A METHOD FOR ISOLATED SIGN RECOGNITION WITH KINECT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE IŞIKLIGİL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2014

Approval of the thesis:

A METHOD FOR ISOLATED SIGN RECOGNITION WITH KINECT

submitted by **EMRE IŞIKLIGİL** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Assist. Prof. Dr. Sinan Kalkan _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Fatoş Yarman Vural
Computer Engineering Department, METU _____

Assist. Prof. Dr. Sinan Kalkan
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Alptekin Temizel
Graduate School of Informatics, METU _____

Assist. Prof. Dr. Ahmet Oğuz Akyüz
Computer Engineering Department, METU _____

Assist. Prof. Dr. Selim Temizer
Computer Engineering Department, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: EMRE IŞIKLIGİL

Signature :

ABSTRACT

A METHOD FOR ISOLATED SIGN RECOGNITION WITH KINECT

IŞIKLIGİL, EMRE
M.S., Department of Computer Engineering
Supervisor : Assist. Prof. Dr. Sinan Kalkan

September 2014, 70 pages

Although there are various studies on sign language recognition (SLR), most of them use accessories like coloured gloves and accelerometers for data acquisition or require complex environmental setup to operate. In my thesis, I will use only Microsoft™ Kinect® sensor for acquiring data for SLR. Kinect lets us obtain 3D positions of the body joints in real time without the help of any other device. After an isolated sign is captured, paths of the discriminative body joints are extracted. Then, a vector consisting of the extracted paths, called Sign Graph, is created to describe the isolated sign. To be able to compare two sign graphs, as the distance metric, I propose using the average warping distance of the joint paths that the sign graphs include. Dynamic Time Warping is used for effective calculation of the warping distance. Once a distance measure is defined between Sign Graphs, they are classified using k Nearest Neighbours algorithm. The proposed method performed better than the state of the art and achieved recognition rate of 59.3% in signer-independent experiments and 91.0% in signer-dependent experiments with a dataset consisting of 40 signs obtained from 13 different signers.

Keywords: Sign Language Recognition, Kinect, Pattern Recognition, Sign Graph

ÖZ

KINECT İLE YALITILMIŞ İŞARET ALGILAMA İÇİN BİR YÖNTEM

IŞIKLIGİL, EMRE
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Sinan Kalkan

Eylül 2014 , 70 sayfa

İşaret dili algılama (İDA) üstüne çeşitli araştırma çalışmaları olmasına rağmen, bunların birçoğu renkli eldiven ve ivme ölçer gibi aksesuarlar kullanmakta veya çalışmak için karmaşık ortam kurulumları gerektirmektedir. Benim tezimde, İDA için veri edinmek için sadece Microsoft™ Kinect® duyargasını kullanacağım. Kinect, vücut eklemlerinin 3 boyutlu konumlarını, başka bir cihaza gerek duymadan gerçek zamanlı olarak elde edebilmemizi sağlamaktadır. Bir yalıtılmış işaret yakalandıktan sonra, ayırd edici vücut eklemlerinin izlediği yollar çıkarılmaktadır. Sonra, bir yalıtılmış işareti betimlemek için, çıkarılan yollardan oluşan bir vektör, işaret çizgesi, oluşturulmaktadır. İki işaret çizgesini karşılaştırmak için, bir uzaklık ölçütü olarak, işaret çizgelerinin içerdiği eklem yollarının ortalama eğrilme uzaklığını kullanmayı önermekteyim. Devingen Zaman Eğrilmesi, eğrilme mesafelerinin verimli olarak hesaplanması için kullanılmaktadır. İki işaret çizgesi arasında bir mesafe ölçüsü tanımlandıktan sonra, k En Yakın Komşu yöntemi kullanılarak sınıflandırılmaktadırlar. Önerilen yöntem mevcut yöntemlerden daha iyi sonuçlar vermiştir ve 13 farklı işaretçiden elde edilen 40 farklı işaret içeren bir veri kümesi ile yapılan işaretçi-bağımsız deneylerde 59.3%, işaretçi-bağımlı deneylerde 91.0% algılama oranlarına ulaşmıştır.

Anahtar Kelimeler: İşaret Dili Algılama, Kinect, Örüntü Tanıma, İşaret Çizgesi

To my bad decisions...

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Assist. Prof. Dr. Sinan Kalkan for his invaluable support, guidance and patience during this research. I would have not been able to complete this work without his support and effort.

I would like to thank all committee members for their invaluable comments and suggestions. Many thanks to department chair Prof. Dr. Adnan Yazıcı, faculty members and my colleagues in Computer Engineering Department.

The most special thanks go to my best friend and lover Gizem Baştürk for being in my life and have been supporting and encouraging me during the last six years. Her family also deserves special thanks for their invaluable support.

Great thanks to all my friends who made me feel happy, have fun and learn. Their friendship motivated me during this work.

Finally, the greatest thanks go to my family for their endless support.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

ABBRV	Abbreviation
ANMM	Average Neighbour Margin Maximization
ANN	Artificial Neural Network
ASL	American Sign Language
BP	Back Propagation
BSL	British Sign Language
DTW	Dynamic Time Warping
FMMNN	Fuzzy Min Max Neural Network
GSL	Greek Sign Language
HMI	Human-Machine Interaction
HMM	Hidden Markov Model
HMU	Human Motion Understanding
HNN	Hopfield Neural Network
IBL	Instance-Based Learning
ISL	Irish Sign Language
KSL	Korean Sign Language
MHI	Motion History Images
MLSHI	Multi-Layered Silhouette Motion Images
NN	Neural Network
PaHMM	Parallel Hidden Markov Models
PC	Personal Computer
SDK	Software Development Kit
SL	Sign Language
SLR	Sign Language Recognition
SON	Self Organising Network
SONN	Self Organising Neural Network
SP	Sequence Pattern
SST	Skin Segmentation and Tracking

SVM	Support Vector Machine
TDNN	Time Delay Neural Network
TMDHMM	Tied-Mixture Density Hidden Markov Models
WFD	World Federation of the Deaf

CHAPTER 1

INTRODUCTION

1.1 Motivation

Computational power of personal computers has been increasing gradually since the production of the first personal computers. Availability has also increased as a result of the decrease in personal computer prices. As more people have started using computers, interaction between users and computers has become more important which has caused scientists and technology companies to be much more involved in researches about human-machine interaction (HMI). Understanding human beings by using machines became one of the most actively researched areas. Various projects which intend to improve communication between human beings and machines were carried out by companies dominating the market and research labs of the best universities all over the world.

As computers became a necessity in our daily life in the last decade, technology companies focused on mobility rather than computational power. When Apple™'s first generation iPhone® was released on June 29, 2007, the reason why it attracted customers was not only its technical properties but also its way of interaction with the users. According to Gartner [4], 968 million units of smartphones have been sold to the end users, with an increase of 42.3% from 2012. This number is 53.6% of the number of mobile phones sold in 2013 which means more smart phones are sold than feature phones in 2013. There are lots of things affecting the number of sales, but simplicity and easy use of smart phones play the most important role in making them this much successful. Although smart phones and tablet PCs have smaller screen

size, less hardware and less computational power compared to desktop and laptop PCs, they provide much more practical and effective usage with touch screens and touch gestures. What brought this success to smart phones and tablet PCs was the evolution of HMI.

HMI is a very broad field including lots of different subjects. Speech recognition and sign language recognition (SLR) are two related subjects of HMI. Speech recognition has made so progress in the last decade that it is used in many commercial products. There are exhaustive projects hold by large companies which will take this subject further, even to the level of controlling computers by speech and tool-less interaction. Gesture recognition, basis for SLR, has also attracted many researchers and developers since the launch of Kinect® sensors by Microsoft™. However, SLR is not far away from where it started despite the developments in gesture recognition. In fact, as Cooper et al. [11] indicated, automatic sign language recognition is still in its infancy.

According to World Federation of the Deaf (WFD) [1], there are approximately 70 million deaf people around the world. Even though hearing impairment is a widespread problem all over the world, very few people understand sign language, the primary communication method for deaf people. Therefore, they have difficulties in communicating with hearing people without help of a sing language interpreter and this prevents them from managing their daily jobs by themselves. Hiring sign language interpreters or making employees learn sign language where needed may be a solution to this problem. However, they are expensive and non-practical solutions compared to a technology-based solution which recognise sign language automatically. Although such a technology is far from being commercially available for the time being, each study on this subject helps us to get closer to that point.

