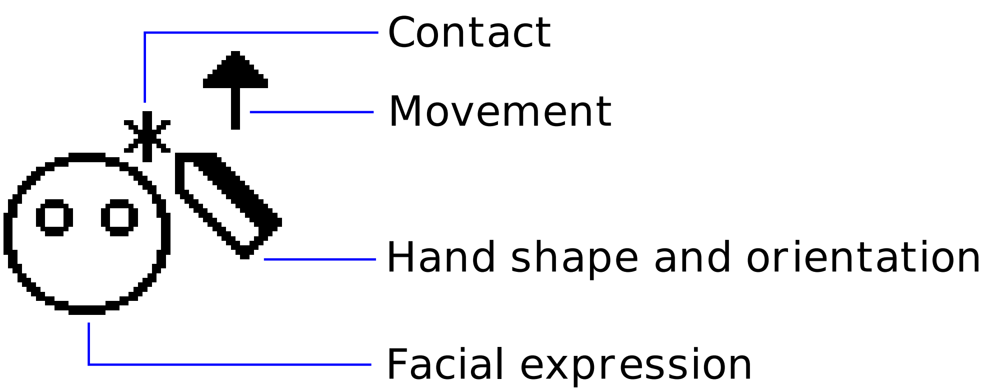

Figure 2.7: The SASL phrase *hello* in SignWriting [56].

SL grammar. The notation is pictorial and descriptive.

2.4.4 Comparison of the SL notations

HamNoSys and Stokoe are for the most part phonetic, including symbols to represent hand-shapes and movement, and in some cases, location, orientation and non-manual features. The symbols are different for each system. With the exception of SignWriting, Stokoe and HamNosys have a set of rules for ordering the phonetic symbols in a linear order. However this convention differs from one system to another. Stokoe and HamNoSys are not intended as practical ways of communicating in a written form of SL. They are intended to represent signs and signed utterances for linguistic analysis. SignWriting is a practical way of communicating in the written form. SignWriting is phonographic. It is chosen as an input notation for the animation of an avatar in this thesis based on:

- Ambiguity - no symbols with the same meaning
- Completeness - captures all the phonemes of SL –(hand-shape, orientation, movement , location and non-manual features)
- It can be transcribed on a computer
- Large corpora available
- Well maintained system
- Easy to use - writing and reading

SignWriting sustains ambiguity between the signed and written form. It can illustrate as much phonetic details as required. Furthermore the number of symbols is adequate to transcribe all the signs. SignWriting has been used to write a lot of documents used on a daily basis such as newspapers, magazines, dictionaries, and children's literature[9]. Novice signers learn signs and SL grammar by using this SL notation. In schools, it is used to teach other subjects, such as Mathematics, History or English.

2.5 Summary

This chapter has shown that SLs are independent languages and are not universal. The structure of SLs was studied. It was shown that phonemes of SLs are not linear but simultaneous and must be correctly rendered in a sign. In order to produce animated signs using an avatar a written form of SL is needed. Thus SL notations were compared to choose an SL notation suitable for the intended system. SL notation used for SL generation should capture all the phonemes of SL. SignWriting was chosen as SL input notation for the intended system.

[9] Available at http://www.signwriting.org/library/

Chapter 3

Issues relating to SL

Given the spatial and visual modality of SL, SL generation is complex. Four issues relating to SL: representation, data, consistency and linguistic use of the signing space are examined.

3.1 Representation

Representation of SL in a written form is necessary for SL generation when employing virtual signing. While this issue can be solved by using any of the SL notations discussed in section 2.4, it must be noted (as previously mentioned) that none of these notations have been accepted as a standard written form of SL. It was decided to use one of the SL notations namely SignWriting. It must generate a SL video using virtual signing and must be able to exploit the linguistic structure of SL.

3.2 Data

The lack of a standard written form of SLs poses challenges in the collection and storage of large corpora. A corpus is more than just a collection of texts. It is assembled in a logical manner. It undergoes linguistic processing to produce sufficient detail for natural language processing. Spoken languages have a large collection on corpora available such as Corpus of Contemporary American English (COCA). The availability of SL corpora is limited. This could be because SLs are commonly not presented in a written text. Traditionally native signers are recorded performing SL from which researchers collect video-based SL corpora [7] [11] [15]. They annotate the videos to record the sequence of body movements. The visual-gestural modality of SLs poses challenges in recording the manual and non-manual features of SL from a video. Motion-capture technology discussed in 4.3.1 is better suited to capture body movements. There are no pre-existing text resources that can be used to create a SASL corpus for linguistic research. There are no standard datasets that can be tested again. This poses challenges in the evaluation of the generated SL. Signs selection and data collection becomes a tedious task because the written form of the signs do not exist, or are merely written in English.

3.3 Consistency

Consistency plays an important in maintaining a smooth transition between signs in a SL sentence. In order to achieve consistency in a SL video all features: cameras, native signers and any other variables should be the same throughout the videos. The signers clothes and the background must also be consistent. The signers must be fluent in SL in order to produce understandable signs. There are minor signing variations that occur from signer to signer that may affect the smooth transitions from one sign to another in a SL sentence. This issue can be solved by using one native signer and keeping all the other features consistent throughout the videos. The signs must be merged with no delay in transitions between the signs in a SL sentence.

3.4 Linguistic use of the signing space

Spatial reference

The visual-gestural modality of SL allows signers to use the signing space. The signing space is used for grammatical and descriptive purposes. Signers use the signing space to structure imaginary placeholders that denote entities under discussion [18][41]. Objects and persons under discussion are assigned to these placeholders. The placeholders function grammatically. The entities are referred to by pointing to the placeholders. Signers do not have to re-describe the entities. The possessive and reflexive pronouns are indicated by pointing to the placeholders. Some verbs will move in the direction of the placeholder. Some will move away to describe an action or event.

Verb inflection

A verb can change its movement path to imply a subject or an object. Such a verb is referred to as an inflecting verb [43]. Each verb has a standard movement path. In general the direction of the movement path moves from the subject to the object. Figure 3.1

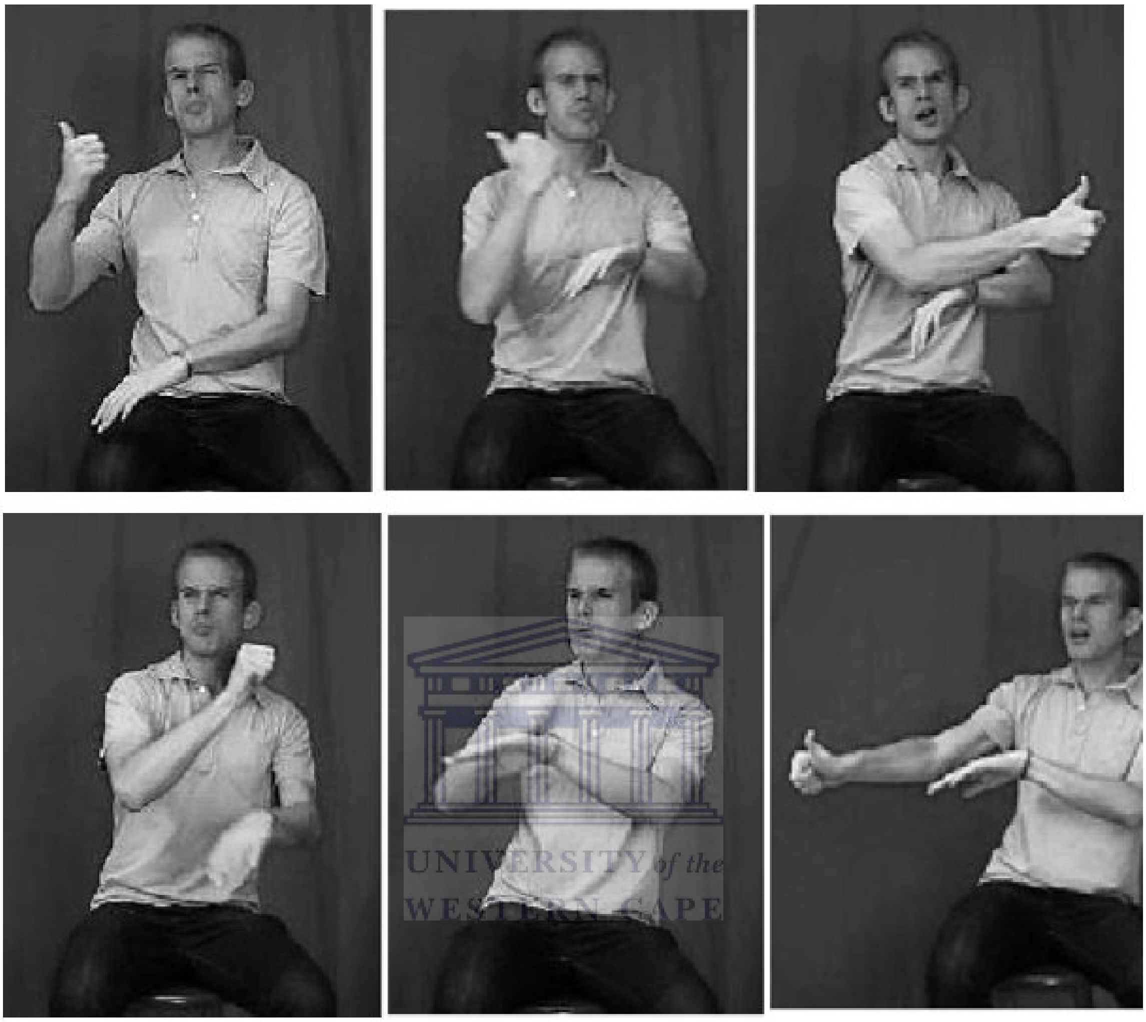

Figure 3.1: Inflecting verb *"blame"*. First row: the subject is on the left and the object on the right. Second row: the subject and the object are swapped [35].

shows an example of an inflecting verb. The first row shows the subject on the left and the object on the right. That is the subject blames the object. In the second row the subject and the object are swapped.

Classifier predicates

Some hand-shapes are used to produce signs that indicate certain semantic features of entities. These hand-shapes are called classifiers [20]. Signs that contain classifiers are called classifier predicates [34]. The movement of classifier predicates transform in a 3D representative way. This movement resembles the manner in which the real-world entity actually moves. Figure 3.2 shows examples of classifier hand-shapes.

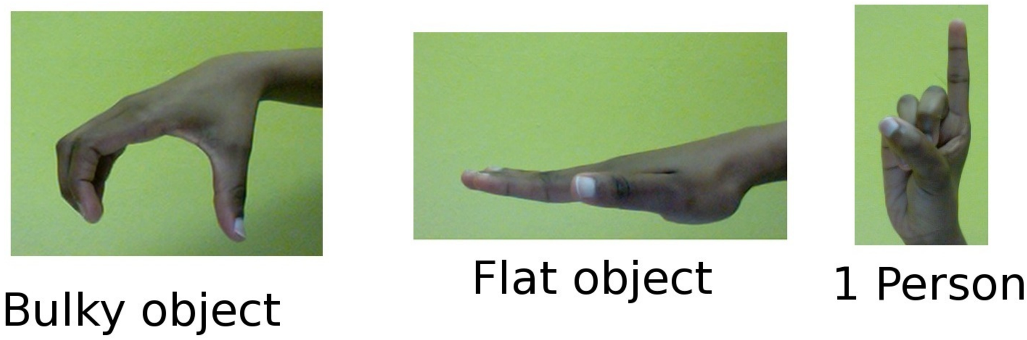

Figure 3.2: Hand-shape classifiers for different object.

3.5 Summary

The issues that relate to SL generation have been discussed. While some of these issues can be solved, the lack of SL corpora poses even more challenges. SL corpora forms the core of SL research. SL data available is either closely guarded or limited to specific topics and limits the extend to which evaluation of SL systems can be tested.

Chapter 4

Sign Language Visualisation

This chapter discusses existing methods that visualise SLs. SL visualisation is a method used to communicate a message in SL and employs communications suitable for the Deaf. These methods are subdivided into: video-based, which records native signers performing SL; and avatar-based, which generates an avatar that performs SL animations. The next two sections describe these methods in detail. For each method advantages, including those that circumvent the issues relating to SL visualization, will be mentioned, as well as its disadvantages.

4.1 Video-based Visualisation

A video-based visualisation system records human signers performing SL gestures. This approach is commonly used in SL dictionaries. Examples of video-based visualisation systems can be found in [11] [15]. The following subsections explores the advantages and disadvantages of this method.

4.1.1 Advantages of video-based visualisation

It is easier to record a large number of SL gestures. In a simple one-to-one mapping of video to SL phrase, a word is looked up in a dictionary to display the video of the corresponding entry. This approach does not require a textual representation of SL which eliminates the issue of the written form of SL discussed in 3.2. The native signers produce natural and realistic performances that include manual and non-manual features.

4.1.2 Disadvantages of video-based visualisation

Videos use a large amount of disk space for storage [22] [53]. The kind of equipment used produces high quality videos to capture the realistic human performances and facial expressions. In order to maintain this high quality over transmission a large amount of bandwidth is required [38]. The pre-recorded videos are limited to a search capability for finding the correct video that corresponds to the English text. Therefore, when a pre-recorded video, needs to be updated or changed, the video must be recorded again. A video-based visualization system is suitable where there are a finite number of sentences. It is suitable where it conveys a message that is known in advance.

The concatenation of signs from other videos to make a new SL sentence in one video is very difficult [53]. There are three major challenges that pose difficulties in the concatenation of these videos.

1. The first challenge is to ensure that the transitions between the signs in each video are smooth. The facial expressions must be blended appropriately with the corresponding signs.

2. The second challenge is to handle the classifier predicates (discussed in Section 3.4) and their movement paths which are not known in advance.

3. The third challenge is to handle the numerous possible movements of the inflecting verbs discussed in discussed in Section 3.4. All the possible inflecting verb signs would need to be pre-recorded. An example of an inflecting verb is *give*. The possible combination of the subject and the object will have to be pre-recorded. For example *I give you, you give me, she gives me, he gives me*. The hand-shape changes to resemble the item given. Each possible combination of the subject and the object must be recorded with each of the possible hand-shape classifiers. The given item can be books, plates, box. Non-manual features can be used to indicate syntactic properties, such as the item given being very small or very big. Each of the possible combinations of subject and object, for each of the possible hand-shape classifiers, must be recorded with each of the possible non-manual features.

4.2 Avatar-based Visualisation

An avatar is a three dimensional computer generated being. Avatar-based visualization systems use avatars rather than human signers to visualise SLs. They use a more complicated procedure to translate spoken languages into SLs and use textual representation of SL notations for animation. The next two subsections explores the advantages and disadvantages of this method.

4.2.1 Advantages of avatar-based visualisation

Avatars can be modified to suit the user's preferences. They can be made to be males or females. The animations can be appealing to children for use in educational applications. Visualisation can be viewed from different viewpoints or at different speeds. Animations can be changed through the SL notations. The avatar-based approach circumvents the issue of consistency by keeping the signer and environment uniform throughout the videos. SL sentences with complex verb inflections, spatial reference and classifier predicate can be visualised to comply to the grammatical rules [19] [22] [28] [45]. Using SL notation, the avatar-based approach can visualise any particular inflected verb by changing its movement path to imply a subject or an object.

4.2.2 Disadvantages of avatar-based visualisation

There are two major challenges when using avatars. The first challenge is to model an avatar that is articulate enough to perform SL gestures [56]. Such an avatar has been developed in [56] and forms part of the work in this thesis. The avatar should be as natural as possible [19] [56]. It should be able to perform gestures as subtle as changes in facial expressions and should be able to smile, raise eyebrows or frown. The second challenge is to animate the avatar. The avatar movements should be physically plausible and realistic in order to be interpreted and be understood by humans.

4.3 Avatar animation

Section 4.1 and 4.2 showed two methods of visualising SLs and that despite the natural and realistic performances produced in a video-based method and avatar-based method is more advantageous. It circumvents most of the issues relating to SL visualisation. Nonetheless, the animation of the avatar poses challenges. In the next subsections two primary animation techniques namely motion capture and the key-frame technique are discussed.

4.3.1 Motion capture technique

Motion capture is performance-based. It records the movements of a live signer using specialised equipment. It uses magnetic or vision-based sensors. The data is then used to animate the avatar. Some motion capture equipment include wearable suit or gloves with potentiometers to measure body and hand movements [55]. Others use optical markers and specialised cameras to record face and body movements [55]. Motion capture is popular for the ease with which it captures human actions. The following subsections state the advantages and disadvantages of motion capture technique.