1.2 Scope

The present study proposes a generic method to recognise isolated gestures of sign language. It is designed to recognise not only signs of a specific language but signs of all 138 sign languages listed by Lewis et al. [38] as long as the format of the input is appropriate. The system consists of a computer for necessary computations

and a Kinect® sensor for data collection. Input data are collected as isolated signs using skeleton tracking feature of Kinect®. An isolated sign is represented as a vector of joint trajectories. Then, each unknown sign is classified based on the label of the closest signs in the training set. Since instance-based learning is used for the classification, no explicit training step is required. However, training data should be present in the system.

Although the proposed method does not constitute a complete system which can substitute a sign language interpreter, it can be used as the base of such a system. It proposes a method to recognise signs using only location and movement of the hands which may be improved further by adding various data collection methods. In addition, it may be used as a part of a more comprehensive system.

1.3 Outline

This chapter explains the reason why this subject is chosen and why it is important, and scope of the study. In Chapter 2, specifications and usage of Kinect® sensor, a summary of sign language linguistics and background for some methods used in the scope of the proposed method is reviewed. Chapter 3 gives brief information about the previous studies on SLR. Details of the proposed method is described in Chapter 4. In Chapter 5, results of the experiments are given and they are compared to the results of some previous works. In the last chapter, an overview of the proposed method and the possible improvements which could be done in the future are given and the performance of the study is commented.

CHAPTER 2

BACKGROUND

In this chapter, specifications and usage of the Kinect® sensor which is the input device of the system, an overview of sign language linguistics and background for DTW and k-NN which are used in the scope of this study are presented.

2.1 Skeleton Capturing Device: Microsoft™ Kinect®

Kinect® is an input device developed by Microsoft™ which aims to sense motion and voice. It is used as the input device of the proposed system. In this section, development history besides technical specifications and usage of it are given.

2.1.1 History

The first version of Kinect® device was released on 4th of November, 2010 [2] and it was compatible with only Xbox 360, a video game console developed by Microsoft™. Beta (non-commercial) version of Kinect for Windows software development kit was released on June 16, 2011. It allowed Kinect® device for Xbox 360 to be used on Windows 7 to develop desktop applications using C++, C# and Visual Basic .NET. After this release, Kinect® has started to be used by more people day by day. Kinect® device received a Guinness Word Record for the fastest selling consumer electronics device ever with a number of 18 million units sold in 2011 [14]. Upon increasing interest on Kinecting applications for desktop, Microsoft™ released both the first commercial version of Kinect for Windows SDK and Kinect for Win-

dows hardware, a Kinect® sensor optimised to be used on computers, on February 1, 2012 [14].

The next version of Kinect for Windows hardware, v2, has been launched in the summer of 2014 with new and improved features, including increased depth-sensing capabilities, 1080p video, improved skeletal tracking and enhanced infrared technology. Because Kinect for Windows v2 was not available during preparation of this thesis, all of the experiments in the scope of it have been performed using the data gathered using Kinect for Windows.

(a) Kinect for Windows

(b) Kinect for Windows v2

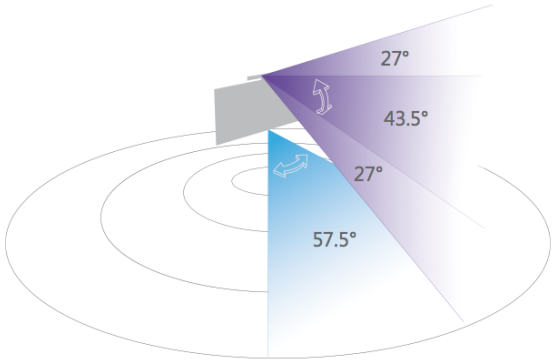
Figure 2.1: Kinect for Windows hardwares

2.1.2 Technologic Overview

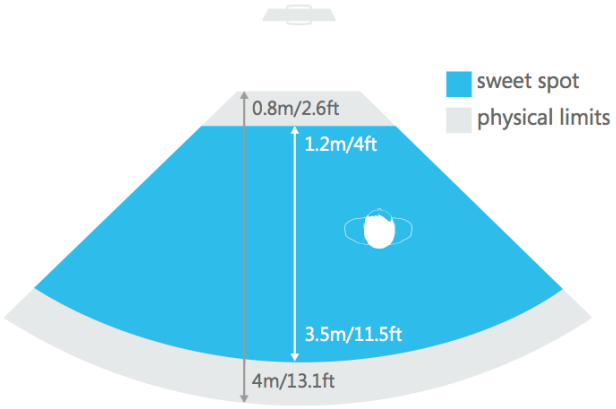
Kinect® device has a color camera, an infrared emitter, and a microphone array consisting of four microphones [45]. Colour camera can save 30 RGB image frames with 640 x 480 resolution or 12 image frames with 1280 x 960 resolution per second [44]. Colour camera can also save images in Raw Bayer, YUV and 16-bit grayscale

channel.

The depth camera can save 30 image frames with 640 x 480, 320 x 240 and 80 x 60 resolutions per second. Because data gathered by this sensor is used for depth-sensing it is also called as depth sensor. Angle of the view of the depth sensor is limited to 57.5 degrees horizontally and 43.5 degrees vertically. The range of the depth the sensor can measure is from 0.8 to 4 meters in the default mode and 0.4 to 3 meters in the near mode. Sweet spot, optimal interaction distance range with the depth sensor, is from 1.2 to 3.5 meters (see Figure 2.2 for visualisations).



(a) Angle of vision



(b) Depth range

Figure 2.2: Depth sensor physical limits (taken from Microsoft^{TM} Corporation [45])

Microphone array is used for directed voice inputs. Voice recognition API is also included in Kinect for Windows SDK to detect specific words as command inputs. The voice sensor can detect inputs from - 50 to + 50 degrees in front of the sensor and can be pointed to points with 10-degree increment within this input range (see

Figure 2.3).

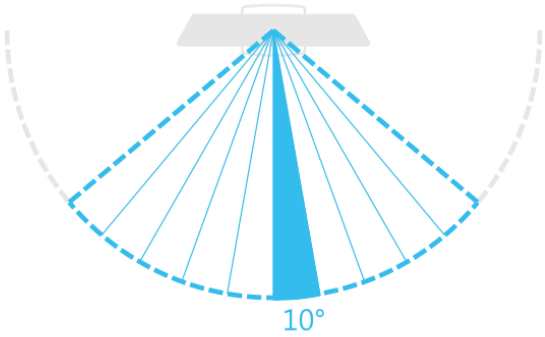


Figure 2.3: Input range of microphone array (taken from Microsoft™ Corporation [45])

Kinect® device is used for research purposes as well as game development to take advantage of features which its SDK provides such as depth-sensing, skeletal tracking and voice recognition. One can obtain raw data from sensors as well as the data processed by SDK. Therefore, it is possible to develop new methods to process sensor data instead of using Kinect for Windows SDK. There are also some other open source drivers and libraries to be used with Kinect® device (e.g. OpenNI®).

In the scope of this thesis, skeleton tracking features of Kinect for Windows SDK is used. The SDK can process the raw data coming from the depth sensor in real time and track skeleton of human beings. It can track whole skeleton of two people at most within its view while position of six people can be tracked at the same time (see figure 2.4).

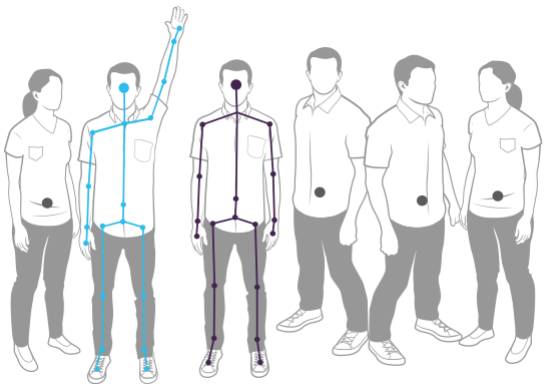


Figure 2.4: Kinect for Windows skeleton tracking limits (taken from Microsoft™ Corporation [45])

Human skeleton can be divided into two parts as upper body and lower body. Kinect for Windows can track twenty joints of human body, half of them belonging to the upper body while the other half belonging to the lower body. Upper body joints consist of right hand, right wrist, right elbow, right shoulder, head, centre of shoulders, left shoulder, left elbow, left wrist and left hand. Lower body joints consist of right foot, right ankle, right knee, right hip, spine, centre of hips, left hip, left knee, left ankle and left foot (see Figure 2.5 for the position of the joints).

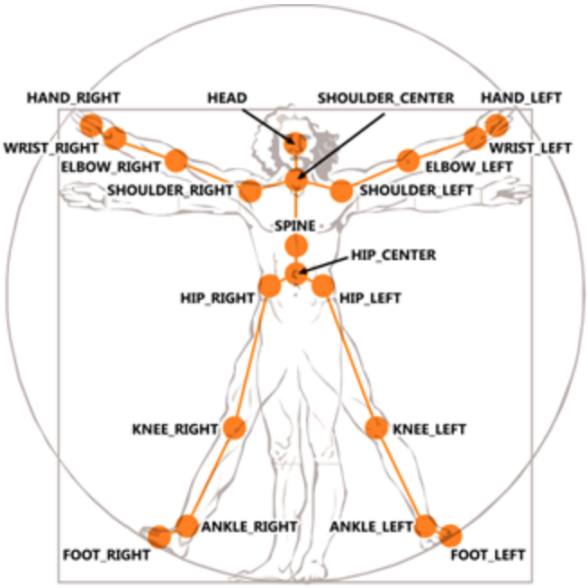
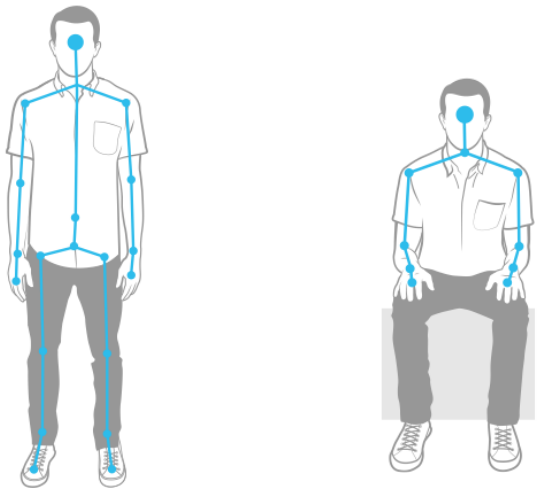


Figure 2.5: Skeleton joints tracked by Kinect for Windows SDK (taken from Microsoft™ Corporation [44])

Kinect for Windows has two modes for skeleton tracking, default and seated mode. In default mode, both upper and lower body joints are tracked while only upper body joints are tracked in seated mode (see Figure 2.6). The SDK outputs 3D position of each joint in real time in skeleton tracking mode. This makes Kinect for Windows an efficient device for gesture recognition and motion capturing purposes.

2.2 Linguistics of Sign Languages

There are various number of rule-based systems, which can be used by following a set of rules, used by people to communicate with each other. Language is one of them as well as Morse code, traffic signs, semaphore, etc. Both spoken language and sign



(a) Default mode (b) Seated mode

Figure 2.6: Kinect for Windows skeleton tracking modes (taken from Microsoft™ Corporation [45])

language have some common features with other rule-based communication systems. According to Clayton [57], some of these features are being composed of symbols, having a systematic structure and having both arbitrary and iconic symbol forms.

Languages and other communication systems are used by combining symbols to produce meaning. Each communication system has different kinds of symbols and ways of composing those symbols. English, for example, has an alphabet consisting of 26 letters each of which is a symbol for a single sound in English words. While Morse code system has codes corresponding to English letters and numerals, American Sign Language has a single sign for an English word.

Sign languages have some sign formation conditions coming from their rule-governed nature. Battison [6] investigated these conditions by observing the structures of American Sign Language (ASL). Based on his results on ASL, Battison proposed that there are two conditions, called the Symmetry Condition and the Dominance Condition which ASL sign formations are based on. According to the Symmetry Condition, if both hands move during a two-handed sign then they are symmetric. In other words, they have the same hand shape and move symmetrically. According to the Dominance Condition, if hands have different hand shapes during a two-handed sign one of them is active and one of them is passive. The active hand is the right

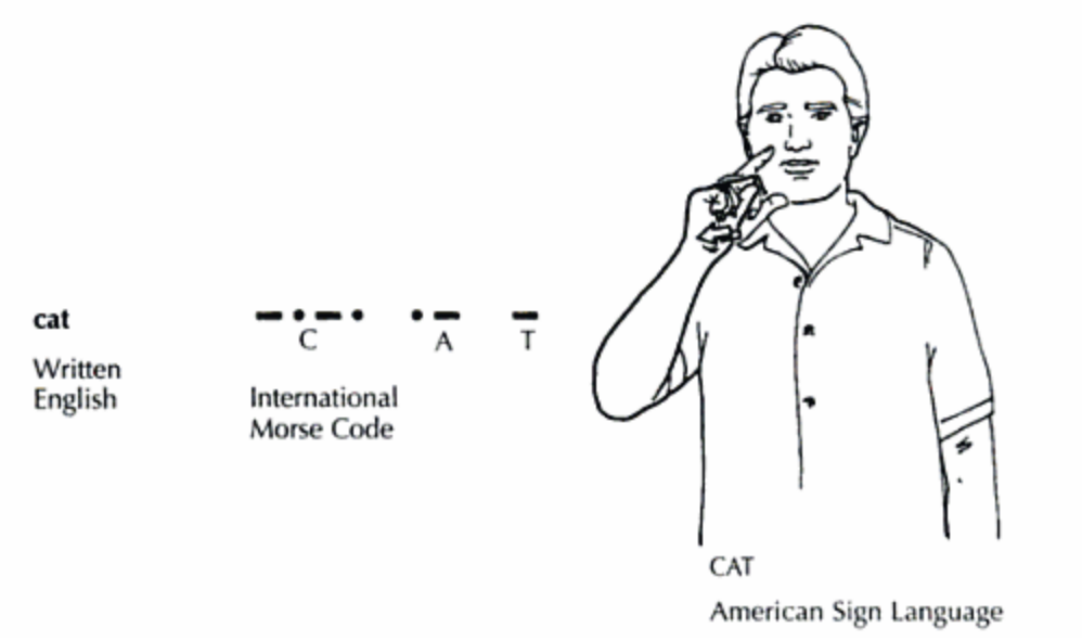


Figure 2.7: Composition of English word 'cat' by different communication systems (taken from Valli [57])

hand if the signer is right-handed and it is the left hand otherwise. The Dominance Condition states that while the active hand moves, the passive hand acts as base in one of seven basic hand shapes (see Figure 2.8). The Symmetry Condition and the Dominance Condition assure that sign languages do not consist of random combination of symbols but have a systematic organisation in the composition and use of the symbols.

In communication systems, a symbol form which does not carry the characteristics of the thing or the activity it defines is called arbitrary. Forms showing some characteristics like voice, shape or visual representation of the thing or the activity they define are called iconic. Having iconic form of symbols, sign languages were claimed not to be "real" languages like ones who have arbitrary forms only. Liddell [39] indicated that having arbitrary and iconic form symbols is not an either-or situation and all languages have both iconic and arbitrary forms. This determination helped to change opinions of linguists who thought sign languages are only drawings in the air.