Advantages of motion capture

- It captures the realistic and lifelike human performances.
- It can obtain real time results and supports automated analysis of the data.
- It is easy to capture complex or long performances. The complexity or the length of the performance is not correlated with the amount of work needed to produce the animation.
- It can accurately capture weight, exchange of forces and secondary movements.
- It can accurately capture difficult-to-model physical movements.
- If an error occurs it is easier to re-capture the performance as opposed to attempting to edit the data.
- It can capture over 100 samples per second which approximates a more continuous representation of the movement. This allows a complete and precise visualisation of gestures.

Disadvantages of motion capture

- It requires specific hardware such as specialised cameras to capture facial and body movements. Other specialised equipment include optical markers and wearable suit or gloves with potentiometers which are used to obtain data.

- The equipment is expensive [28] [32] [55]. The cost includes software, equipment (video cameras, lights) and personnel. Advanced software, video cameras and techniques are upgraded every few years. The technology can become outdated.

- It is time consuming and difficult to set up and calibrate motion capture equipment [28] [32] [55]. Some capture systems have special needs for the environment in which they are used, depending on the camera field of view or magnetic distortion.

- The recorded content is limited to what can be performed during capturing.

- If the proportion of the signer and the avatar differ, discrepancies may occur. For example if an avatar has over-sized hands, the rendering by the avatar may show the hands of the avatar going through its body.

4.3.2 System that uses the motion capture technique

TESSA

An example of a system that uses motion capture technique for the animation of avatars is TESSA (TExt and Sign Support Assistant) [10]. TESSA was developed to enable post office counter-clerks to communicate with Deaf customers. It translates the clerk's speech into British Sign Language (BSL). An avatar is used to visualise the BSL gestures. It uses automatically generated SL gestures. It uses a limited set of pre-defined phrases of most popular transactions used in the post office [10].

It captures movements using separate sensors for the hands, body and face. Cybergloves with 18 resistive elements for each hand are used to record the finger and thumb positions relative to the hand. Polhemus magnetic sensors are used to record the position of the upper torso. Facial expressions are captured using a helmet mounted camera with infrared filters and surrounded by infrared light emitting diodes to illuminate the face. 18 Scotchlight reflectors are stuck onto the face, in regions of interest such as the mouth and the eyebrows. Figure 4.1 shows the motion capture equipment that is used in TESSA. The system is calibrated at the beginning of each session [10], with variation depending on the signers. The interference between glove sensors depends on how tightly the gloves fit. The positions where fingers just touch the thumb must be well adjusted to produce accurate signing and reduce computation times. The sensors are sampled at between 30 and 60 Hz. The movement data captured from the live signer is used to animate the avatar. The system was tested and the results indicated that the accuracy of identification of the signed phrases was 61% for complete phrases and 81% for sign units. 70% of the errors

Figure 4.1: Motion capture using optical markers [10].

found were due to unclear signs. The domain in which the system is used is limited to Post Office, which is small and easy to predict the phrases.

4.3.3 Key-frame technique

Key-frame animation is an animation technique of specifying a set of intermediate key-frames that approximate the expected movement. The avatar is animated according to the sequences of the pre-defined key-frames. It requires the animator to specify the key-frames that define the movement. The movement is obtained by interpolating between the key-frames. Interpolation generates the missing frames between the key-frames to complete the animation. The avatar can be controlled by using kinematics or dynamic constraints in order to have more realistic movements. The definition of a movement requires in-depth details of the movement. In this way the interpolation curves can produce the specified movement. The movement should avoid collision between the limbs of the avatar. The work in [40] describes an interpolation method that produces autocollision-free paths. The following subsections state the advantages and disadvantages of key-frame technique.

Advantages of key-frame

- It requires less disk space for storage than motion capturing. The key-frames can be optimised.
- It is inexpensive to create animation data.
- The animator has total control of animation. It allows artistic expressiveness by providing control to the animator.
- Some interactive modelling packages provide automatic key-framing features to aid in the process [2] [6] [8].

Disadvantages of key-frame

- It can become complex and time consuming as it is dependent on the number of animation parameters that must be controlled.
- Modelling packages that can be used for key-framing requires some training.
- Difficult to specify realistic interactions.
- Difficult to specify large, dynamic movement.
- Modelling packages do not provide dictionary look-up facilities that can aid in the creation of complex movements by combining stored key-frame data [2] [6] [8].
- It requires significant effort from the animator. It is not independent of the human creator and thus is not flexible.
- The animator has to focus on avoiding collisions.

4.3.4 Systems using the key-frame technique

Vsigns

Vsigns is an example of a SL visualisation system that uses SignWriting as a key-framed based input. It is a Greek project developed by Papadogiorgaki et al. [44] at the Informatics and Telematics Institute in Greece. VRML and MPEG-4 animation are used to animate the avatar. The system interprets and converts SWML to MPEG-4 FBAP which is generated as a VRML animation that is compatible to an H-Anim avatar [44]. There

Figure 4.2: Avatar used by Vsigns [44].

are three different avatars that can be used to sign the constructed gestures one of which is shown in Figure 4.2. Some points of criticism are:

1. No documented experiments on the system.
2. A limitation is that only some facial expressions and animation symbols are supported but facial expressions, e.g. cheek wrinkles, have not been implemented.
3. The MPEG-4 FAP suggested for facial animations is limited because it is avatar-dependent.
4. VRML animations are complex because they include CoordinateInterpolator nodes, which increases the computational demands for the hardware in order to produce animations.

5. Another criticism is the use of an H-Anim avatar, which has hand bones that are not flexible enough for the hand-shapes used in SLs [16].

SignSynth

SignSynth is an example of a SL visualisation system that uses Stokoe notation as input. It is a free, non-profit and open source project developed by Angus Grieve-Smith at the University of New Mexico [22]. It uses ASCII-Stokoe as input in order to animate an avatar that can produce manual and non-manual SL gestures. The ASCII-Stokoe is parsed with a Perl script which is provided as a key-frame to the animation module [22]. The system outputs a VRML file containing the avatar as well as the animation data. The technologies used in this system are Web3D technology and VRML. There is no documentation provided on the development and evaluation of the system in [22]. Points of criticism are:

1. According to the author, the system is not flexible because it relies on ASCII-Stokoe.
2. ASCII-Stokoe does not denote all the hand-shapes, locations and movements used in SLs.
3. An H-Anim avatar is used which has hand bones that are not flexible enough for the hand-shapes used in SLs [16].

SASL-MT

The SASL-MT project developed at the University of Stellenbosch translates English to SASL [19]. A generic pluggable avatar system that use an H-Anim compliant avatar was developed [19]. SignSTEP was designed as input to control and animate the avatar. An H-Anim avatar used in the system is shown in Figure 4.3.

Scripting Technology for Embodied Persona (STEP) is a scripting language for embodied agents, in particular for communicative acts of embodied agents. The SignSTEP notation is based on the STEP and XSTEP notation for humanoid animation [27]. This system was designed to be more generic. The modules are kept independent from one another by restricting interaction between modules to clearly defined interfaces. The system uses an interface notation which resides between the input notation and the animator. The notation is primarily a list of joints and their corresponding movements. It also includes temporal information that consists of rotation speed and start times for the various joint movements. It uses Java3D to implement the avatar system. The animator receives input from the interface notation converted into animations, as well as input in the form of an

Figure 4.3: Avatar used by SASL-MT [19].

avatar model that needs to be animated. The system provides pluggable input notation and pluggable avatar functionality. Some points of criticism are:

1. It uses a VRML file loader to implement the avatar animator. The disadvantage of using a VRML file loader is the computational overhead of the loading process [19]. The VRML file loader is easier to implement and offers a cleaner and more elegant design at the cost of an increased loading time.

2. The input notation developed, SignSTEP, forces the author of the script to have knowledge of the coordinate system that the animator uses [19].

3. The most important functionality that is still lacking from this animation system is support for facial expressions [19].

4. The H-Anim standard that is used as a standard for all avatars only supports severely limited joints for facial animation [19] and has hand bones that are not flexible enough for the hand-shapes used in SLs [16].

5. The avatars are catoon-like.

4.4 Summary

A system that generates SL includes the grammatical details that are communicated through the use of space. It must be able to dynamically generate all representations.

Video-based visualisation produces natural and realistic performances. However avatar-based visualisation presents several advantages that address the problems associated with SL visualisation. The work discussed in this thesis extends part of the SASL project presented in [56]. Therefore this chapter did not intend to give an extensive overview of SL visualisation, but rather an overview so as to provide context for the rest of the thesis. The limitation of visualisation systems have been discussed. Most of these system use an H-Anim compliant avatar, which does not fully cater for the linguistic use of the signing space, phonological and syntactic rules of SLs. The solution to this limitation is explained in the next chapter. Another limitation was the use of a SL notation that does denote all the parameters of the SL phonemes. In this research SignWriting is used and overcomes this limitation. Most of these systems use VRML which requires a VRML browser. A system that uses a mobile phone as a service-delivery platform has been developed in [21] which will eventually be integrated with the work presented in this thesis. Contrary to the systems investigated, the system implemented in this work was evaluated. The evaluation setup is available in Chapter 7 and the resulted are documented in Chapter 8.

Chapter 5

Design

This chapter addresses the design of the system. It discusses the standards and technologies used. The first section discusses the structure of the avatar. It explains the technologies used to achieve physically plausible poses. The second section discusses architecture of the system. The third section outlines of the SL notation provided as input for SL generation. The fourth section discusses the software used for the animation and modelling of the avatar and the chapter ends with a summary.

5.1 H-Anim Avatar

The previous chapter investigated the different SL visualisation systems that use avatars to perform SLs. Animation system such as that developed for SASL-MT use the H-Anim standard as a framework for avatars. H-Anim [24] is a set of specifications for the description of human animation. It is based on how the body is segmented and connected. It describes a human body as a chain of segments that are linked together through joints such as elbow, wrist and ankle. It recommends relative proportions for the segments which are not compulsory. The main goals of the H-Anim standard are compatibility, flexibility and simplicity. It defines the key body features in a consistent approach. Bones are labeled and positioned consistently on the human body skeleton. The avatar presented in this work has been discussed in [56]. In order to perform complex SL movements, the H-Anim standard was extended and adapted. The first section that follows discusses the limitation of the H-Anim standard. The second subsection discusses how these limitations were circumvented.

5.1.1 Limitations of the H-Anim standard

The H-Anim standard is very flexible but has a few limitations. It does not state where the joint centre should be located or joint rotational limits. It does not specify where to place the joint centres which determine how long the segments should be. It does not specify the joint rotational limits to ensure poses that are possible for humans [56]. The hand bones in H-Anim are not flexible enough for the hand-shapes used in SLs [16]. H-Anim standard only provides a simple set of bones for basic facial expressions. A workaround to these limitations was proposed in [56]. The H-Anim standard was adapted and extended with a slight variation of MPEG-4 FDP facial key points. The MPEG-4 FDP standard fully supports facial animation. The MPEG-4 FDP standard is a set of body and facial animation parameters that can be used to animate avatars using body animation parameter (BAP) and facial animation parameter (FAP) players. To take full advantage of the advanced facial animation supported by the MPEG-4 SNHC standard, the facial model of the H-Anim avatar needs to be altered and this is discussed in the following subsection.

5.1.2 Adapting and Extending H-Anim

In [56], the author discusses the MPEG-FDP and H-Anim standards widely adopted to satisfy the SL requirements. The avatar should control its body movements in the same way a human signer does with his hands, arms, fingers, body and head movements. All bones were given limits for their rotational Degree of freedoms (DOFs) to ensure physically plausible poses during animation. Figure 5.1 shows the physical plausible poses of the avatar. The structure of the H-Anim Level of articulation (LoA) 2 skeleton was extended by adding facial bones to the skeleton and model based on the MPEG-4 FDP facial feature points. The bones added for the face follow a different but more informative naming convention than that proposed by the H-Anim specification. The reader is referred to [56] for more information on the adaptation and extension of the H-Anim standard. In [56] the author only takes the structure and the naming interface used by the H-Anim standard. An H-Anim compliant application was not developed. Given this extension of the H-Anim standard, it is able to provide for the linguistic use of the signing space, phonological and syntactic rules of SLs.

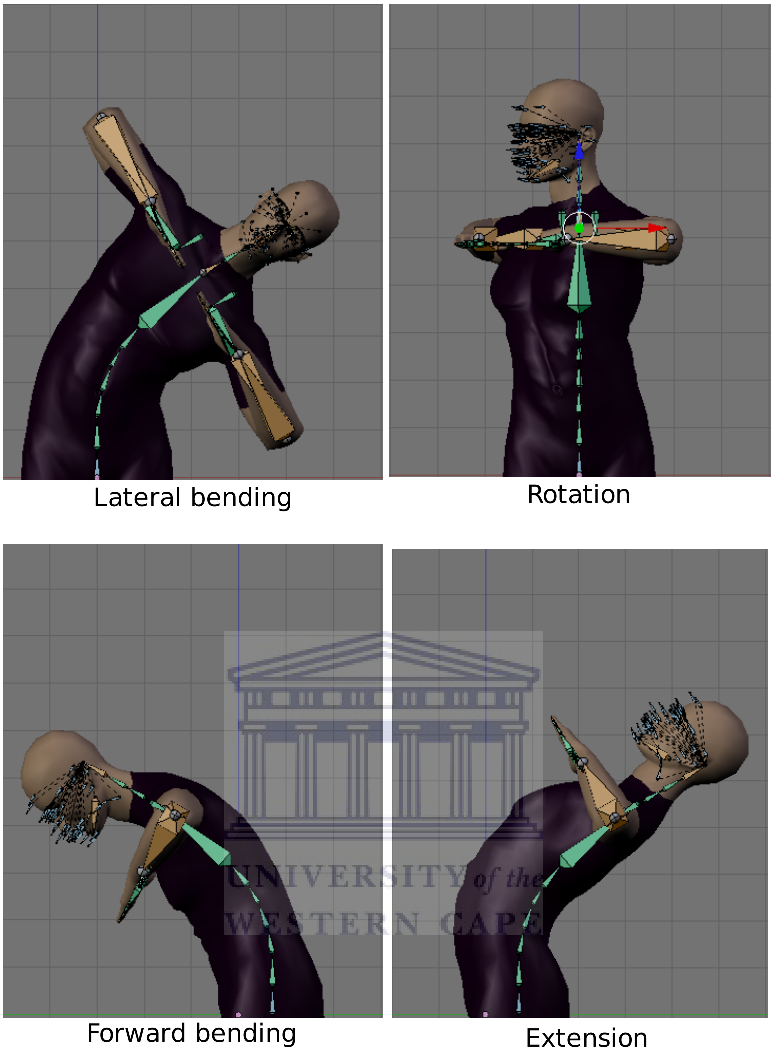

Figure 5.1: The physical plausible poses of the avatar [56].

5.2 System Architecture

The main components of the system are the input notation and the SL generation. This section combines these components to illustrate the system overview. The SignWriting notation is provided as input and is converted to (SignWriting Markup Language) SWML which is sent to the parser. The SWML input is analysed and semantic information about the phonemes of the sign is extracted to generate animation parameters. These animation parameters are then used to produce SL gestures corresponding to the input. The output produces a natural movement of the avatar performing the SL gestures. The process developed builds an animation in four steps and the overview of the system design is shown in Figure 5.2.

1. **Conversion of SignWriting notation to SWML**
   The SignText Editor is used to convert SignWriting into SWML.

2. **Parsing of the SWML**
   SWML is parsed and validated against the DTD. The sign is partitioned into two phases. The sign is analyzed and validated. Then the sign is dismantled into separate key poses. The key poses are sequenced into individual kinematics problems.

3. **Generation of the animation parameters**
   The individual kinematics issues are resolved. Key-poses are generated based on the key poses defined in 2. The key poses are merged into a complete smooth transition of poses.

4. **Exporting the animation**
   The animation is exported to a standard video format. An avatar performing the SL gestures is displayed.

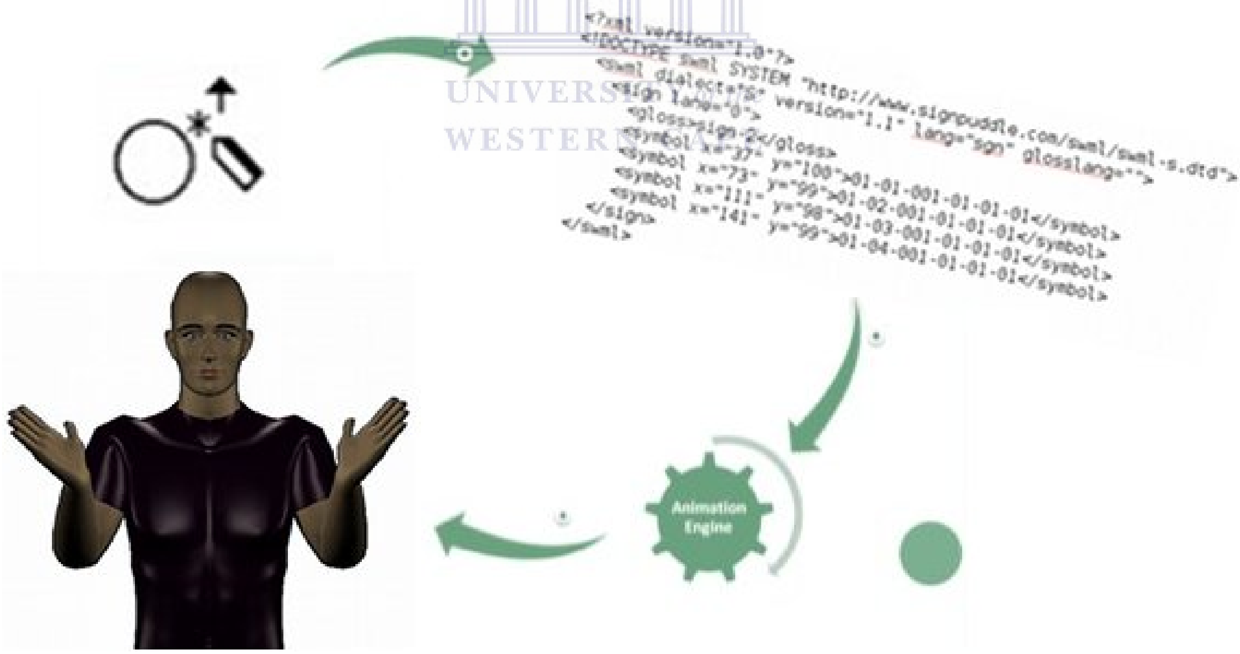

Figure 5.2: An overview of the system design.

5.3 Input Notation

The graphical nature of the SignWriting symbols in Section 2.4 poses challenges in parsing for SL generation. The symbols are usually rendered as GIF files. In order for these symbols to be used in an SL generation system they must allow processing, storage and indexing. It is for these reasons that SWML was proposed. The first section explains what SWML is and its symbol structure is discussed in the second subsection.

5.3.1 SignWriting Markup Language

SWML was developed by [12]. It defines a set of XML tags for each symbol in SignWriting. It is an XML-based language for encoding SL text written in SignWriting. SWML describes SL gestures in a manner ideal for input to a signing avatar. The files are represented as plain text which require less storage. They allow for fast and efficient transmission. It can be used in language and document processing such as translation and animation [44].

The version of SWML used in this system is version 1.1 defined by the DTD included in Appendix A. The symbol set used in this system is the International SignWriting Alphabet (ISWA) 2008. Each symbol in the ISWA 2008 corresponds to some phonological aspect of SL. Section 5.3.2 explains how the symbols are labelled to identify which phonological aspect of SL they refer to.

5.3.2 Symbol structure

Each symbol is identified by a unique 6-tuple ID such as "*01-01-002-01-02-08*". The symbol ID is defined as:

$$s_{id} = \{c - g - b - v - f - r\} \tag{5.1}$$

The symbol ID in Eq 5.1 identifies the transformation to which the symbol was subjected when included in the sign. For example the symbol ID specifies that a symbol is a hand symbol, the hand-shape, the orientation and the rotation. The fields of the symbol ID are explained as follows.

- c = category. The symbols are categorised as: hands, movement, face-and-head, body, dynamics, punctuation and location for sorting. The c field identifies which category the symbol belongs to. For example c = *01* refers to the hands category.

- g = group. Each category is further divided into groups. For example in the hands category there are ten groups because the hands has 10 fingers.
- b = baseSymbol. It specifies the shape of the hand within a group.
- v = variation. It indicates if the fingers of the baseSymbol are curved or bent.
- f = fill. It indicates orientation. For example it specifies the palm orientation from the signer's viewpoint.
- r = rotation. It indicates rotation. For example, it specifies how the hands are rotated.

```
<?xml version="1.0"?>
<!DOCTYPE swml SYSTEM "http://www.signpuddle.com/swml/swml-s.dtd">
  <swml dialect="S" version="1.1" lang="sgn" glosslang="">
    <sign lane="0">
      <gloss>sign-2</gloss>
      <symbol x="37" y="100">01-01-001-01-01-01</symbol>
      <symbol x="73" y="99">01-02-001-01-01-01</symbol>
      <symbol x="111" y="98">01-03-001-01-01-01</symbol>
      <symbol x="141" y="99">01-04-001-01-01-01</symbol>
    </sign>
  </swml>
```


Figure 5.3: An example of SWML.

The ID of the one of the symbols shown in SWML in Figure 5.3 is "01-01-001-01-01-01". From the symbol ID it can be deduced that the symbol belongs to a hand category, c = *01*. The symbol belongs to a 1 finger group, g = *1*. The shape is an index finger, b = *001*. The index finger is not bent or curved, v = *01*. The signer sees the front of the hand, f = *01*. The hand is not rotated, r = *01*.

Since SignWriting does not denote a symbol for location, so is SWML. Section signwritingss indicated that a sign is written in a two dimensional space. From this arrangement the x and y coordinates are used to determine how the symbols are arranged relative to each other. They are denoted as s_x and s_y. The length and width of a symbol are denoted by s_{length} and s_{width} respectively. Thus the symbol is denoted by s, and is defined as:

$$s = \{s_{id} - s_x - s_y - s_{length} - s_{width}\} \tag{5.2}$$

It is important to note that a sign is composed of a set of symbols and is denoted as:

$$sign = \{s_1, s_2, ..., s_n\} \tag{5.3}$$

5.4 Blender

Section 5.3 discussed the outline of SWML from which SL animations must be generated. This section discusses the software that is used to generate SL animations corresponding to SWML. The software used is this thesis is Blender which is a free open source 3D content creation suite [6]. It offers full functionality for modelling, rendering and animation. It is an advanced key-frame animation system. It has features such as constraints for its skeleton system: forward kinematics (FK) and inverse kinematics (IK) that support the modelling and animation of 3D models. It has an embedded Python interpreter with an application programming interface (API). Python is a high level, multi-paradigm, open source language. Blender has modules for Internet data handling and structured markup processing such as XML [56]. Given all these features Blender is a well-suited software for animation.

5.4.1 Avatar representation and animation

An avatar is built as a 3D surface mesh. A mesh is formed by a linked network of small polygons, and is deformable. It order for the mesh to be displayed using rendering techniques, the coordinates of the polygon vertices mush be known. Underlying, the base of an avatar is a human body skeleton, a hierarchical structure of bones. The skeleton and the mesh are attached together. However, the skeleton is invisible and the mesh is visible. The location as well as the orientation of the polygons is defined based on those of the bones of the skeleton. Thus, a change in the skeleton's posture changes the posture of the mesh. This is the method used in this thesis, to change the manner in which the avatar is posed. The skeleton's posture uses less data compare to the entire mesh.

5.4.2 Python Scripting

Blender has a game engine that allows users to create and play 3D games. This engine allows programmers to add scripts to improve game interaction, control, etc. The powerful feature of Blender is its internal fully-fledged Python interpreter. Blender's functionalities can be added and extended to real-time [6]. The are two main ways of extending Blender; binary plugins and Python scripting.

5.4.3 Python SAX API

Simple API for XML (SAX) retrieves data from an XML document. It is a programming interface for event-based parsing of XML files. XML documents are processed using parsers. The parser verifies that the XML document is well formed by validating it against a Document Type Definition (DTD). DTD is a set of declarations that define the structure of an XML document. It defines the structure with a list of elements. The package xml.sax as well as its sub-packages provide a SAX interface of Python implementation [46]. A SAX application is structured in such a way that it requires input sources, parser and handler objects [46]. The input source is sent to the parser from which a sequence of events on the handler is produced. The SAX API defines four basic interfaces. The SAX interfaces are implemented in the xml.sax.handler module as the following Python classes [46]:

- **ContentHandler:** It implements the main SAX interface for handling document events.
- **DTDHandler:** It handles DTD events.
- **EntityResolver:** It resolves external entities.
- **ErrorHandler:** It reports all errors and warnings.

The basic methods of the xml.sax package are:

- **make_parser()** - It creates and returns a SAX XMLReader object.
- **parse(*filename, handler*)** - It creates a parser and parses the given document. It can be passed as a file object or a stream. The *handler* is one of the SAX interfaces.

5.5 Summary

H-Anim avatars developed do not have all the necessary features to effectively visualise SL. The H-Anim standard is very flexible but has a few limitations. It was extended to satisfy the SL requirements. This system intends to animate a realistic avatar in a natural way. The Blender animation software was used in order to achieve this goal. SWML format was found suitable for avatar animation. This chapter introduced the technologies and framework used in this system. It highlighted the benefits of these technologies.

Chapter 6

Implementation

The previous chapter introduced the technologies and framework used in this system. It highlighted the benefits of these technologies and framework. This chapter discusses the implementation of animating the avatar from the SignWriting notation. It discusses how animation parameters are extracted from SWML. The implementation is influenced by how symbols are represented in SignWriting. The chapter discusses the responsibility of each of the classes of the parser. It discusses how animation parameters are generated for each category of the SignWriting symbols.

6.1 Input notation

The input notation, as already discussed in the previous chapter, is used to drive the avatar. The input is provided to a web browser. The SignText editor is written in JavaScript. The JavaScript communicates with the animation engine to send the SWML input to the parser. It also receives the animation exported from the animation engine. The web browser is used to allow the user to describe the gestures. An external scripting approach is used to interface the input notation and the parser. The JavaScript saves the SWML input file in a local directory where it will be accessible to the parser. An illustration of the is shown in the Figure 6.1.

6.2 Parser

The parser interprets the SWML from which it generates animation parameters. The parser implemented is divided into two classes: *SignSpelling Sequence* and *ActionStrip*.

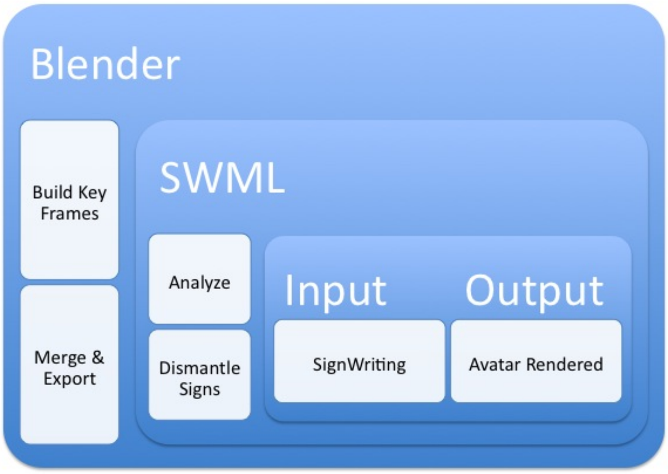

Figure 6.1: A overview of the implementation process.

SignSpelling Sequence is responsible for spell-checking and determining the order of reading the symbols in a sign. *ActionStrip*, is responsible for generating animation parameters. The two classes of the parser are discussed in detail in the next subsections.

6.2.1 SignSpelling Sequence Class

SignSpelling Sequence is responsible for spell-checking and determining the order of reading the symbols in a sign. It first checks that the sign is valid based on the SignWriting rules. It checks if the symbols are used correctly to form a sign. The symbols that refer to the right hand are different from those of the left hand. An example is shown in Figure 6.2.

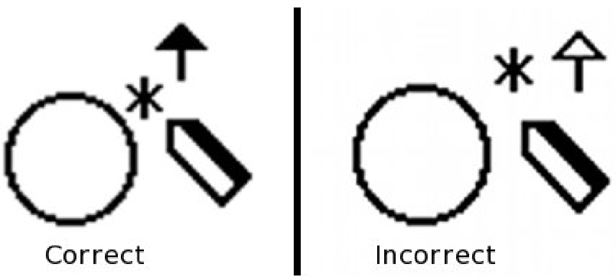

Figure 6.2: A example of signs written correctly according to the SignWriting rules.

The movement symbol (arrow with a white arrowhead) corresponds to the left hand, but a right hand symbol is used instead. There is a mismatch between the left hand movement and the right hand symbol. Thus the sign is incorrect. *SignSpelling Sequence* rejects incorrect signs and cannot be processed. The correct way of writing the sign is shown on the left hand side of Figure 6.2. The movement symbol corresponds to the right hand. The sign is correct because the right hand symbol is included. The rules differ based on how the signs are written. For this reason, it was necessary to distinguish three types of signs. The three types of signs are defined as finger-spelling, basic signs and compound signs. A finger-spelling sign contains hand symbols only. It may include movement symbols as well. A basic sign contains only one sign. It may be mirrored. That is the movement, location, orientation of the left and right hand are the same. Furthermore the movement can be alternate or simultaneous. If the sign is not mirrored, it may not include more than one symbol of the same category. Else, it is a compound sign. A compound sign consists of multiple basic signs.

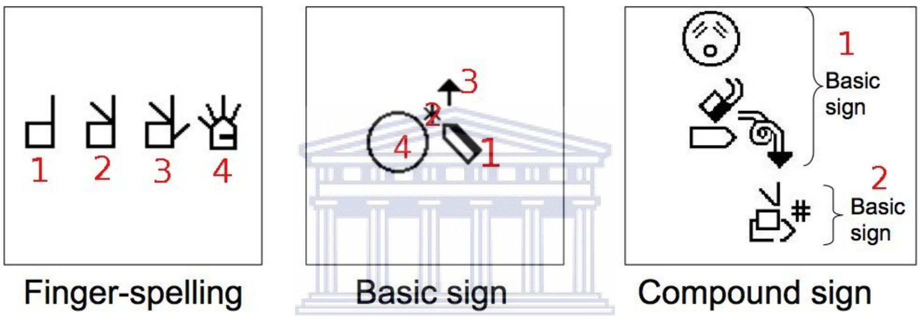

Figure 6.3: Three types of signs

The symbols in a sign are not written linearly from left to right. A sign can be read from any direction depending on the symbols used to compose the sign. Therefore the order of reading the symbols in a sign must be established. In Figure 6.3, the three signs are read differently. The signs are labelled to show how the symbols are read, with label 1 indicating the first symbol read. The compound sign poses challenges in determining the order of reading the symbols. The SignWriting movements are sequential, but are not always read from left to right. The symbols are in visual units relating to the center of the sign. The hand-shape that starts the sign follows the movement symbols to the second finishing hand positions. The centering symbol is used to determine the order of reading the symbols. It is used as a reference to determine the symbols on the right, left, top or bottom.

6.2.1.1 Determining the centering symbol

Figure 6.4 illustrates the sorted list of symbols in a sign. The sorted symbols are read from top to bottom. *SignSpelling Sequence* sends the sorted list of symbols to the *ActionStrip* where the symbol IDs are interpreted and animation parameters are generated.