Liddell [40] indicated some misunderstandings about sign languages. One of them is the thought that communication through sign languages is grammar-less although they have well-defined grammar rules. Another misunderstanding is that people think

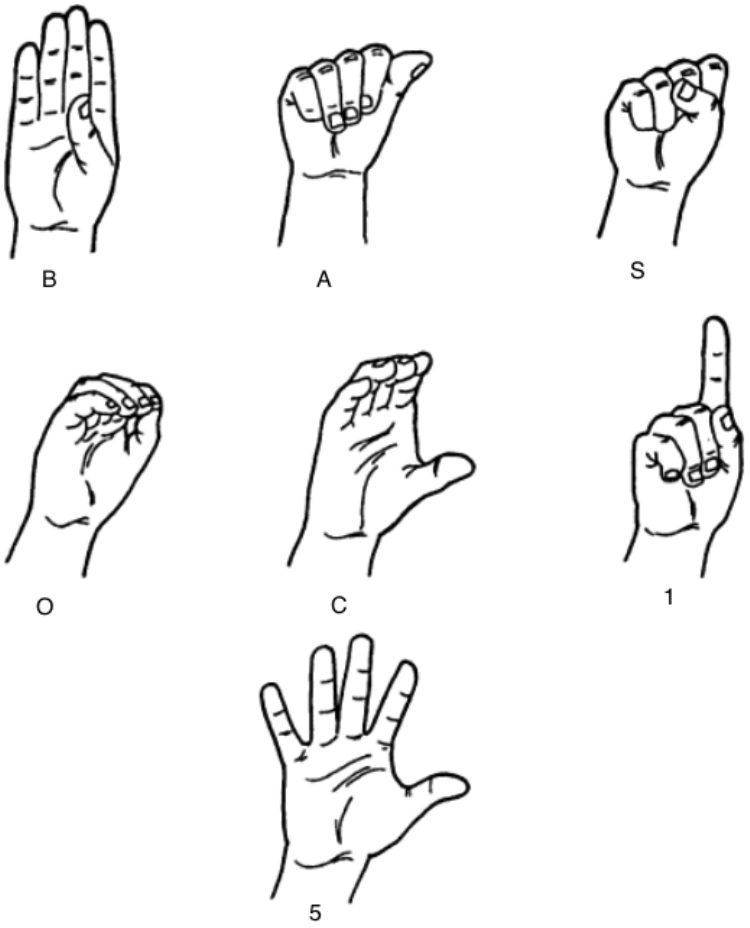


Figure 2.8: Seven basic handshapes of the passive hand (taken from Valli [57])

if they learn a sign language they can communicate to deaf people all over the world. Yet, sign languages have developed independently from each other in deaf communities in different regions of the world. Therefore, both their grammatical structures and meaning of the symbols differ from each other. People also think that sign languages have been adapted from spoken languages. However, sequences of signs in sign language sentences do not mirror sequences of words in spoken languages. These misunderstandings are results of the opinion of people claiming that sign languages are not distinct languages but are symbolisation of spoken languages.

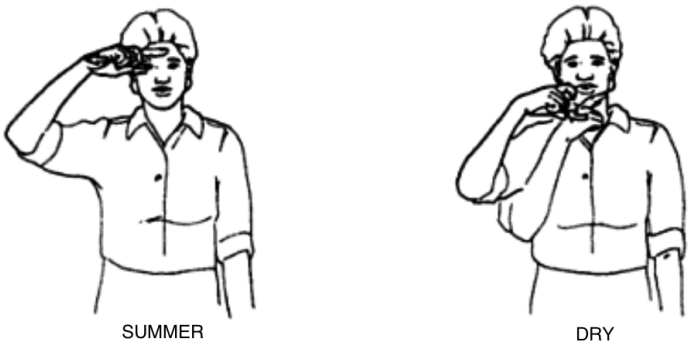
Phonology is a division of linguistics which study organisation of sounds in spoken languages. It is also used by sign language linguists for the branch which study organisation and structure of signs. Phonological studies state that signs can be analysed using five main parts or parameters; hand shape, movement, location, orientation and non-manual signals. Signs may have the same value in some parameters. Each sign

language has hundreds of signs, some of which differ from each other in only one of these parameters. For example, two different signs may differ in only location of the hand or orientation of the palm while other parts are the same. This makes it difficult to distinguish some signs even manually. Thus, it is likely to classify these signs wrong using automatic recognition. Figure 2.9 shows some signs in ASL which differ from each other in only one parameter.

Non-manual signals consist of facial expressions and body posture which either support manual parameters (hand shape, location, orientation, movement) or are required to produce some signs correctly. For some signs, it is not possible to explain the meaning exactly without using non-manual parameters. Clayton [57] gives some examples of signs which require non-manual features to be produced correctly. These examples are NOT_YET which is produced with the mouth open and the tongue is slightly out and FINISH which is produced with the lips protruded.

The first phonological study about sign language structure was carried out by Stokoe [55], before that time signs were thought to have no internal structure, grammar rules and to be not analysable. Stokoe developed a system, called Stokoe System, which divides signs into three parts, contrary to modern phonological systems which divide signs into five parts. In the Stokoe System, parameters of a sign are hand shape, location and movement, which are combined simultaneously to produce a sign. His system deals with palm orientation and non-manual features indirectly. The Stokoe System treats these parts, referred as chremes by Stokoe, as meaningless elements which form meaningful signs when combined. Each part may have a set of values called primes. For example, a set of primes for location parameter includes trunk, face, nose; that for hand shape parameter includes A, B, 5 and movement primes include upward, downward, away from signer.

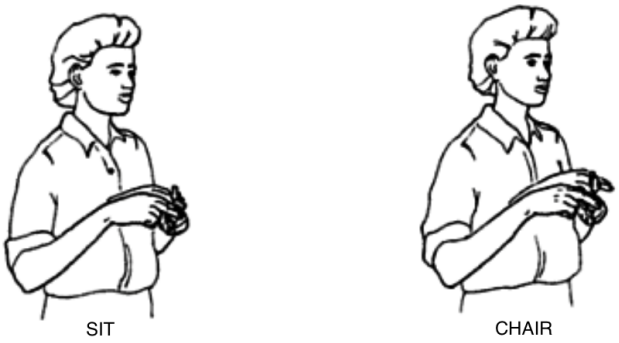
Although the Stokoe System is the first system describing the internal structures of sign language, it has some issues. The first issue is that it does not convey enough detail to describe some signs correctly. Clayton [57] gives some examples of signs which are not described well by Stokoe System because of not having primes specific enough. For example, when describing signs HEAVEN, SIGN and CHILDREN, the Stokoe System use “neutral place where hands move” prime to describe location pa-



(a) Signs differ only in location

(b) Signs differ only in handshape

(c) Signs differ only in palm orientation



(d) Signs differ only in movement

Figure 2.9: Signs differ from each other in only one part (taken from Valli [57])

rameter. However, producing these three signs in the same level is unacceptable (see Figure 2.10). Therefore, location for these signs should be described more specificly. As another example, in the Stokoe System, signs GIVE, NUMBER and NOTHING are described as they have the same hand shape, although NOTHING has totally distinct hand shape compared to GIVE and NUMBER (see Figure 2.11).