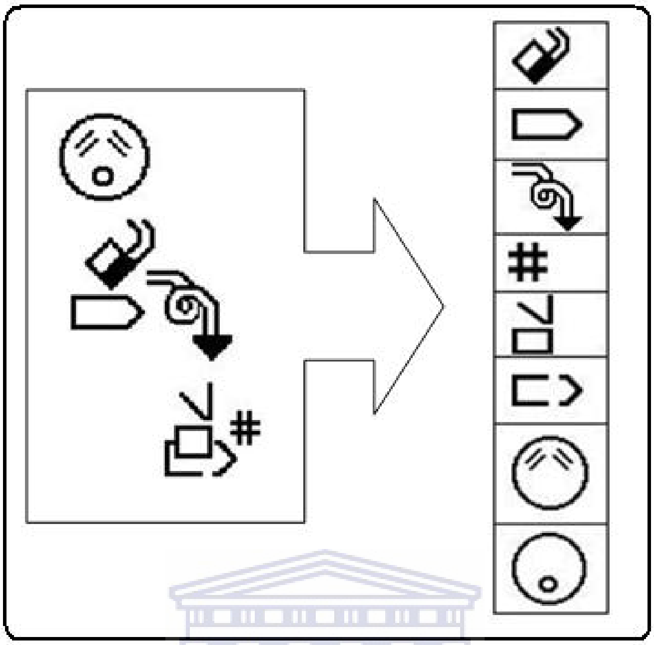

Figure 6.4: A graphical representation of the SignSpelling class. The symbols on the right are organized in the order of reading the sign from top to bottom

The x and y coordinates of the symbol are encoded in SWML, denoted as s_x and s_y respectively. They indicate the position of a symbol in the signbox. The length and width of a symbol are denoted by s_{length} and s_{width}. The coordinates of the center of a symbol is defined as c:

$$\begin{aligned} x_c &= \frac{s_x + s_{width}}{2} \\ y_c &= \frac{s_y + s_{length}}{2} \end{aligned} \tag{6.1}$$

Suppose **C** is the centering symbol. 8 possible positions relative to the centering symbol are defined. These positions are used to determine the symbols at the beginning or the end of the sign. The 8 positions are shown in Figure 6.5.
The direction in which the arrowhead is pointing is determined using the *rotation* field. Generally the symbols at the stem of the arrow begin the sign, followed by arrow then the symbols at the arrowhead. The symbols are then rearranged sequentially in the order of

	Left	Center	Right
Top	1	2	3
Middle	4		5
Bottom	6	7	8

Figure 6.5: Positions relative to the centering symbol.

reading the symbol. The results are indicated in the second column of Figure 6.4. Note that the actual results are in SWML format.

6.2.2 Action Strip Class

ActionStrip is responsible for generating animation parameters. It receives a sorted list of symbols from *SignSpelling Sequence*. *ActionStrip* contains information about each field of the symbol ID. It uses this information to specify the BAPs of the corresponding joints. The subsections to follow explain how the *ActionStrip* class generates animation parameters for each of the three categories.

6.2.2.1 Hands Category

A basic symbol undergoes numerous transformations such as rotation, orientation or mirroring. All basic symbols are right handed with a rotation angle of 0°, with the hand orientated parallel to the wall with the palm facing towards the signer. Figure 6.6 shows how one basic symbol can be rotated and orientated.
The basic symbol is right-handed and the mirrored symbol refers to the left hand. The *rotation* (r) field specifies whether the symbol is mirrored. The right hand is indicated by $1 \leq r \leq 8$. The left hand is indicated by $9 \leq r \leq 16$. The *rotation* field indicates the rotation of the hand, which occurs at a multiple angle of 45°. The left hand is rotated clockwise, where *r=9* represents no rotation. The right hand is rotated anti-clockwise where *r = 1* indicates no rotation as indicated in Figure 6.6. *ActionStrip* is able to extract this kind of information using the fields of the symbol ID. The combination of *c-g-b-v* is used to identify the basic symbol. It determines the animation parameters

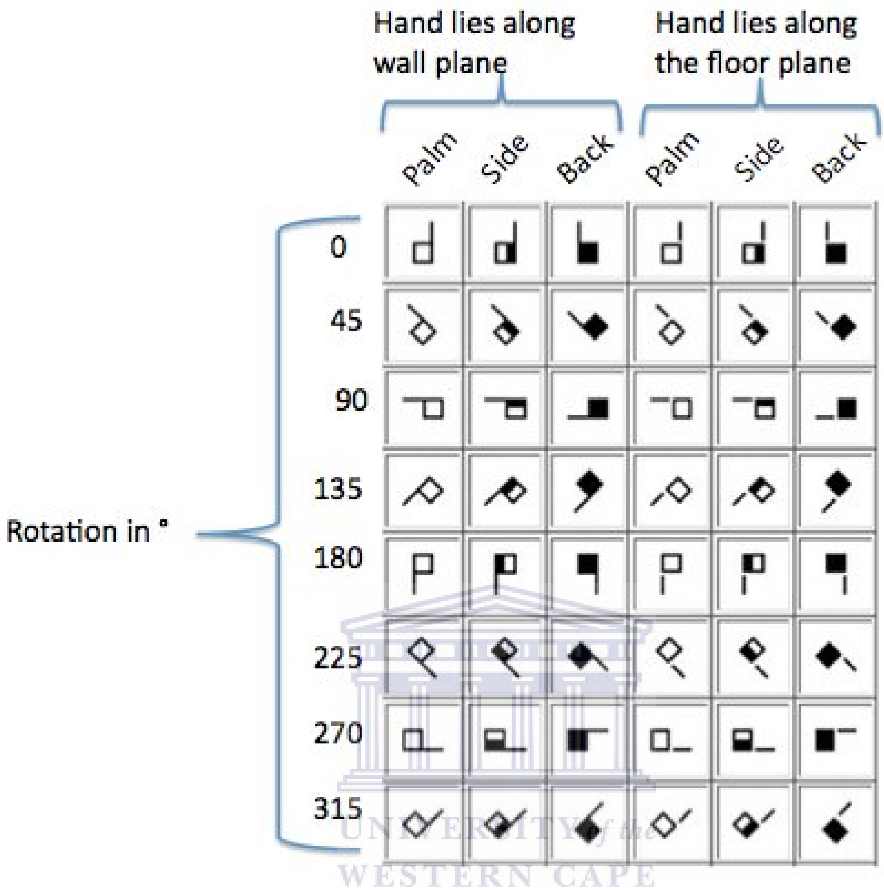

Figure 6.6: The angles of rotation corresponding to each orientation of the right hand.

of the finger joints to obtain a corresponding hand-shape. The hands are very complex articulated structures with multiple DOFs [56]. Figure 6.7 shows the skeleton of the hand.

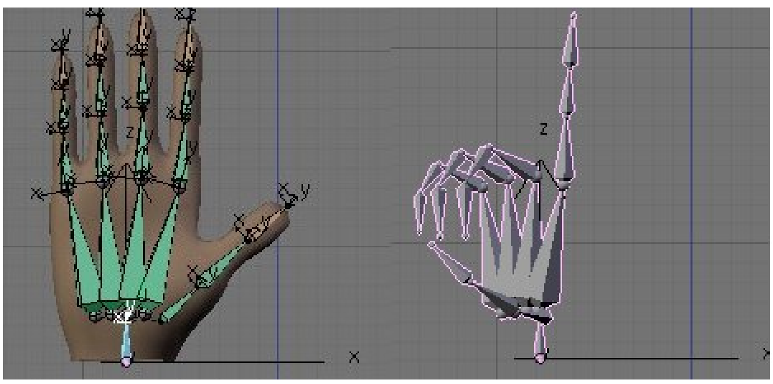

Figure 6.7: The skeleton of the hand of our avatar.

The skeleton of a hand can be divided into three main sections namely the carpal bones (carpals), metacarpal bones (metacarpals) and phalangeal bones (phalanges). The bones colored green are FK enabled. The bones colored brown are IK enabled. The finger joints are rotated to achieve a hand-shape corresponding to the basic symbol shown in Figure 6.8.

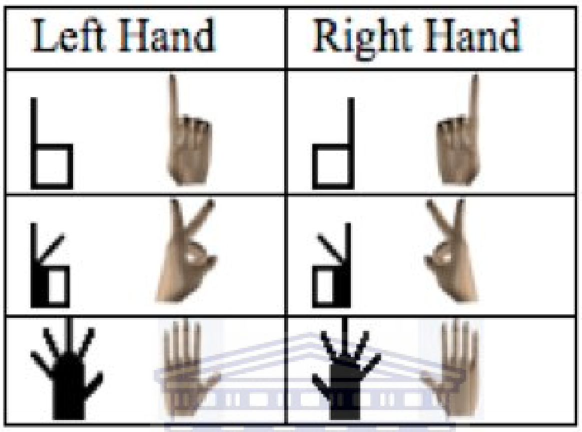

Figure 6.8: The hand-shapes of the avatar corresponding to the basic symbols.

A database of hand-poses that correspond to the basic symbols was created. *c-g-b-v* is used to identify a basic symbol in the database. *c-g-b-v* can distinguish between different hand-shapes. Table 6.1 shows the procedure of obtaining the animation parameters with regards to the hands. After determining the animation parameters of the finger joints, the hand orientation is determined.

Action	**Symbol ID tuple**	**Bones of interest**
Determine left or right hand	*rotation*(r)	none
Determine basic hand-shape	*c-g-b-v*	hand bone
Determine orientation 1	fill(f)	elbow
Determine orientation 2	fill(f)	elbow and shoulder
Determine rotation	rotation(r)	shoulder

Table 6.1: The fields of the symbol ID corresponds to the bones of interest.

Hand orientation is divided into two parts. The first part of the orientation is the direction in which the palm is facing. This orientation is divided into 3 views; front, side and back. It is viewed from the signer's point of view. Orientation is identified by the *fill* field. The second part of the hand orientation is the plane which the hand is parallel to. The hand can be parallel to the wall (vertical) or to the floor (horizontal) plane and is identified by the *fill* field, $1 \leq f \leq 3$ and $3 \leq f \leq 6$ respectively.

Plane	**Direction**	***fill***
Wall	Front	f = 1
Wall	Side	f = 2
Wall	Back	f = 3
Floor	Front	f = 4
Floor	Side	f = 5
Floor	Back	f = 6

Table 6.2: Orientation as interpreted by *ActionStrip*.

Table 6.2 shows how *ActionStrip* interprets the orientation in each plane for every view. The bone rotated to achieve the different orientations is the *elbow*. Figure 6.9 shows the different orientations along the wall and floor plane. As indicated in Table 6.1, the *rotation* field determines the animation parameters of the shoulder to achieve the specified rotation. The *wrist* is also used for rotation.

6.2.2.2 Facial expression Category

In order to achieve the facial expression, the facial part is identified. *c-g-b-v* is used to identify the facial part. It also specifies if for example, the cheeks are puffed or sucked in, or eyes are half opened or wide opened. The face of the avatar contains additional bones added on facial feature points such as eyes and eyebrows as indicated in Figure 6.10. These bones were given no limitation in order to make it easier to perform facial expressions. However, this means that the avatar may produce unnatural facial expression. Figure 6.10 shows some of the facial expressions produced by the avatar. For example the eyebrows in Figure 6.10 are frowned. The facial expression symbols encode information about which part of the face must be moved and how it should move. An approach that is used to determine the animation parameters of the finger joints is also used to determine the FAPs. The *rotation* field indicates the direction in which the facial part must move. For the facial expressions that include eyes, cheeks, eyebrows and ears, the *fill* field is used to determine if the left, right or both facial parts are used. If $f = 1$ the facial expression includes the left and right side of the facial part. If $f = 2$ the facial expression includes

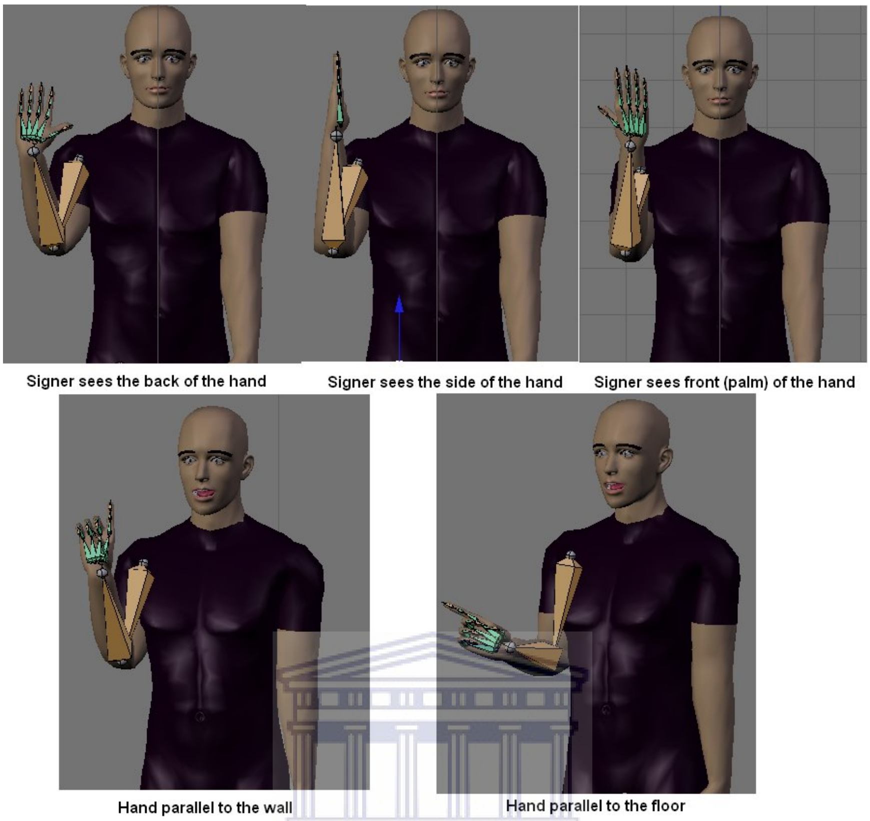

Figure 6.9: The different views of orientation from the signer's point of view.

the right side of the facial part. If $f = 3$ the facial expression includes the left side of facial part. Table 6.3 summarises how *ActionStrip* interprets the facial expressions.

Action	Symbol ID field
Identify facial part	*c-g-b-v*
Identify the left, right or both facial parts	*fill*
Identify the movement	*rotation*

Table 6.3: The fields of the symbol ID corresponds the facial bones of interest.

A database of facial expression was also created. The *c-g-b-v* combination of the symbol ID is used to search the database for the corresponding facial expression.

6.2.2.3 Movement Category

Movement is denoted by arrows to indicate the path and direction. The arrows differ to show the movements associated with the left, the right hand and both hands, as well as

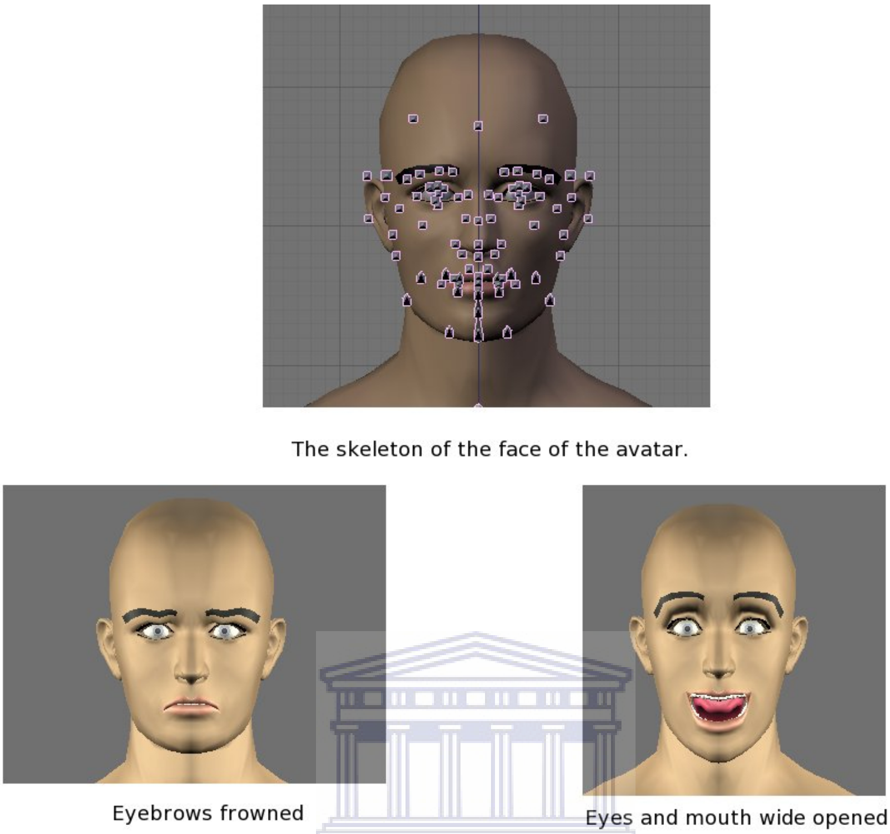

Figure 6.10: The facial expression of the avatar.

the plane in which the movement lies. Figure 6.11 shows the varying movement sign used in SignWriting.

	Left	Right	Both Hands	Curves	Circles
Wall	⇑	⬆	⇑	↷	↻
Floor	↑	⬆	↑	↶	

Figure 6.11: Different movement symbols used in SignWriting. The arrow heads differ to indicate left, right or both hand moving.

The *fill* field is used to identify the hand to which the movement is associated. If $f = 1$ then the movement is associated with the right hand. If $f = 2$ then the movement is associated with the left hand. If $f = 3$ then the movement is associated with the left and right hand. *c-g* indicates whether the movement is along the floor or wall plane. *c-g-b-v* indicates the type of movement. It indicates whether the movement is a straight line, a curve, or a circle (see Figure 6.11). To distinguish between shorter and longer movements, the *variation* field is used. It indicates the length of the movement. The direction of the movement is determined using the *rotation* field. Similar to rotation of the hands, the movement rotation occurs at a multiple angles of 45°. Table 6.4 summarizes how *ActionStrip* interprets the movement.

Action	Symbol ID field
Identify the hand to which the movement is associated	*fill*
Identify floor in which the movement lies	*c-g*
Identify the type of the movement	*c-g-b-v*
Identify the length of the movement	*variation*
Identify the direction of the movement	*rotation*

Table 6.4: The fields of the symbol ID corresponds the facial bones of interest.

Movement is obtained using the key-frame technique. The key-frame technique is used in animations to define the starting points and ending points of a movement [9]. Interpolation is then used to produce in-between frames to obtain a smooth transition. Different types of movement will require varying numbers of key-frames. A straight wall plane movement will require two key-frames, the starting point and the end point. However a circular movement will require more key-frames to produce a circular motion.

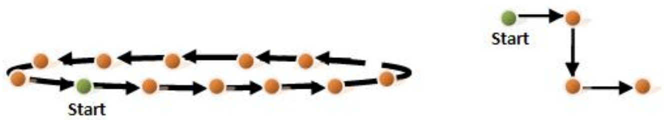

Figure 6.12: Using more key-frames to produce circular and other complex movements. The orange dots indicate the key-frames. The green dots indicate where the movements start.

The timing was defined manually by inserting a key-frame at a certain position in time, where the position defines the timing of the movement. Let the starting point be defined as point x(t_0). The next position of the key-frame at x(t_i), is defined as:

$$x(t_i) = x(t_{i-1}) + s(\Delta t) \tag{6.2}$$

where Δ t = 25, s = speed. There are dynamic symbols that are used to indicate slower or faster movements. Speed was defined as $s = 1$, $s = 2$, $s = 3$ for very fast, neutral and very slow movements respectively. Thus if $x(t_0) = 0$, then the next key-frame for a fast (s = 1) speed will be at $x(t_1) = 25$. For a normal speed $x(t_1) = 50$ and for a slower movement $x(t_1) = 75$. This means it will take 25, 50 and 75 seconds for fast, normal and slow movement respectively. Using FK and IK the hand is able to move in the specified path and direction. The bone's rotational constraints allow the avatar to perform poses that are possible to humans.

6.2.3 Interpolation

In order for a SL animation to be realistic, it requires functional body movements and smooth transitions. Functional body movements, in turn require the joint orientations to be represented efficiently. Interpolation is used to obtain smooth transitions between poses. For example, in Frame 1 the avatar is posed with his right hand on his head. In frame 8 the hand has moved forward. Interpolation generates the in between frames to gradually and sequentially move the arm from the head in Frame 1 to in front of the head in Frame 8. This is shown in Figure 6.13.

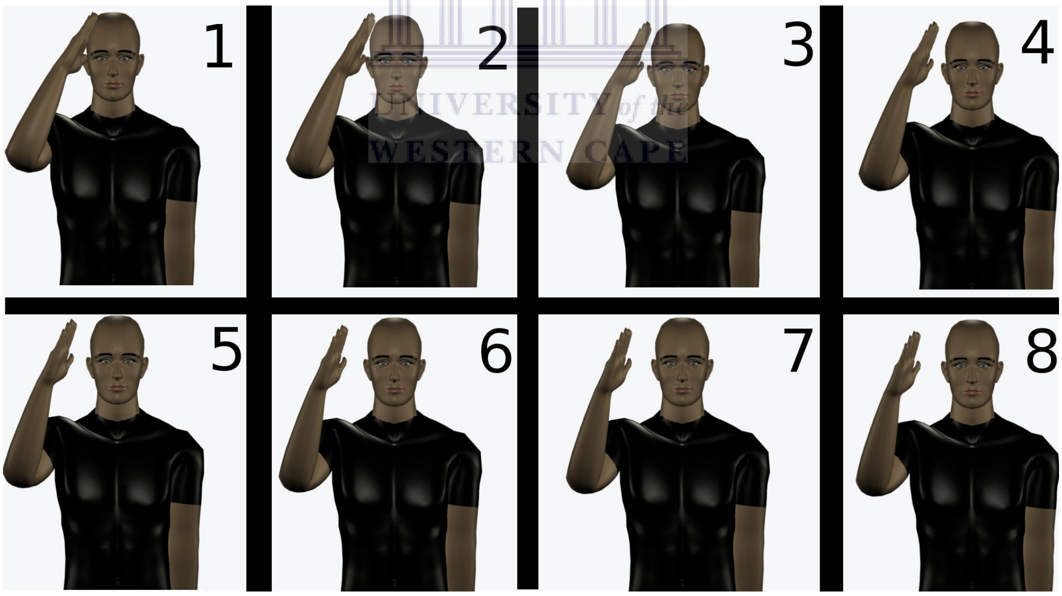

Figure 6.13: Example of Interpolation Between Frames in Blender.

6.2.3.1 Euler angles

Euler angles describe the orientation of an object using three rotations around three axes of rotation. This representation is a simple and intuitive approach of describing object orientation, however it has some problems. Gimbal lock occurs when one axis of rotation lines up with another axis of rotation (Figure 6.14). The order of specification is important. An example of gimbal lock is shown in Figure 6.14. The first rotation is Q_1 around the u axis, the second rotation is 90° around the v axis, and Q_3 is the rotation around the w axis. Q_3 rotation around w axis has the same effect as the rotation around the initial u axis. The v axis rotation causes u and w axes to get aligned. 1 DOF is lost since the last rotation is not unique.

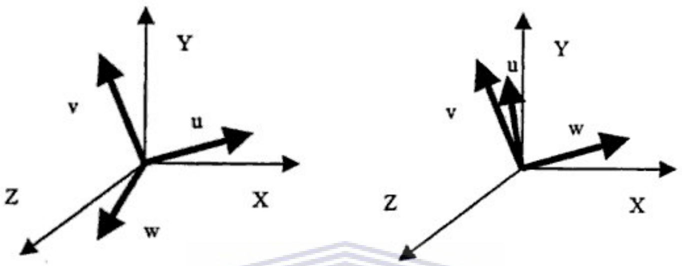

Figure 6.14: An example of gimbal lock using Euler angles.

6.2.3.2 Quaternions

The representation of rotation using quaternions is compact. It is four-dimensional and contains 4 DOFs. The set of all possible rotations fits naturally into quaternions. The rotations are exactly the unit quaternions, which are symbols of the form

$$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \tag{6.3}$$

where a, b, c, and d are real numbers satisfying

$$a^2 + b^2 + c^2 + d^2 = 1 \tag{6.4}$$

Quaternions are not as simple and intuitive as Euler angles. However, they are more advantageous. They do not suffer from the problem of gimbal lock. Quaternions provide a simple and more accurate interpolation. Blender's bone system uses quaternions to perform rotations stored in key-frame poses [56]. Blending or transition between key-frame poses and different actions is based on quaternion interpolation [56]. Given two quaternions q1 and q2 the interpolated quaternion q is given by:

$$\mathbf{q} = \frac{\mathbf{q}_1 \sin(1-t)\theta + \mathbf{q}_2 \sin t\theta}{\sin\theta}$$

t determines the amount of interpolation and varies between 0 and 1 [56].

The effectiveness of both Euler and quaternion interpolation methods were studied. Euler angles don't always interpolate in a natural way. They suffer from gimbal lock [19]. Some orientations end up with one less DOF. Interpolation between two Euler angles therefore does not necessarily take the shortest path. The main advantage of quaternions is that there's no singularity like there is with Euler angles.

6.2.4 Blender's Python Scripting

Blender creates animations using two methods: scripting and manual posing. Scripting facilitates consistency across all animations. It facilitates fluid and natural signing. Manual posing involves the positioning of the avatar directly in the interface by a user. This method allows for precise positioning of the avatar. It provides accurate poses for the hands and the face. The manual posing of individual signs can be a laborious process. When compared to using Python scripts for generating a sign, Python scripts are a more practical process. Manual posing enables the user to produce detailed movements and articulations of the model, particularly the fingers and the facial-features. It allows the user to see exactly what is happening to the avatar as it is being manipulated. For these reasons, scripting and manual posing were combined to create animations. Complex hand-shapes and facial expressions are manually posed and stored in key-frame poses. These poses are merged using the scripting method.

6.3 Summary

The parser implemented is divided into two classes: *SignSpelling Sequence* and *ActionStrip*. *SignSpelling Sequence* is responsible for spell checking and determining the order of reading symbols in a sign. *ActionStrip*, is responsible for generating animation parameters. Interpolation is used to obtain smooth transitions between poses. An efficient means for representing joint orientations is important. Euler angles suffer from gimbal lock. Quaternions provide a simple and more accurate interpolation.

Chapter 7

Experimental setup

This chapter describes how the system was evaluated. Based on the research question in Chapter 1 there are two criteria that a system must adhere to. The animations should accurately match the SignWriting input. Two aspects of the system were tested, namely the recognition of signs generated by the system as viewed by the evaluators and the accuracy with which the signs generated by the system match the SignWriting input. This chapter gives an overview of what the criteria are that were used for judging the recognition and the accuracy of the signs rendered by the system, how the evaluators for the system were selected, how the signs used to evaluate the system were selected and how the evaluation of the criteria was conducted. It also provides a brief overview of the online evaluation tool used to perform the evaluation of the system.

7.1 Testing criteria

Evaluations were conducted to indicate the level of quality of the animations. The work presented in this thesis forms part of an MT system that will be used by Deaf people who will need to understand the sign rendered by the system. The following subsections discuss the recognition criterion followed by the accuracy criterion.

7.1.1 Sign recognition

Recognition is an important criterion for evaluating the adequacy of the SL animations. Sign recognition evaluates the readability, realism and functionality of the SASL animations performed by the avatar. A set of criteria exists that a SL animation system should strive to accomplish in order to achieve the most understandable avatar. These criteria are based on issues relating to SL visualisation discussed in Chapter 3. The criteria defined in [37] are discussed below.

Realism

The avatar should be as realistic and as close to a real human as possible. The avatar should be realistic for the SL gestures to be recognizable and understandable. It is important that the discrepancies between the avatar and humans be very few.

Functionality

The avatar should perform physically plausible poses. The body, arms, head and fingers should move realistically. It should be able to perform plausible facial expressions. Using rotational limits on joints allows the avatar to perform joint rotations applicable to humans.

Fluidity

Consistency should be maintained across all animations. The background, cameras and the avatar should be consistent. The movements should flow smoothly throughout the animations. The animations should also be smooth when merging signs.

7.1.2 Accuracy of the match between the SignWriting input and the animation

This criterion evaluates how accurately the animations match the SignWriting input. The SL animations are generated from the SignWriting input. The animations produced are therefore dependent directly on the input. It is therefore important to evaluate how accurately the animations match the SignWriting input. SignWriting specifies five parameters of SLs. These parameters are hand-shape, location, orientation, movement and non-manual features. In order for a sign to be accurate all these parameters have to be correctly rendered in the animation output. If one or more parameters are incorrect, then the animation is considered inaccurate. Depending on the sign, if one of the parameters is incorrect, the sign could mean something completely different if they are not.

Some signs are very similar and are only distinguished by a slight change in one of the SL parameters. As an analogy, in English the word *meat* may be misspelled as *meet* by changing one letter. The word *meet* exists and has a completely different meaning from *meat*. Although the spelling of these words differ in only one letter, if no context is provided, it would be very difficult to point out that *meat* is misspelled for *meet*. The role played by the letters in a word spelling, is played by the SL parameters in SL gestures. It is very important that these parameters are rendered accurately. Evaluators were asked to examine the quality of the sign relating to these parameters. These parameters have been discussed in Section 2.3.

7.2 Evaluators

SignWriting is not very well known in South Africa. It was therefore decided to use SignWriting experts from other countries. To find evaluators, members of the SignWriting mailing list (which is an international list) were invited via e-mail to participate in the evaluation of the system if they knew both SignWriting and ASL. A link to a website gave them access to a web interface which hosted the evaluation setup. The e-mail provided the project background and described the aim and instructions of the evaluation as well as the evaluation criteria.

It was decided to use these evaluators to evaluate both recognition of signs and to evaluate the accuracy of the match between the SignWriting input and the animation. This was done to be able to correlate the results of the two evaluations.

7.3 Sign Selection

It is the intention to produce a system that can be used with any SL, but the primary focus of this research is on SASL. The evaluators that participated in the testing of the system were from other countries and therefore ASL signs that also occur in SASL were selected for the testing. To select signs that have the same phonemes in ASL and SASL, videos were viewed and signs were selected manually by trial and error. The ASL signs were obtained from the website Signing Savvy, a SL dictionary containing high resolution videos of ASL signs and finger-spelled words [50]. The SASL videos were obtained from (Sign Language Education Development) SLED organisation in Cape Town, South Africa.

After selecting common ASL and SASL signs the corresponding SignWriting for these signs were obtained from the SignWriting repository [51]. Eight common signs were identified. Initially the feasibility of constructing sentences using signs common to both ASL and SASL was considered, but it was decided that it is beyond the scope of this research. The experiments were therefore limited to the evaluation of single signs. These single signs were completely different thus not minimal pairs—minimal pairs differ only in one parameter. It would have been good to test the system with minimal pairs however it was difficult to find minimal pairs signs which are common in both ASL and SASL. It was decided to limit the number of signs used in the test so as to encourage an in-depth analysis of the accuracy of the rendering system by the evaluators without consuming an unreasonable amount of their time.

7.4 Experimental procedure

Due to the dispersion of evaluators, the online evaluation tool was developed to be accessible by means of a web browser. The online evaluation consisted of eight animations. The order in which the animations were presented was the same for all the evaluators. Consistency was maintained throughout the animations, such that the position, background and lighting conditions were the same throughout the animations. Each sign was evaluated for recognition and then accuracy. The following subsections explain the experimental setup for the recognition criterion followed by the accuracy criterion.

7.4.1 Recognition

A video clip of a sign produced by the system is presented to the evaluator. The evaluator identifies the sign without seeing the input or the SignWriting from which the sign was produced. The evaluator types what they interpret the sign to represent rather than choosing the correct answer from a list of words. Thus the answer is completely independent and is not influenced by words listed. The answer provided by the evaluator is dependent only on the animation. This means that the answer is based on what the evaluators sees, rather than choosing the correct word from a few words. It can be assumed that the evaluator uses knowledge rather than being influenced by words listed. The answer is written in English. The answers given by the evaluators might be misspelled. Spelling and grammatical errors by the evaluators are ignored. For example *food* and *eat*, can be confused because even though they are similar signs they have different meanings. Recognition is deemed to have been successful if the interpretation of the sign by the evaluator is the same as the meaning of the rendered sign. If the interpretation of the evaluator differs from the meaning of the sign, recognition is deemed to have been unsuccessful. For example if the evaluator interprets *food* instead of *house*, the recognition is unsuccessful. Recognition is either successful or unsuccessful, thus no point scale is used to evaluate the recognition of the sign. For each animation, the evaluators were asked to type what they thought the sign was and to click on the Submit button. A screenshot of the recognition criterion is shown in Figure 7.1.