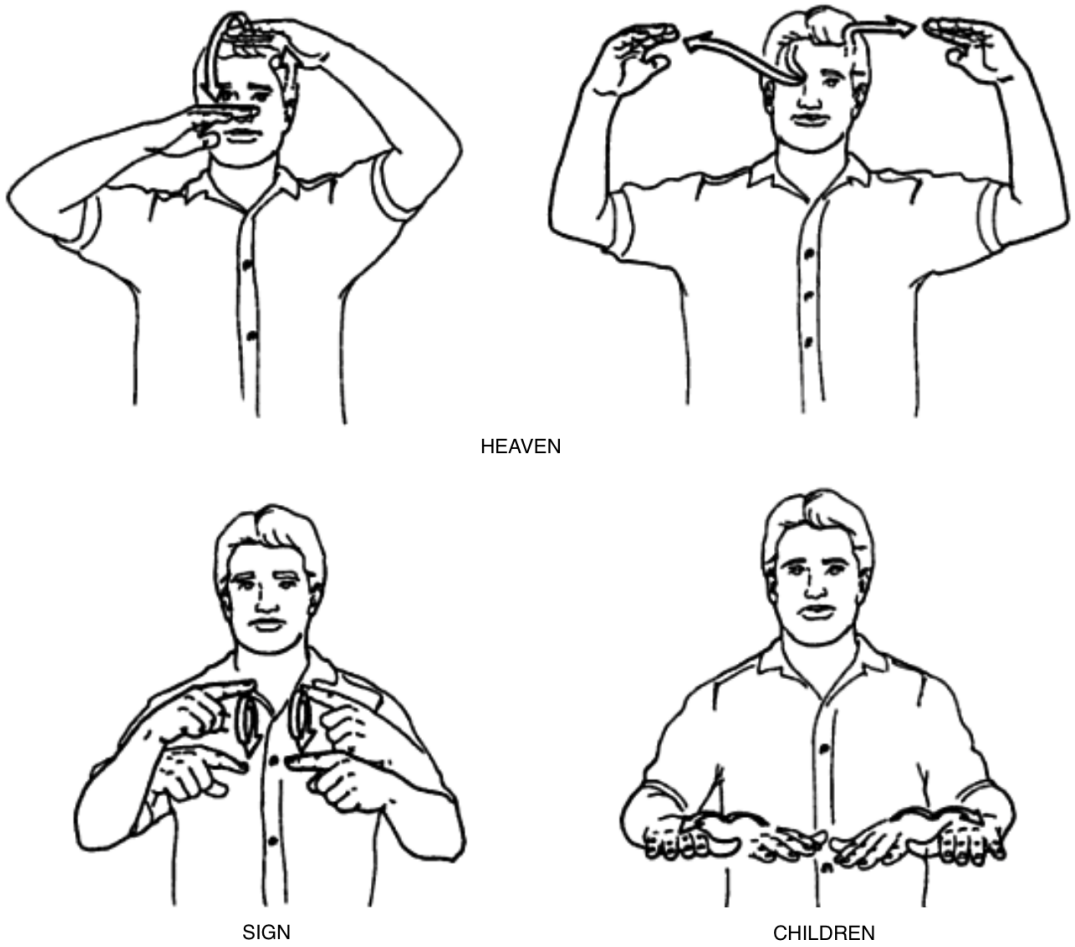


Figure 2.10: Signs that have same location in Stokoe System(taken from Valli [57])

Another issue of the Stokoe System is the representation of sequences in signs. While some signs are formed with only one hand shape, location, movement, palm orientation and non-manual signal, some signs are produced with more than one primes for some parameters. That is, some signs are formed with a sequence of hand shape, location, movement, palm orientation or non-manual signals. The Stokoe System uses movement part to show changes in other parts. In other words, if a sign consists of a sequence of hand shape, location, palm orientation or non-manual signals, these sequences are shown using the movement parameter. It does not totally ignore

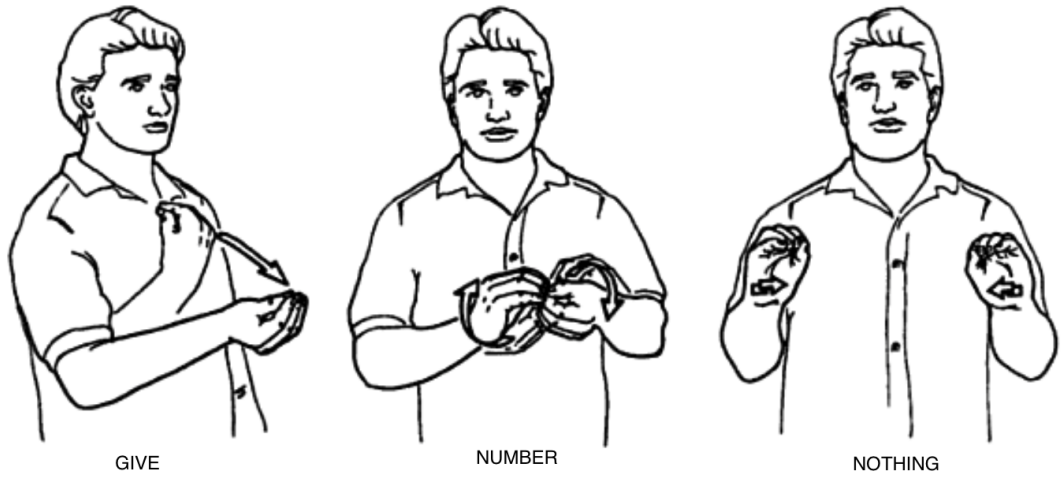


Figure 2.11: Signs that have same handshape in Stokoe System(taken from Valli [57])

sequences of parameters other than movement, it nevertheless sees these sequences unimportant in describing signs. Accordingly, only simultaneous contrast is defined in the Stokoe System. Although there are examples of sequential contrast in sign languages, the Stokoe System does not deal with it.

Liddell and Johnson [41] developed a phonological system, called Movement-Hold System, which overcome the issues which Stokoe's simultaneous system has, especially the sequential contrast issue. Their system introduced the concept of segments as the central element in the structure of signs. The system represents the segments individually and signs as strings of segments. Segments are composed of two feature bundles, named articulatory bundle and segmental feature bundle, describing posture of the hand and activity, respectively. Articulatory features represent the location and the orientation of the hand, hand shape and non-manual signals while segmental features account whether the hand is moving or not and the description of the movement if so. The Movement-Hold System takes its name from the type of segments. Time periods during which articulatory features of segments do not change are called holds. Movements are defined as the time periods during which articulatory features change. Both holds and movements contain one bundle of segmental features while it is not the case for articulatory bundles. Holds have only one bundle to represent articulatory features while movements have two, one specifying the hand posture at the inception of the movement and one specifying it at the conclusion of the movement.

Hence, sequences of articulatory features are involved in the description of signs in the Movement-Hold System.

The system developed by Liddell and Johnson solves descriptive problems of Stokoe's system by giving adequate details on description of signs and accounting sequential contrast between signs by handling sequences in articulatory features. Even though the Stokoe System and the Movement-Hold System have significant differences on description of signs, both systems show that sign languages are not random symbols drawn in the air or visualisation of spoken languages, but have systematic internal structures and systems which describe those structures.

2.3 Overview of Dynamic Time Warping

DTW is a well-known technique used to optimally align two time series data which may vary in time or speed. It is generally used in speech recognition applications to cope with inter-speaker differences in speaking speed. Think about two samples of time-dependent sequences which have local changes independent from each other in time and speed (see Figure 2.12). DTW finds an optimal path which align these sequences without requirement that they have the same length. Whether or not two sequences differ non-linearly in time, DTW measures similarity or distance between them independent of these variations by warping them non-linearly in time. However, the distance metric of DTW does not necessarily hold triangle equality.

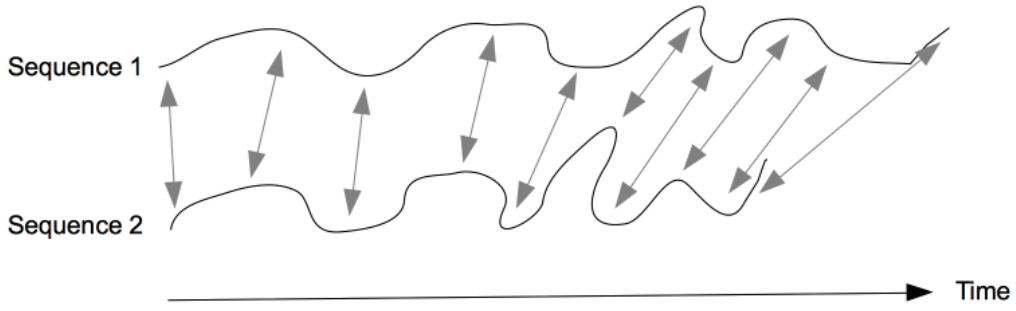


Figure 2.12: Alignment of two time-dependent sequences

DTW compares a sequence $X = (x_1, x_2, ..., x_N)$ of length N with another sequence $Y = (y_1, y_2, ..., y_M)$ of length M, aligns their members and returns an alignment path in addition to a similarity/distance measure. Members of these sequences are, in general, sets of features. Muller [46] names these sets as *feature space* and denotes it by $\mathcal{F}$. Then, *local cost measure*, c, is the function designed for comparing two feature spaces $x, y \in \mathcal{F}$.

$$c : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}_{\geq 0}. \tag{2.1}$$

Measuring local cost of each pair of sequences X and Y, *Local cost matrix* is obtained. It is defined by $C \in \mathbb{R}^{N \times M}$ and $C(n, m) = c(x_n, y_m)$. An optimum alignment path of sequences X and Y goes along the way where the values of local cost are low on the local cost matrix (see Figure 2.13 for illustration).