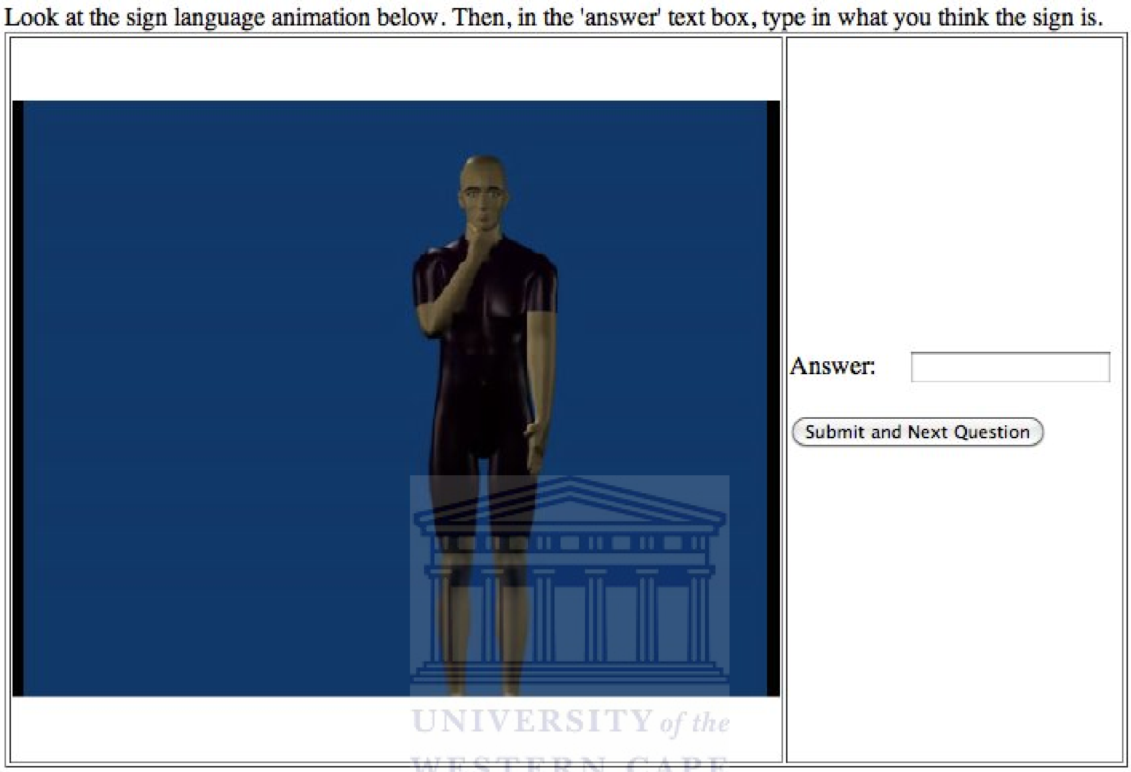

Figure 7.1: Screenshot of the recognition rate evaluation.

7.4.2 Accuracy

The evaluators were then asked to evaluate whether the same sign they had just been shown matched the SignWriting input from which the system produced it. The evaluators are presented with a sign expressed in SignWriting notation. They are also presented with the English interpretation of the sign as well as the SL video clip produced from the SignWriting input by the system. They are then asked to evaluate the accuracy of the sign produced by the system. The evaluators had a *yes* or *no* option to indicate if the sign corresponded to the SignWriting as shown in Figure 7.2. If the SignWriting is inaccurate, the evaluator assesses the video based on the inaccurate SignWriting since the accuracy of the video is tested against SignWriting. The evaluators could add comments in free form in a text box on the same page. Data was collected using MYSQL.

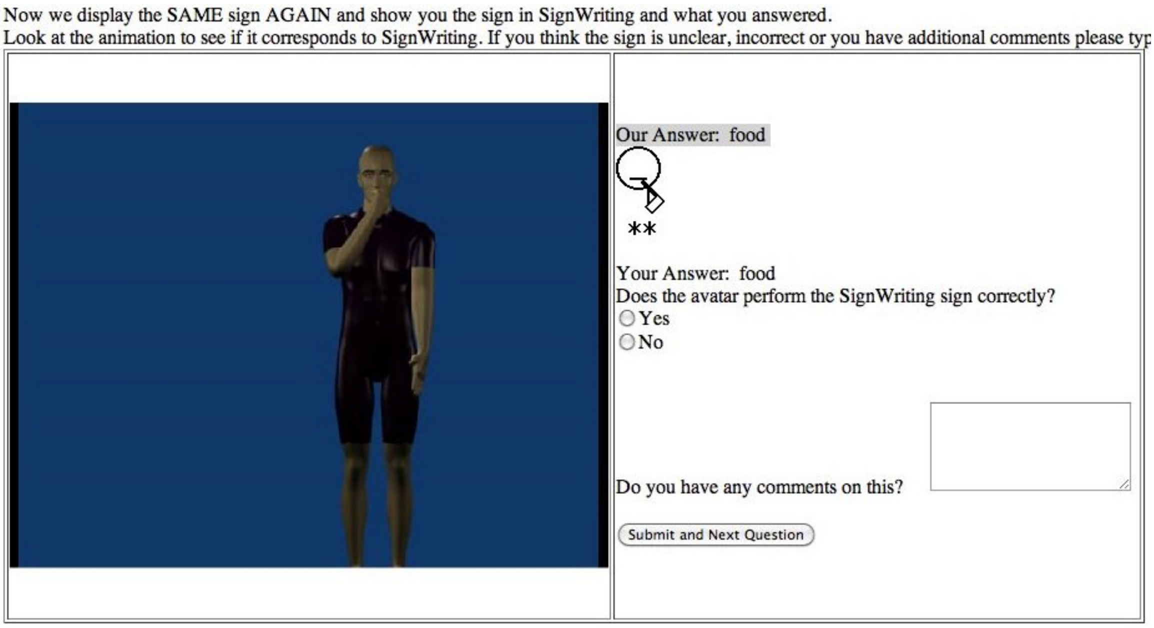

Figure 7.2: Screenshot of the accuracy rate evaluation.

7.5 Summary

This chapter discussed the evaluations that were carried out to evaluate the quality of signing of the system. Two criteria of the system were tested, namely the recognition of signs generated by the system as viewed by the evaluators and the accuracy with which the signs generated by the system match the SignWriting input. The results are discussed in the next chapter.

Chapter 8

Results

This chapter presents the results of the research question based on the signs produced by the system. It evaluates the recognition and accuracy of the SASL sign rendered by the system. It evaluates the intelligibility of the SASL signs rendered by the system from SignWriting input. It investigate how accurately the SASL animation match the SignWriting input. The results of each evaluation is presented separately in the sections to follow.

Each sign was evaluated for recognition and accuracy respectively. The results for recognition and accuracy were separated in order to evaluate the criteria separately. Recognition and accuracy results were collected from the evaluators individual scores. The overall recognition per sign was examined, to study the distribution of recognition per sign. The recognition results differed per evaluator. The recognition per evaluator was also studied in order to examine the distribution of the recognition among the evaluators. The overall accuracy was also studied to examine the distribution of recognition per sign. Since the accuracy was based on the five SL parameters, the distribution of accuracy among these parameters was also studied. Finally, the relation between the recognition and accuracy was evaluated by examining the relationship between them. The results of recognition and accuracy evaluations are presented separately in the sections to follow.

8.1 Recognition results

Table B.1 in the Appendix B summarizes the recognition results. Overall, a recognition rate of 82% was achieved across all signs and evaluators. The results are very promising result, and show that the system can render recognizable SL. The following two subsections discuss the recognition results in more detail. The first subsection studies the recognition results per sign. The second subsection studies these results per evaluator.

8.1.1 Recognition per sign

Figure 8.1 shows the recognition rate per sign. The figure shows that five of the eight signs: *Thank you, Hello, Food, Understand?*; and *Time* obtained 91% recognition. An additional two signs, *now* and *house*, obtained accuracies of 73% and 82% respectively. These signs constitutes seven of the eight signs tested and all achieved very encouraging recognition rates. Only one sign, *same*, scored below 70% with an accuracy of 45%. The next subsection investigates the reason for the recognition rate of this sign.

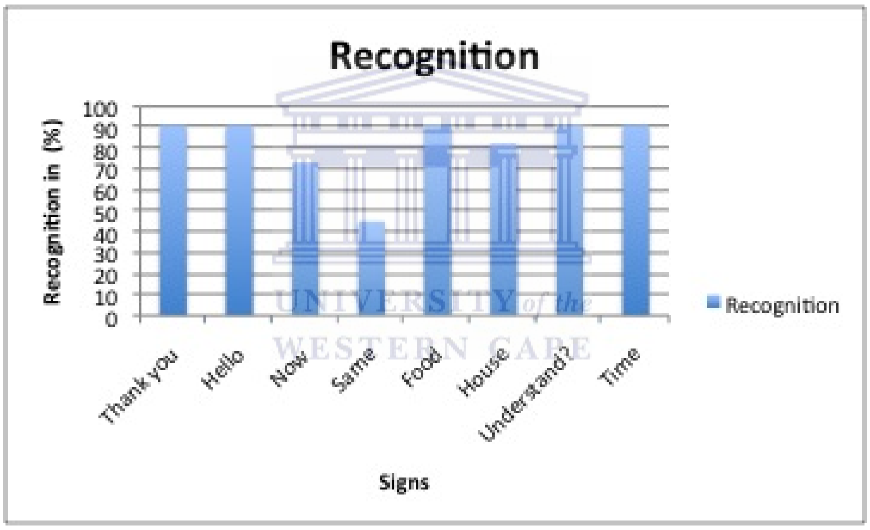

Figure 8.1: Overall recognition rate.

8.1.2 Recognition per evaluator

Figure 8.2 shows the recognition score of each evaluator. Six of the eleven evaluators recognized 88% of the signs. All the remaining evaluators recognized 75% of the signs. These results indicate a very high level of consistency in the responses of the evaluators. This is despite the fact that evaluations were carried out completely independently over the Internet.

Analyzing the comments provided by evaluators revealed that the recognition of signs was a subjective issue. For example, evaluators 4, 6 and 7 commented that the sign *same* was

performed incorrectly in their opinion. Some evaluators commented that the sign *same* should have been performed by making two contacts rather than one. This indicates that these evaluators use a different dialect to the one rendered. This was the reason for the recognition rate of the sign *same* discussed in the previous subsection.
Other comments revealed that some evaluators would have preferred small changes in the animations. Examples of comments were *the left hand seems to be too angled downward* and *the movement is too large.*

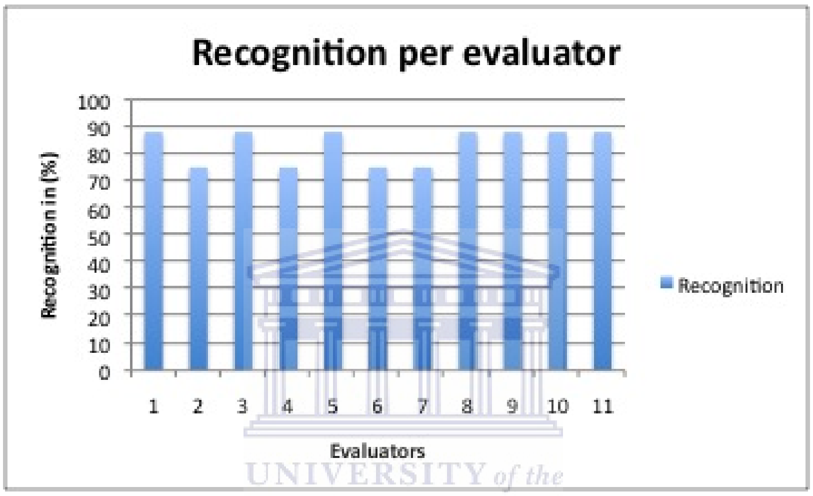

Figure 8.2: Recognition per evaluator.

Despite the issues of subjectivity and dialect, the test results indicate that the signing animations can be understood with a very high recognition rate. This implies that the avatar performs realistic and recognizable SL gestures.

8.2 Accuracy

The accuracy results are divided into two subsections. The first subsection discusses the accuracy results as obtained from the evaluators. The second subsection discusses the accuracy results on a majority vote.

8.2.1 Breakdown of accuracy results

Table B.2 in Appendix B shows the overall accuracy across all signs and evaluators. The 11 evaluators each evaluated 8 signs based on the 5 SL parameters. This implies an overall total of 440 evaluations, a total of 40 parameter evaluations per evaluator and 88 sign evaluations across all evaluators. Of the 440 evaluations, 404 were deemed to be correct. That means that 92% of the parameters were rendered correctly which is very encouraging.

Table 8.1 shows the accuracy score per number of correct parameters as awarded by the evaluators. The rightmost column shows the percentage of the evaluators that felt that all five parameters were correct for each sign. Four signs obtained a high accuracy of 91% and another sign an accuracy of 82%. The remaining signs obtained low accuracy scores. It can therefore be said that 63% of the signs rendered were an exact representation of the input notation.

		Parameters Incorrect					
	0	**1**	**2**	**3**	**4**	**5**	**Signs rendered accurately**
Thank you	10	1	0	0	0	0	91%
Hello	10	1	0	0	0	0	91%
Now	4	4	3	0	0	0	36%
Same	2	9	0	0	0	0	18%
Food	10	1	0	0	0	0	91%
House	9	2	0	0	0	0	82%
Understand?	0	11	0	0	0	0	0%
Time	10	1	0	0	0	0	91%
Total (88)	55	30	3	0	0	0	
Average	63%	34%	3%	0%	0%	0%	63%

Table 8.1: Sign accuracy per number of correct parameters.

As stated previously, the overall percentage of parameters classified correctly was 92%. When this is broken down on a per-sign basis across all evaluators and parameters, 4 signs obtained 98% accuracy, one sign obtained 96% accuracy and the remaining signs obtained scores of 84%, 82% and 80% respectively. These results are shown in Figure 8.3. These results set very a high accuracy bar with the majority of signs attaining accuracies in excess of 95% and no sign attaining an accuracy less than 80%.

8.2.1.1 Accuracy per parameter

Hand-shape and facial expression obtained an accuracy of 100% across all signs and all evaluators. Orientation and Location performed very well with accuracies of 92% and 95% respectively. Movement performed well with an accuracy of 72% accuracy across all signs and all evaluators, which was lower than the other parameters. These results are shown in Figure 8.4. On closer investigation of the results, the lower performance

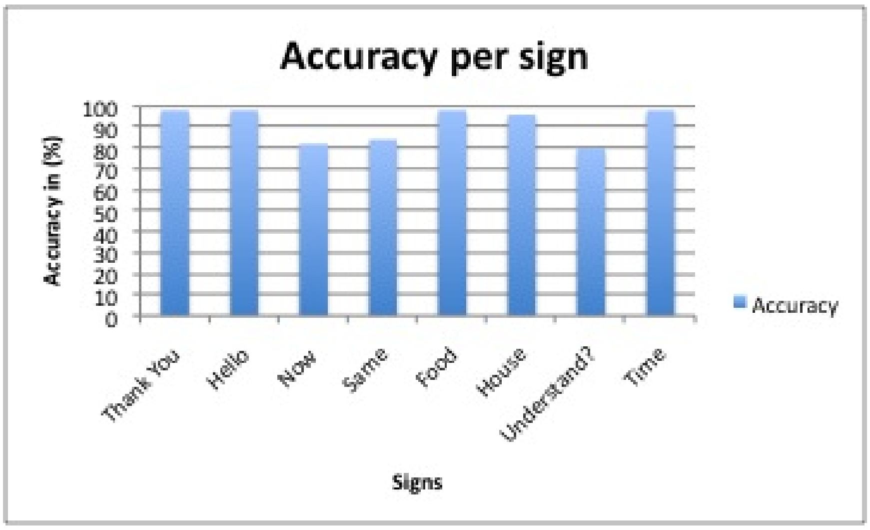

Figure 8.3: Accuracy per sign

of the movement parameter was not a general problem, but was mostly manifested in specific signs such as *understand* and *same*. It was only with these specific signs where it was indicated that the accuracy of the movement parameter could be improved, because over 75% of the evaluators stated that it was incorrectly rendered in these specific signs. Nonetheless, this parameter was generally rated as correct in other signs. The consistency of the results shows that the movement parameter was really a problem for these specific signs. Improving the movement will increase the accuracy of the system even further.

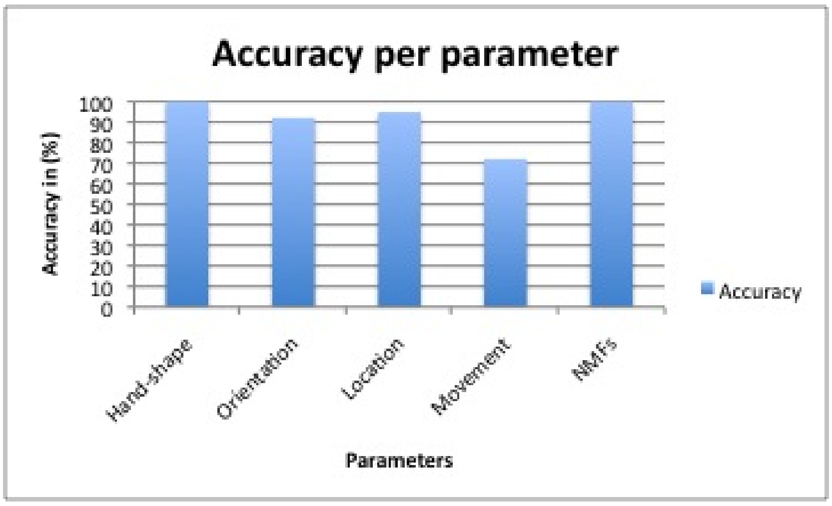

Figure 8.4: Accuracy per parameter.

8.2.1.2 Accuracy per evaluator

An average accuracy score of 92%, with a standard deviation of only 2.26, was obtained across all evaluators. The percentage of parameters classified correctly per evaluator ranged from 88% to 95%. These results are depicted in Figure 8.5 and show how closely the average accuracies of the evaluators are distributed. 3 evaluators are 2% below the average of 92%. 5 evaluators are 1% above the average. 2 evaluators are 3% above the average. Only one evaluator is 4% below the average. This evaluator has found at least one phoneme inaccurate in four signs. Why this is the case is unclear. Had some demographic data been collected with the other data it might have been possible to identify the reasons for this discrepancy. A very high level of consistency in the results across all evaluators exists especially in light of the fact that the evaluations were carried out completely independently and over the Internet.

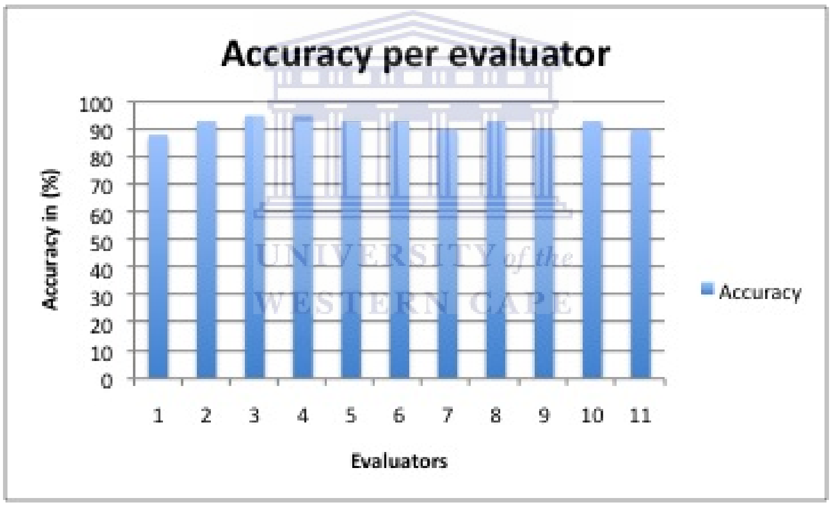

Figure 8.5: Accuracy per evaluator

8.2.2 Accuracy using majority vote

The results stated in the previous section are very encouraging. It was however noted that for some of the evaluations there were a minority of evaluators that did not agree with the majority view. This section looks at how the system performs if a majority vote is considered. Therefore, if a parameter is classified as correct by 80% of the evaluators, it will be considered 100% correct, and the views of the other 20% will be disregarded.

From a total of 440 evaluations as discussed in Section 8.2.1, a minority of 8 evaluations was disregarded. This produced a total of 432 evaluations across all signs, evaluator and parameters, as well as a total of 80 evaluations across all signs and parameters, which can be seen in Table B.3 in Appendix B. The overall accuracy increased by one percent from 92% to 93%, obtaining 402 correct evaluations.

The accuracy of signs rendered accurately with all five parameters classified as correct, increased to 66% obtaining 53 correct evaluations of the 80. This means that 66% of signs obtain 100% accuracy. As indicated in Table 8.2, 5 signs obtain an accuracy score of 100% (increasing from 91% and 82%), one 66%, one 36% and two 0%.

		Parameters Incorrect					
	0	**1**	**2**	**3**	**4**	**5**	**Signs rendered accurately**
Thank you	10		0	0	0	0	100%
Hello	10		0	0	0	0	100%
Now	4	4	3	0	0	0	36%
Same		9	0	0	0	0	0%
Food	10		0	0	0	0	100%
House	9		0	0	0	0	100%
Understand?	0	11	0	0	0	0	0%
Time	10		0	0	0	0	100%
Total (88)	53	24	3	0	0	0	66%
Average	66%	30%	3%	0%	0%	0%	0

Table 8.2: Sign accuracy per number of correct parameters by majority.

8.3 Accuracy vs Recognition

The average recognition discussed in Section 8.1.1 showed recognition of 82%. However the accuracy of the match between the SignWriting input and the animation, indicated that the signing animations were performed accurately with an accuracy of 63% in Section 8.2.1. There appears to be a relationship between the recognition rate of signs and the accuracy with which they were judged to have been performed. The sign *Thank you* had recognition of 91% and accuracy of 91%. In contrast, the sign *understand?* obtained an accuracy of 0% and a recognition of 91%. This raises a question whether a sign that may have been rendered inaccurately, can still be recognizable. The accuracy and the recognition of all signs are compared in Figure 8.6.

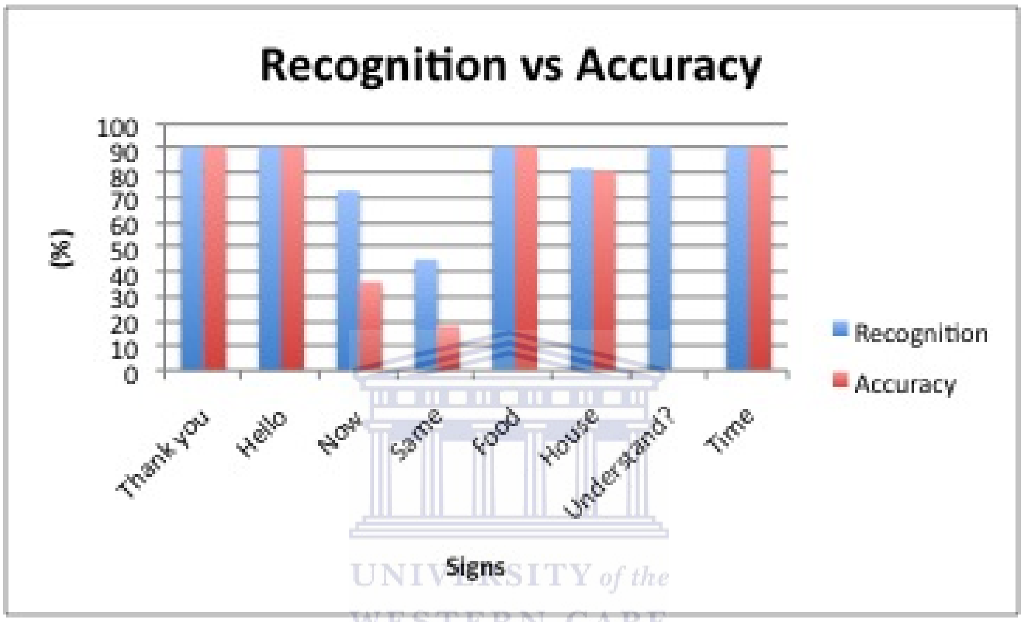

Figure 8.6: Recognition vs Accuracy

Signs *now, same* and *understand?* obtained higher scores in the recognition evaluation by 37%, 27% and 91% respectively. The sign *understand?* obtained an accuracy of 0% and a recognition of 91%. This sign was recognizable because even though one of the SL parameters was not correct, it did not change the meaning, as previously mentioned in 7.1.2 that a change in one parameter could mean something totally different. As an analogy, in English the word *correct* may be misspelled with one letter as *correkt*. The misspelled work *correkt* does not exist in English and does not mean something different. Therefore one can identify what the word was supposed to be. This is the case with the sign *understand?*, even though it did not accurately match the SignWriting input, it was still recognizable. That is, even though the spelling was incorrect, the evaluators were able to recognise it. This indicates that a sign that may have been rendered inaccurately, can still be considered understandable and correct. Five signs on the other hand, show a significantly better understanding with a higher accuracy. Figure 8.6 shows the relationship between recognition and accuracy. Signs *Thank you, Hello,* *Food* and *Time* were more evenly distributed across recognition and accuracy ratings with an average difference of 0% between them. With the exception of three signs, it would seem that there is a relationship between recognition and accuracy. This indicates that if a sign was unrecognized, it did not accurately match the SignWriting input.

8.4 Discussion

In general, the animations gained good acceptance with evaluators. For the most part, the avatar rendered movements that can be recognized and understood. The speed of signing was generally fine. Beginning and ending signs at a neutral position was a bit distracting. It made the animated signs look unnatural. It was one of the main complaints and problems reported by the evaluators. The details of the fingers and the avatars movements also played a role. These errors are discussed in the next subsection.

8.4.1 Finger direction

Insufficient accuracy of interpreting the input contributed to the misinterpretation of the signs rendered. In the test the sign *same* was confusing due to the direction in which the fingers were facing. *Same* requires two parallel index fingers to lie on the floor plane and make contact with each other twice. However two index fingers slightly crossing each other making contact were produced. The direction of the fingertips of the index fingers was confusing. This problem can be solved by determining the direction of the fingers. Fourie classifies discrepancies into two categories [19]. The first category discrepancies make the gesture unrecognisable but will not change the meaning of the gesture. These are discrepancies that are also found between different signers and only make the gesture unrecognisable in severe cases. An example is the sign *Thank you*, in which one evaluator added that the movement is not large enough. The second category discrepancies can change the meaning of the gesture. These are the discrepancies that are most important to avoid. The second discrepancy is found between generated gestures and the actual gesture signed by a real person [19]. An example is the sign *same*, in which evaluators added that the sign looked more like the sign *different*.

8.4.2 Collision detection

The avatar movement is controlled by the joint rotations. The joint restrictions enable the avatar to perform all rotations that are possible by humans. When some joints are

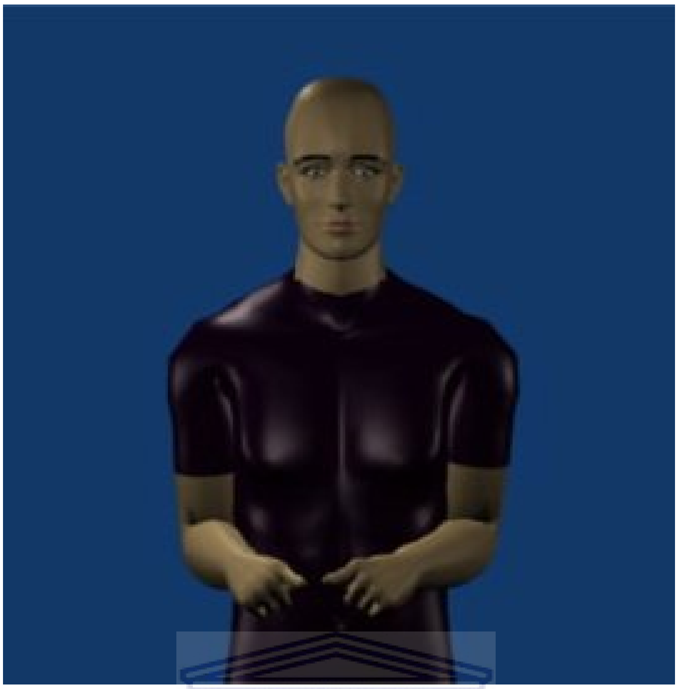

Figure 8.7: The avatar signing the word *same*.

rotated to a certain degree, it becomes easier to describe gestures when the restrictions on that joint are relaxed. The avatar may encounter a rotation that is impossible based on the joint restriction. In such cases a maximum possible rotation is applied. The movement might not be large enough. This may add difficulties in recognizing the sign. The restrictions must be relaxed as much as possible to simplify describing the gestures. In some cases the hand moves through the avatar's body. Using collision detection, this problem can be solved.

8.5 Summary

Recognition results showed that the experiment achieved 82% recognition rate. The accuracy results showed that the experiment achieved 62% accuracy rate. In some case recognition rates were higher than accuracy rates. Correlations between these rates suggests that a sign that may have been rendered inaccurately, can still be recognizable. Based on the results it can be concluded that the animations met the animation criteria. The results have proved that the SL synthesis is understandable. The biggest issue was where the avatar begins and ends signs at a neutral position. The signing speed was not a problem. The results show significantly better understanding of the signed words with a higher accuracy. The results are very encouraging.

Chapter 9

Conclusions

This thesis explored the SL visualisation and the problems relating to SL visualisation namely:

- Representation
- Data
- Consistency
- Linguistic use of the signing space

The thesis showed that virtual signing is the most practical approach to visualise SL. However there are certain criteria that virtual signing must adhere to in order to produce recognizable SL animation. It must be realistic, consistent, functional and fluid in order to produce natural SL animations. Each of these criteria was well addressed in the SL animations output. Fluidity was affected by the avatar beginning and ending signs at a neutral position, evident from the comments added by the evaluators.

The recognition rate showed that in most cases the animations were well recognized with an average of 82% recognition. Accuracy of the match between the SignWriting input and the SL animation on the other hand obtained an average of 62%. This accuracy was calculated on the condition that a SL animation is accurate only if all the five SL parameters are accurate. Thus, it is inaccurate if one or more SL parameters are inaccurate. This condition was applied because inaccuracy in one of the SL parameter can cause a sign to mean something completely different.

The movement parameter achieved an accuracy of 75% across all the signs. One of the criticisms given for this parameter was *I'd like to see a larger arcing movement*, pointing out nothing about the unnatural nature of the movement. These criticisms may have been the results of the subjectivity of the evaluators. In general, the animations were well recognized. For the most part, the avatar rendered movements that can be recognized and understood.

This thesis presented an approach to animating an avatar using SignWriting for SL visualisation. The research question, which was to render recognizable SL animation and to produce accurate SL animations that match the SignWriting input are affirmed by the experiments. In conclusion the thesis has shown the huge potential for SignWriting as an input to SL animation system.

9.1 Future Work

Future work on the animation avatar needs to improve the details of the direction in which the fingers point. A further and important improvement is for collision detection to be added to the system. The system can be tested on SL sentences to evaluate the smooth transition between signs. The most important functionality that needs to be tested on the system is the merging of single words to construct a SL sentence. The results showed that some signs could be recognized even though one of the SL parameters was not correct; the meaning was not changed. Again, the system can be tested on signs that differ only in one parameter.

There are three different types of signs that were defined in Section 5.2.1 and illustrated in Figure 5.3. The system can also be tested on fingerspelling signs as well as compound signs. The fingerspelling test must include all the letters of the alphabet. Another test can be done that evaluates the emphasis of facial expression. It can use signs or sentences that use yes or no questions as well as wh-questions. There are four different avatars that can be used to render the SL gesture developed in [56]. These avatars are of a boy, woman, man and big man [56] and are shown in Figure 8.1.
A test can be done to check whether the approach implemented in this thesis produces the same results throughout all the avatars. That is to test whether the exact same animations are produced from all of the four avatars given the same input. A further analysis can be done to compare how well each of the avatars performs against each other. A test

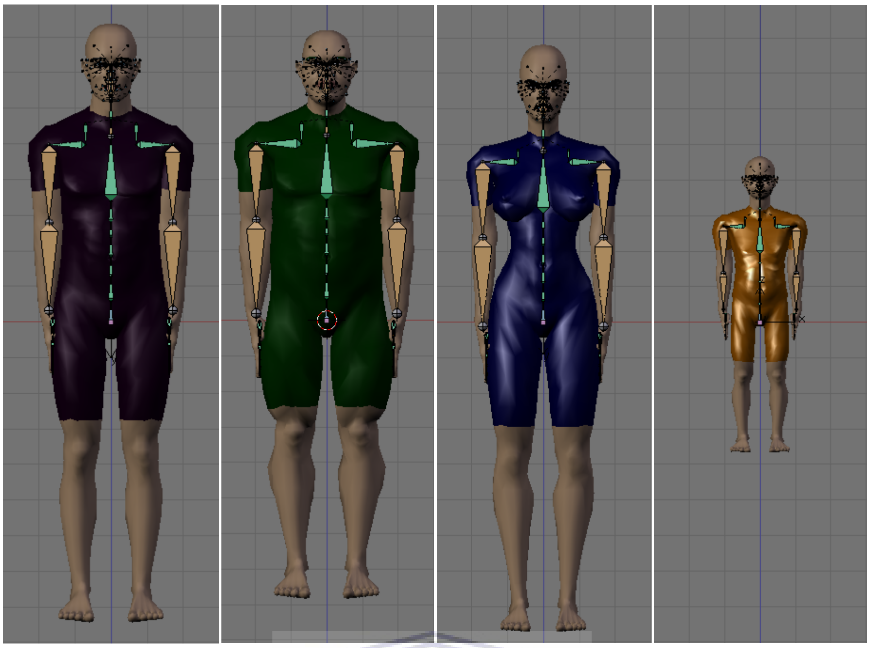

Figure 9.1: There four different avatars developed in [56].

can also be performed to check if the appearance of the avatars has an influence on the recognition of the signs they render.