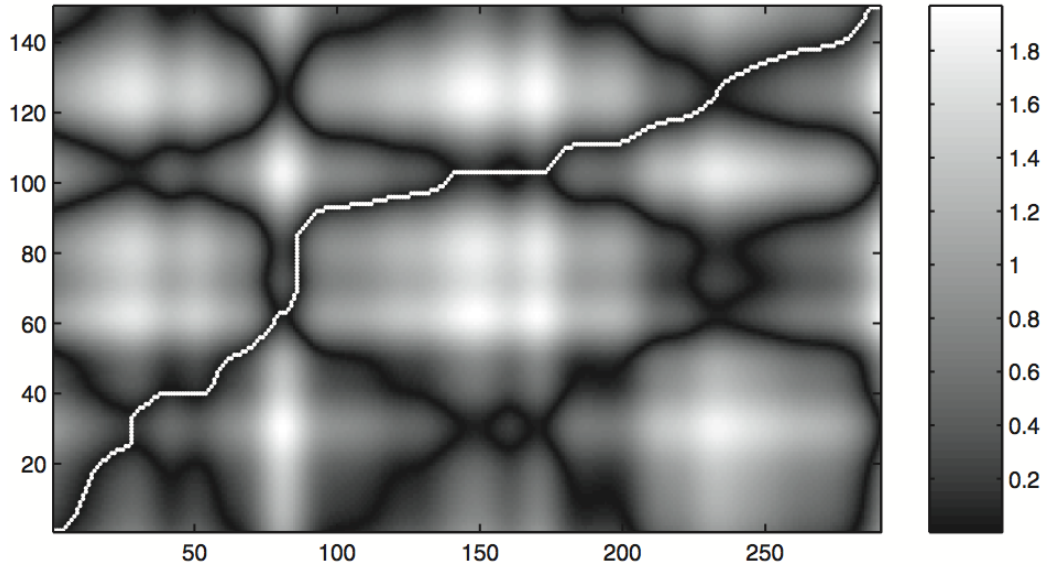


Figure 2.13: Illustration of optimum alignment path on the local cost matrix (taken from Müller [46])

Muller [46] gives the definition of the warping path obtained as a result of DTW technique as

Definition 2.1. *Warping path of two sequences of length N and M is a sequence $p = (p_1, p_2, ..., p_L)$ where $p_l = (n_l, m_l) \in [1 : N] \times [1 : M]$ for $l \in [1 : L]$. The warping path satisfies the following conditions:*

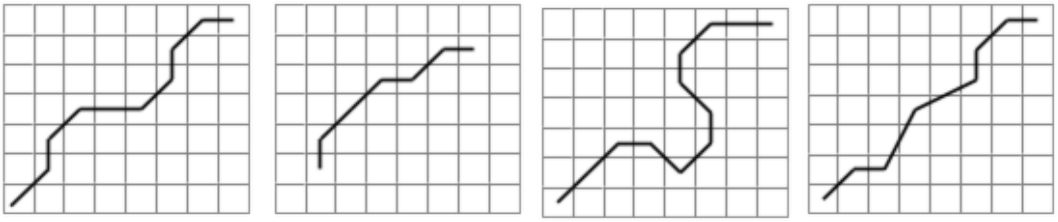
1. $p_1 = (1, 1)$ *and* $p_L = (N, M)$.

2. $n_1 \leq n_2 \leq ... \leq n_L$ *and* $m_1 \leq m_2 \leq ... \leq m_L$.

3. $p_{l+1} - p_l \in (0,1), (1,0), (1,1)$ *for* $l \in [1 : L-1]$.

An alignment of two sequences $X = (x_1, ..., x_N)$ and $Y = (y_1, ..., y_M)$ is defined by a warping path by matching the element x_{n_l} of X with the element y_{m_l} of Y. A warping path which accomplishes the optimal matching should satisfy the three conditions defined above. Figure 2.14 shows some examples of warping paths. The cost of a warping path p of two sequences X and Y can be calculated as:

$$c_p(X, Y) = \sum_{l=1}^{L} c(x_{n_l}, y_{m_l}). \tag{2.2}$$



(a) (b) (c) (d)

Figure 2.14: Illustration of warping path indices of two sequences of length 8 and 7. **(a)** Warping path satisfying all three conditions. **(b)** Warping path violating the first condition **(c)** Warping path violating the second condition. **(d)** Warping path violating the third condition.

DTW guarantees to find an optimal warping path which has the lowest total cost compared to all possible warping paths even if there exists more than one optimal path. The optimal path is denoted by p^*. Then, the DTW distance of two sequences X and Y can be defined as:

$$\begin{aligned} DTW(X, Y) &= c_{p^*}(X, Y) \\ &= \min \{c_p(X, Y) \mid p \text{ is a warping path of } X \text{ and } Y\}. \end{aligned} \tag{2.3}$$

In order to get rid of computational complexity of calculating all possible warping paths to find the optimal one, an algorithm based on dynamic programming having $O(NM)$ computational complexity, where length of sequences are N and M, can be used.

Definition 2.2. *Let X and Y be two sequences of length N and M, respectively, and $X(1:n) = (x_1, ..., x_n)$ for $n \in [1:N]$ and $Y(1:m) = (y_1, ..., y_m)$ for $m \in [1:M]$ be sequence prefixes. Then,*

$$D(n,m) = DTW(X(1:n), Y(1:m)), \tag{2.4}$$

*where D is a $N \times M$ matrix and referred to as **accumulated cost matrix**. It can be easily obtained from Equation 2.4 that $D(N,M) = DTW(X,Y)$.*

Definition 2.3. *Accumulated cost matrix can be calculated using the following recursive rule:*

$$D(n,m) = \begin{cases} c(x_1, y_1) & \text{if } n = 1, m = 1 \\ \sum_{k=1}^{n} c(x_k, y_1) & \text{if } 1 < n \leq N, y = 1 \\ \sum_{k=1}^{m} c(x_1, y_k) & \text{if } x = 1, 1 < m \leq M \\ \text{min}\{D(n-1, m-1), D(n-1, m), & \\ \qquad D(n, m-1)\} + c(x_n, y_m) & \text{if } 1 < n \leq N, 1 < m \leq M. \end{cases} \tag{2.5}$$

Computation of $DTW(X,Y) = D(N,M)$ using the Equation 2.5 has computational complexity of $O(NM)$.

The cost of warping two sequences in time, i.e. their DTW distance, is given by accumulated cost matrix which is also used to obtain optimal warping path $p*$. Algorithm 2.1 extracts optimal warping path. Note that the algorithm uses only accumulated cost matrix as input. The optimum alignment path goes along the way on the accumulated cost matrix where values of the elements are low, as expected. Figure 2.15 represents the visualization of the accumulated cost matrix obtained from the sequences in Figure 2.13 and shows the optimum alignment path on the matrix.

2.4 Overview of k-Nearest Neighbour

The k-NN is a simple non-parametric classification method used in pattern recognition and machine learning. It classifies examples based on the class memberships of the k closest training examples. It is a type of instance-based classifier since the

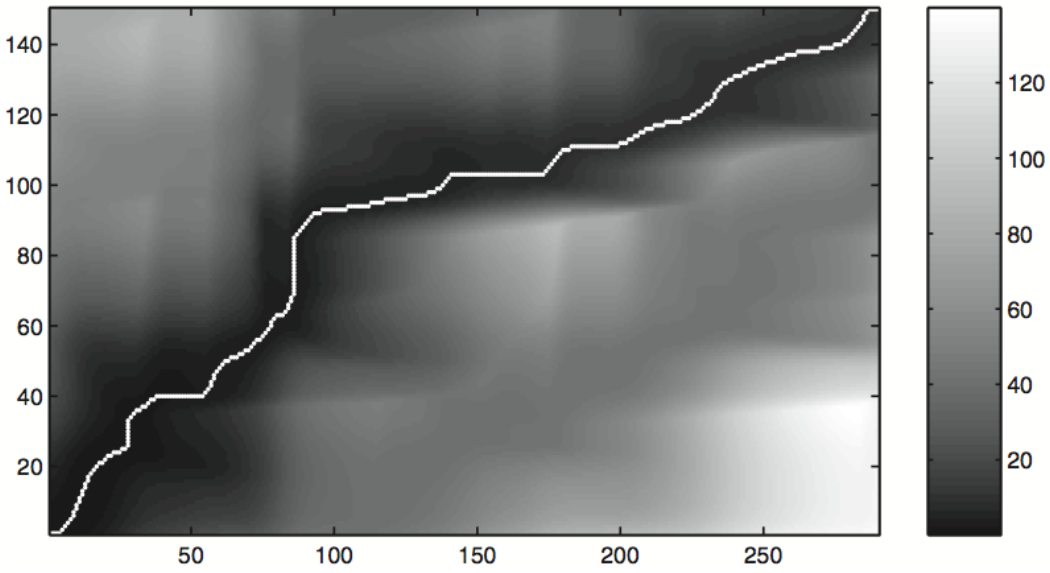


Figure 2.15: Illustration of optimum alignment path on the accumulated cost matrix obtained from sequences in Figure 2.13 (taken from Müller [46])

classification is directly based on training examples and lazy learner since all computation is delayed to the runtime, i.e. no explicit training phase is required. Although it is one of the simplest machine learning algorithms, it does quite well in practice. Moreover, it is easy to implement, thus commonly used in classification problems. All types of data consisting of examples between which a measure of dissimilarity can be determined can be classified using k-NN.

k-NN has two stages; in the first stage, k nearest neighbours are found from the training set based on a distance metric. In the successive stage, the class of the test sample is determined using the class labels of the selected neighbours in the first stage. Let q be the query sample, $t_1, ..., t_k$ be k closest training samples, b_i be the class label of t_i where $i \in [1:k]$ and $H = (h_1, ..., h_{|H|})$ be set of labels. Then, vote of a class label is calculated as

$$Vote(h_j) = \sum_{i=1}^{k} \left(\frac{1}{dist(q, t_i)} \right)^s e(h_j, b_i), \tag{2.6}$$

where $dist$ returns the distance between two samples and e returns 1 if the parameters are equal, 0 otherwise. One can choose a distance function which always returns 1

to ignore distance between the test sample and the closest neighbours while counting the vote of a class label. It is also possible to increase or decrease the influence of the distance by decreasing and increasing s, respectively. The distance between two samples can be computed by

$$dist(t_1, t_2) = \sum_{f \in F} w^f \delta(x^f, y^f), \tag{2.7}$$

where F is the set of features, w^f is the weight of the feature f and δ returns the distance between two features. It is possible to assign the same weight or different

Algorithm 2.1: Optimal Warping Path

Input: Accumulated cost matrix, D, of size $N \times M$

Output: Optimal warping path, $p*$

$n \leftarrow N, m \leftarrow M$ `// Opt. path computed in reverse order`

$i \leftarrow 1$ `// Array index assumed to be starting at 1`

while $n > 1$ *or* $m > 1$ **do**

  $p[i] \leftarrow (n, m)$

  if $n = 1$ **then**

    $m \leftarrow m - 1$

  else if $m = 1$ **then**

    $n \leftarrow n - 1$

  else

    `// Row and column number of the entry with the minimum value`

    $(n, m) \leftarrow IndicesOfArgMin(D[n-1][m-1], D[n][m-1], D[n-1][m])$

  $i \leftarrow i + 1$

end

$p[i] \leftarrow (n, m)$ `// where n and m equals to 1`

$ReverseOrder(p)$ `// Reverse the order of elements`

`//` $p*$ `is the optimal warping path of length` $l = i$

weights to the features. In pattern recognition, extracted features, generally, have different importance in classification. Therefore, assigning different weights to the features affects the performance of the classification.

The value of k is an important factor in determining the class label of the test sample. For example, the class label of the test sample in Figure 2.16 is determined as blue when $k = 4$ while it is determined as red when $k = 9$. The choice of k depends on the size and the structure of the data. Outliers in the data may cause misclassification in small values of k. For this reason, choosing larger values for k may be preferred when the data contain outliers. Nevertheless large values of k cause distinction between classes disappear. Therefore, the optimal value of k should be found to get better results in classification. There are various methods proposed to find the optimal value of k. In the scope of this thesis, the optimal value is decided based on the experiments.

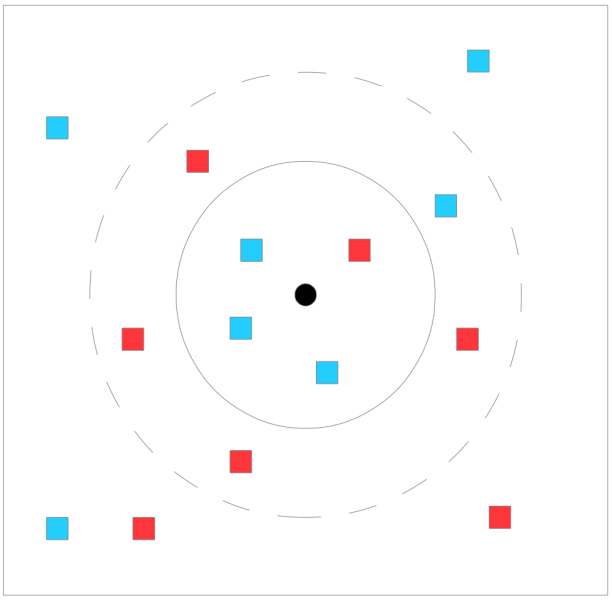


Figure 2.16: Nearest neighbours with different values of k

Being a lazy learner, the algorithm makes use of all training examples during the run-time which may cause excessive memory usage for large training sets. Moreover, as training set gets larger computational cost increases due to the fact that all computations are deferred to the run-time. Therefore, k-NN may not be tractable for very large training sets. On the other hand, it does not require explicit training and removes the need of repeating training step after each time a new training sample is added to the database. In other words, training set can be extended easily if k-NN is employed as

the classification method.

CHAPTER 3

OVERVIEW OF AUTOMATIC SLR

Automatic SLR studies have started in the middle of 90s, remarkably long time after the development of systems describing the systematic structure of signs. It has not attracted too much attention since then. Though, there are some exhaustive studies on it. In this chapter, literature on SLR is presented using a similar categorisation to the one introduced in the study of Cooper et al. [11]. The literature on SLR has been examined according to the methods used in data acquisition and feature extraction and the methods used in classification. A summary of complete SLR systems is given in Table 3.1.

3.1 Data Acquisition and Feature Extraction

Data acquisition is the first and a crucial step in automatic SLR. Because the meanings of the signs are derived mostly from the manual features, i.e. location and movements of the hands, hand shape and palm orientation, most of the SLR systems have focused on these features. The non-manual features such as facial expressions and the shape of the mouth are not in the scope of this study. Data acquisition methods are investigated in 3 categories, tracking methods, non-tracking methods and hand shape recognition.

3.1.1 Tracking Methods

Development and use of hand tracking methods are common in automatic SLR. Hand tracking for SLR is a difficult task since the sign gestures are not designed for au-

tomatic recognition purposes and they, generally, consist of quick movements of the hands and occluded body parts. Therefore, various methods are proposed for this non-trivial task in the scope of SLR.

Using wearable sensors and accessories is one of the most preferred methods for data acquisition. Some of the earlier systems made use of data gloves and accelerometers. For example, Waldron and Kim [61] developed a system which obtained input from a DataGlove mounted with a Polhemus sensor for a two-stage neural network. PowerGlove was used by Kadous [32] in SLR. Vogler and Metaxas [58] used magnetic sensors interchangeably with computer vision techniques to track movement of the wrists accurately. Hernandez-Rebollar et al. [23] presented a complex system composed of a DataGlove and accelerometers, called Accele Glove, for gathering position and shape of the fingers and the hands. Brashear et al. [7] combined the input obtained from the view of a hat mounted camera pointing downward and accelerometers on the wrists and the body to develop a totally wearable system.

Using coloured gloves to aid computer vision techniques is also popular in SLR. Holden et al. [26] used a colour-coded glove to extract 3D configuration of the hands from 2D images. Zhang et al. [67] made use of a multi-coloured glove for the dominant hand and a single coloured glove for the non-dominant hand since the most discriminative features of a sign are performed by the dominant hand. Fingers, palm and the back of the dominant hand are indicated by 7 different colours to enable detailed feature extraction.