The system can be extended to be more generic and use other SL notations as input. The performance of each SL notation can be compared. Another test that can be done in the future is to use the system in other SLs as well.

Appendix A

The SWML DTD

<**ELEMENT** swml (sign+) >

<**ATTLIST** swml dialect **CDATA # FIXED** ’S’ >

<!**ATTLIST** swml version **CDATA #FIXED** ’1.1’ >

<!**ATTLIST** swml lang **CDATA #REQUIRED** >

<!**ATTLIST** swml glosslang **CDATA #IMPLIED** >

<!**ELEMENT** sign (gloss | symbol)+ >

<!**ATTLIST** sign lane (-1 | 0 | 1) ’0’ >

<!**ELEMENT** gloss **(#PCDATA)**>

<!**ELEMENT** symbol **(#PCDATA)** >

<!**ATTLIST** symbol x **CDATA #REQUIRED** >

<!**ATTLIST** symbol y **CDATA #REQUIRED** >

Appendix B

Recognition and accuracy results

	Evaluators												
Signs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**Total**	%
Thank You	1	1	1	1	1	1	1	1	1	0	1	10	91%
Hello	1	1	1	1	1	1	1	0	1	1	1	10	91%
Now	0	1	1	0	1	0	1	1	1	1	1	8	73%
Same	1	0	0	0	1	1	0	1	0	1	0	5	45%
Food	1	0	1	1	1	1	1	1	1	1	1	10	91%
House	1	1	1	1	1	1	0	1	1	0	1	9	82%
Understand?	1	1	1	1	0	1	1	1	1	1	1	10	91%
Time	1	1	1	1	1	0	1	1	1	1	1	10	91%
Total	7	6	7	6	7	6	6	7	7	6	7		82%
(%)	88	75	88	75	88	75	75	88	88	75	86		

Table B.1: Recognition score across all signs per evaluators.

		Evaluators											
Signs	**Parameter**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**Total**
Thank You	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location	1	1	1	1	1	1	1	1	1	1	1	11
	Movement	0	1	1	1	1	1	1	1	1	1	1	10
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Hello	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location 1	1	1	1	1	1	1	1	1	1	1	1	11
	Movement	1	0	1	1	1	1	1	1	1	1	1	10
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Now	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	0	0	0	0	0	0	0	4
	Location	1	1	1	1	1	1	0	1	0	1	0	8
	Movement	1	1	1	1	1	1	1	1	1	1	1	11
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Same	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location 1	1	1	1	1	1	1	1	1	1	1	1	11
	Movement	1	1	0	0	0	0	0	0	0	0	0	2
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Food	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location	1	1	1	1	1	1	1	1	1	1	1	11
	Movement	0	1	1	1	1	1	1	1	1	1	1	10
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
House	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location	1	1	1	1	1	1	1	1	1	1	1	11
	Movement	0	0	1	1	1	1	1	1	1	1	1	9
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Understand?	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location	1	1	1	1	1	1	1	1	1	1	1	11
	Movement	0	0	0	0	0	0	0	0	0	0	0	0
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Time	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location	0	1	1	1	1	1	1	1	1	1	1	9
	Movement	1	1	1	1	1	1	1	1	1	1	1	11
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
	Total(40)	35	37	38	38	37	37	36	37	36	37	36	404
	% Marked correct	88%	93%	95%	95%	93%	93%	90%	93%	90%	93%	90%	92%

Table B.2: Accuracy score across all signs, parameters and evaluators.

		Evaluators											
Signs	**Parameter**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**Total**
Thank You	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location	1	1	1	1	1	1	1	1	1	1	1	11
	Movement		1	1	1	1	1	1	1	1	1	1	10
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Hello	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location 1	1	1	1	1	1	1	1	1	1	1	1	11
	Movement	1		1	1	1	1	1	1	1	1	1	10
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Now	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	0	0	0	0	0	0	0	4
	Location	1	1	1	1	1	1	0	1	0	1	0	8
	Movement	1	1	1	1	1	1	1	1	1	1	1	11
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Same	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location 1	1	1	1	1	1	1	1	1	1	1	1	11
	Movement			0	0	0	0	0	0	0	0	0	0
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Food	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location	1	1	1	1	1	1	1	1	1	1	1	11
	Movement		1	1	1	1	1	1	1	1	1	1	10
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
House	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location	1	1	1	1	1	1	1	1	1	1	1	11
	Movement			1	1	1	1	1	1	1	1	1	9
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Understand?	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location	1	1	1	1	1	1	1	1	1	1	1	11
	Movement	0	0	0	0	0	0	0	0	0	0	0	0
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
Time	Hand-shape	1	1	1	1	1	1	1	1	1	1	1	11
	Orientation	1	1	1	1	1	1	1	1	1	1	1	11
	Location		1	1	1	1	1	1	1	1	1	1	9
	Movement	1	1	1	1	1	1	1	1	1	1	1	11
	NMFs	1	1	1	1	1	1	1	1	1	1	1	11
	Total(40)	34	36	38	38	37	37	36	37	36	37	36	402
	% Marked correct	94%	95%	95%	95%	93%	93%	90%	93%	90%	93%	90%	93%

Table B.3: Accuracy score across all signs, parameters and evaluators on majority vote.

Bibliography

[1] D. Aarons and P. Akach. 6 South African Sign Language: one language or many? *Language in South Africa*, page 127, 2002.

[2] Autodesk. Autodesk Maya. [Online] Available at `http://usa.autodesk.com` [Accessed 18 August 2010].

[3] A. Baker and B. Woll. The Modern Language Journal. In *Sign Language acquisition*, volume 94, pages 531–532. John Benjamins, 2010.

[4] D.C. Baynton. *Forbidden Signs: American Culture and the Campaign against Sign Language*. University of Chicago Press, 1996.

[5] J. Berke. Communication—Total Communication. [Online] Available at `http://deafness.about.com/cs/communication/a/totalcomm.htm` [Accessed 12 October 2010].

[6] Blender. Blender Foundation. [Online] Available at `http://www.blender.org` [Accessed 18 August 2010].

[7] J. Bungeroth, D. Stein, P. Dreuw, M. Zahedi, and H. Ney. A German Sign Language Corpus of the Domain Weather Report. In *Fifth International Conference on Language Resources and Evaluation*, pages 2000–2003, 2006.

[8] Caligari Corporation. trueSpace 7. [Online] Available at `http://www.caligari.com/Products/trueSpace/tS75/brochure/intro.asp?Cate=BIntro` [Accessed 18 August 2010].

[9] S. Coquillart and P. Jancene. Animated Free-Form Deformation: An Interactive Animation Technique. In *Proceedings of the 18th Annual Conference on Computer graphics and Interactive Techniques*, pages 23–26. Association for Computing Machinery, 1991.

[10] S.J. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt, and S. Abbott. TESSA, a system to aid communication with Deaf people. In *ASSETS 2002, Fifth International ACM SIGCAPH Conference on Assistive Technologies*, pages 205–212, Edinburgh, Scotland, 2002. Association for Computing Machinery.

[11] O. Crasborn, H. Sloetjes, E. Auer, and P. Wittenburg. Combining video and numeric data in the analysis of sign languages within the ELAN annotation software. In *Proceedings LREC 2006 Workshop on Representation & Processing of Sign Languages*, pages 82–87. Paris : ELRA, 2006.

[12] A.C. da Rocha Costa and G.P. Dimuro. SignWriting and SWML: Paving the way to sign language processing. In *Proceedings of TALN 2003 - Traitement Automatique des Langues Naturelles, Workshop on Natural Language Processing of Minority Languages and Small Languages*, Batz-sur-Mer, 2003.

[13] DEAFSA. Deaf Federation of South Africa. [Online] Available at `http://www.deafsa.co.za/index-2.html` [Accessed 4 June 2010].

[14] K. Denham and A. Lobeck. *Linguistics for Everyone: An Introduction.* Wadsworth Pub Co, 2009.

[15] E. Efthimiou and S.E. Fotinea. GSLC: Creation and Annotation of a Greek Sign Language Corpus for HCI. In *Universal Access in Human Computer Interaction. Coping with Diversity, 4th International Conference*, volume 4554 of *Lecture Notes in Computer Science*, pages 657–666, Beijing, China, 2007. 4554, Springer.

[16] R. Elliott, J.R.W Glauert, and J.R Kennaway. A framework for non-manual gestures in a synthetic signing system. In *2nd Cambridge Workshop on Universal Access and Assistive Technology, CWUAAT*, pages 127–136, Cambridge, UK, 2004.

[17] B.B. Fant, B.G Miller, and L. Fant. *The American Sign Language phrase book.* McGraw-Hill Professional, 2008.

[18] S.D. Fischer and P. Siple. *Theoretical issues in Sign Language research*, volume 2. University of Chicago Press, 1991.

[19] J. Fourie. The design of a generic signing avatar animation system. Master's thesis, University of Stellenbosch, 2006.

[20] N. Frishberg. Arbitrariness and Iconicity: Historical Change in American Sign Language. *Language*, 51(3):696 – 719, 1975.

[21] M. Ghaziasgar. The use of Mobile Phones as Service-Delivery Devices in a Sign Language Machine Translation System. Master's thesis, The University of the Western Cape, 2010.

[22] A. Grieve-Smith. SignSynth: A Sign Language synthesis application using Web3D and Perl. *Gesture and Sign Language in Human-Computer Interaction*, 2298:134–145, April 2001.

[23] G. Gustason. Signing Exact English. *Manual Communication: Implications for Education*, page 108, 1990.

[24] H-Anim. ISO/IEC FCD 19774—Humanoid animation (H-Anim). [Online] Available at `http://h-anim.org/Specifications/H-Anim200x/ISO_IEC_FCD_19774/` [Accessed 1 November 2010].

[25] M. Heap and H. Morgans. Language policy and SASL: interpreters in the public service. *Disability and social change: A South African agenda*, pages 134–47, 2006.

[26] J. A. Holt. Stanford Achievement Test–8th Edition for deaf and hard of hearing students: Reading comprehension subgroup results. In *American Annals of the Deaf*, volume 138, pages 172–175, 1993.

[27] Z. Huang, A. Eliens, and C. Visser. STEP: A Scripting Language for Embodied Agents. [Online] available at `ttp://wasp.cs.vu.nl/step/step.`tml [Accessed 18 August 2010].

[28] M. Huenerfauth. *Generating American Sign Language classifier predicates for English-to-ASL machine translation.* PhD thesis, Computer and Information Science, University of Pennsylvania, 2006.

[29] T. Janzen. *Topics in Signed Language Interpreting: Theory and Practice.* John Benjamins Publishing Co, 2005.

[30] R.E. Johnson, S.K. Liddell, and C.J. Erting. Unlocking the curriculum: Principles for achieving access in Deaf education. In *Gallaudet Research Institute Working Paper 89-3*. Gallaudet University, 1989.

[31] T. Johnston, T.A. Johnston, and A. Schembri. *Australian Sign Language (Auslan): An introduction to sign language linguistics*. Cambridge Univ Press, 2007.

[32] R. Kennaway. Synthetic animation of Deaf signing gestures. *Gesture and Sign Language in Human-Computer Interaction*, pages 149–174, 2002.

[33] E.S. Klima and U. Bellugi. *The signs of language*. Harvard Univ Press, 1979.

[34] S. Liddell. *An Investigation into the Syntactic Structure of American Sign Language*. PhD thesis, University of California, San Diego, 1977.

[35] P. Lu. Modeling animations of American Sign Language verbs through motion-capture of native ASL signers. *ACM SIGACCESS Accessibility and Computing*, 96:41–45, 2010.

[36] R. Morgan. Barriers to justice: Deaf people and the courts. *Issues in Law, Race and Gender*, 8, 2001.

[37] S. Morrissey. *Data-driven machine translation for Sign Languages*. PhD thesis, Dublin City University School of Computing, 2008.

[38] L. J. Muir and I. E. G. Richardson. Perception of Sign Language and Its Application to Visual Communications for Deaf People. *Journal of Deaf studies and Deaf education*, 10(4):390–401, 2005.

[39] N.L. Naidoo. Gesture recognition using feature vectors. Master's thesis, The University of the Western Cape, 2009.

[40] J. Nebel. Keyframe Animation of Articulated Figures Using Autocollision-Free Interpolation. In *Proceedings of 17th Eurographics Conference*, Cambridge, UK, 1999.

[41] C.J. Neidle. *The syntax of American Sign Language: Functional categories and hierarchical structure*. The MIT Press, 2000.

[42] K. Nobutaka. The sign languages of Africa. In *Journal of African Studies [Africa Kenkyu] Japan Association for African Studies*, pages 43–64, 2004.

[43] C. Padden. *Interaction of Morphology and Syntax in American Sign Language*. PhD thesis, University of California, San Diego, 1988.

[44] M. Papadogiorgaki, N. Grammalidis, L. Makris, N. Sarris, and MG Strintzis. VSigns-A virtual sign synthesis web tool. In *Proceedings of Sixth COST 276 Workshop on Information and Knowledge Management for Integrated Media Communication*, volume 276, pages 25–31, Thessaloniki, Greece, 2004.

[45] M. Papadogiorgaki, N. Grammalidis, L. Makris, and M. G. Strintzis. Gesture Synthesis from Sign Language Notation using MPEG-4 Humanoid Animation Parameters and Inverse Kinematics. In *Second international conference on Intelligent Environments 2006*, pages 151–160, Athens, Greece, 2006.

[46] N. Poulou. Parsing XML with SAX and Python. [Online] Available at `http://www.devarticles.com/c/a/XML/Parsing-XML-with-SAX-and-Python/1/` [Accessed 13 August 2010].

[47] SASL Project. Integration of Signed and Verbal Communication: South African Sign Language Recognition, Animation and Translation. [Online] available at `http://www.coe.uwc.ac.za/index.php/SASL.html` [Accessed on 18 August 2010].

[48] F. Rubino, A. Hayhurst, J. Guejlman, W. Madson, and O. Plum. Gestuno: International Sign Language of the Deaf. *Carlisle, England: The British Deaf Association*, 1975.

[49] Signgenius. Disability Statistics: South Africa. [Online] Available at `http://www.signgenius.com/info-statistics.shtml` [Accessed on 1 November 2010].

[50] SignSavvy. Sign Language Dictionary. [Online] Available at `http://www.signingsavvy.com/sign/FAT/6679/1` [Accessed 1 November 2010].

[51] SignWriting. SignPuddle Search by Words. [Online] Available at `http://www.signbank.org/SignPuddle1.5/searchword.php?ui=1\&sgn=4` [Accessed on 1 November 2010].

[52] T. Skutnabb-Kangas. *Linguistic genocide in education, or worldwide diversity and human rights?* Lawrence Erlbaum, 2000.

[53] C. Steinback. CARPI: A database for computer generated fingerspelling. In *Twelfth annual technology and persons with disabilities conference*. Center on disabilities, 2001.

[54] W.C. Stokoe Jr. Sign language structure: An outline of the visual communication systems of the American deaf. *Journal of Deaf Studies and Deaf Education*, 10(1):3, 2005.

[55] D. J. Sturman. A brief history of motion capture for computer character animation. *SIGGRAPH, Character Motion Systems, Course notes*, 1994.

[56] D. E. van Wyk. Virtual Human Modelling and Animation for Real-Time Sign Language Visualisation. Master's thesis, University of the Western Cape, 2008.

[57] B. Watermeyer and L. Swartz. *Disability and social change: a South African agenda.* Human Sciences Research Council Press, 2006.