In the earlier studies, skin colour segmentation was also widely used in addition to wearable accessories and sensors for tracking position and movement of the hands in the space. One of the earliest systems designed by Huang and Huang [28] tracked 2D hand motion using consecutive image frames. Their system requires that the speaker should wear dark clothes with long sleeves, the background should be uncluttered and change in the shape of the moving object should stay in the predefined limits. Some systems made use of depth cameras to get rid of restrictions on the background and the clothes of the signers [17, 20]. Han et al. [22] applied skin segmentation and tracking (SST) with improved colour model by integrating SVM active learner and region segmentation for hand and face tracking. Akyol and Alvarado [3] presented a

system for fast detection of the hands which modifies skin colour model with motion information obtained from motion history images (MHI). While most systems require only one subject in an isolated environment Hong, Setiawan and Lee [27] overcame this issue by combining skin-based segmentation with contour-based segmentation to subtract the background and detect the main subject among multiple subjects. Nevertheless, their system works only for some predefined gestures.

One issue with skin segmentation is occlusion between the head and the hands. Holden et al. [25] dealed with this issue by making use of the snake algorithm [24] in combination with motion information to detect the contour of the moving foreground objects. Zieren and Kraiss [68] followed multiple hypothesis in parallel in tracking state and then, chose the best one for retrospective handling of the ambiguities at the sign level.

Skin segmentation is considered to be a more natural way of tracking hands since the systems requiring the users to wear accessories and sensors are not convenient in real life situations. Moreover, requiring complex environmental setup, systems using sensors are not portable. Thus, some systems are extended to make use of skin segmentation instead of wearable sensors. Imagawa et al. [29] improved their SLR system [43], which requires the users to wear coloured gloves in order to track hands, to track hands using skin colour segmentation and get rid of occlusion between the head and the hands by making use of motion difference blobs. Awad et al. [5] also extended their previous work [53] by using a combination of colour, motion and position information for segmentation of the body parts.

Placing the camera on a different position than the position of an observer, i.e. opposite of the signer, or using combination of cameras positioned in different places are some of the other preferred methods for hand tracking in SLR. Starner and Pentland [54] developed two systems each of which places the camera on different positions. In the first system, the camera is mounted in the opposite of the signer as it has second-person view whereas the other system tracks the hands using a hat-mounted camera pointing downward to see the hands. Vogler and Metaxas [59] employed a combination of three orthogonally placed cameras to estimate 3D shape and motion of the arms using multiple images collected from them. Although the orthogonally

placed cameras provide more detail for accurate detection of the depth and the hand shape compared to a single camera, the system is not portable and it requires complex calibration.

Another approach is to use stereo input in order to get benefit of the depth data as well as RGB images. A system adopting this approach is developed by Muñoz-Salinas et al. [47] by extending binary silhouettes with the depth data (creating depth silhouettes) for gesture recognition. Their system assumes that the position of the signer is known. Grzeszcuk et al. [19] also employed a stereo camera system for their system. They, firstly, subtract the background and detect the orientation of the arm and 3D pose of the hand. Then, a planar homographic transformation is applied to gather the frontal view of the hand. After necessary normalisation on the frontal view is done, the hands are segmented based on a probability density estimate. One drawback of the system is the restriction that the stereo camera should view only the arms of the signer, which is difficult to arrange in real world situations.

With the release of the Kinect® sensor, many SLR systems have started using it. Having integrated colour camera, infrared emitter, and open source framework for automatic skeleton tracking, it is a good choice for data acquisition in SLR. For example, Lang, Block and Rojas [36] created a system that uses automatic skeleton tracking features of Kinect® to train an HMM and recognise sign gestures. Ong et al. [50] used an isolated Greek Sign Language (GSL) dataset consisting of the frames of the upper body skeleton captured by Kinect® to test their novel classification method. Doliotis et al. [13] utilised Kinect® device as a depth camera to compare gesture recognition performance using colour and depth information by applying tests on a limited gesture dataset defined by them.

Some systems made use of methods other than commonly used ones to track head and hands. Kadir et al. [31] employed two weak classifiers trained using boosting techniques to detect position of head and hands. Buehler et al. [8] proposed a method that uses inference methods to reduce the cost in generative models of the images. Images were the frames of continuous videos taken from TV broadcasts.

3.1.2 Non-tracking Methods

Because tracking head and hands is a non-trivial task, some studies proposed non-tracking methods to recognise signs globally in order to avoid the tracking cost. Wong and Cipolla [62] derived a motion feature vector from motion gradient orientation images obtained from raw video. Then, the feature vector is classified using sparse Bayesian classifier trained to estimate 10 elementary gestures. Zahedi et al. [65] also extracted a feature set from raw video to classify 50 ASL words using HMM. The feature set composed of the images obtained by multiplying down-scaled version of the original images by binary skin-intensity images, and horizontal and vertical derivatives of these images. A similar method is used in [66] to recognise 10 ASL words. In another study by Zahedi et al. [64], besides appearance-based features, geometric features consisting of 34 features of the dominant hand are also extracted. These geometric features are divided into 4 feature groups which are basic geometric features, moments, Hu moments and combined geometric features. Cooper and Bowden [10] broke signs down into 5 main visemes (like phonemes in speech) and made use of 3 of them which are placement, movement and arrangement. These visemes were extracted from video using 3 different viseme level classifiers which do not require tracking of hands. Non-tracking classifying methods get rid of complexity and computational cost of tracking process. However, they also lose the discriminative nature of hand tracking. Consequently, despite the fact that run-time complexity of non-tracking methods are better than tracking ones, their recognition performance is not as good as that of tracking methods.

3.1.3 Hand Shape Recognition

Most of the SLR systems have focused on location and movement of the hands whereas shape and orientation of the hands are also effective in conveying meaning of the signs. Therefore, some studies have investigated hands in more detail for SLR. Kelly et al. [33] presented a system which estimates hand posture from the contour of the hand which is extracted by segmenting the hand on binary images. They defined a size function to detect similar and distinguish different hand shapes. Ong and Bowden [49] developed a boosted classifier tree consisting of cascade of classifier

layers for hand shape detection. Going deeper in branches of the tree detector classifiers trained using smaller and more specific set of images are employed to classify the input image into more specific hand shape group. Hand shapes are grouped using a shape context which is created by constructing compact shape descriptors from samples extracted from the contour of a hand shape. Hamada et al. [21] also used hand contour for matching hand shapes. Their model includes position and velocity of the hand in case of not being able to detect contour due to fast movement of the hand. Fillbrandt, Akyol and Kraiss [16] proposed a method to infer 3D posture of the hand and finger constellation by modelling them by a set of 2D appearance models. Liu and Fujimura [42] deployed a depth camera as input device and detected hand shapes by matching unlabelled input data to a hand shape in the training set using Chamfer Distance. They also extracted motion and orientation of the hands for better recognition of hand gestures. It is expected that adding hand shape information into recognition process result in better recognition performance. Yet, it may not be the case always due to the fact that performance of sign recognition is highly related to performance of hand shape recognition. A hand shape classifier having inadequate recognition rate will possibly cause inadequate sign recognition rates.

3.2 Classification Methods

In section 3.1 an overview of SLR system was given based on employed data acquisition and feature extraction methods. It is obvious that extracting necessary data is a crucial part in SLR. However, the success of recognition process depends on not only the performance of the feature extraction method but also the performance of the classification method. In this section, how SLR systems classify signs after features are extracted is reviewed. In other words, an overview of SLR systems is given based on the classification methods used.

Earlier systems employed variations of neural networks (NN) for classification of the extracted features. Kim, Jang and Bien [35] used fuzzy min max neural network (FMMNN) to classify the features extracted from a data glove which is improved with a Polhemus sensor to capture constellation of fingers as well as 3D position of the hands. FMMNN yielded a success rate of 85% for 25 selected words of Korean

Sign Language (KSL) in person-dependent experiments. Lee et al. [37] presented a real-time recognition system consisting of four stages. In the first stage, constellation of the fingers with the positions and the orientations of the hands are obtained using CyberGlove and Polhemus sensors. The system segments continuous motion to extract isolated signs by removing meaningless motion occurring when the hands move from the end of the current gesture to the beginning of the next gesture. Then, directions of the hands are classified to one of the predefined direction classes using feature extraction and fuzzy inference rules. In the last stage, the FMMNN classifier is deployed to classify 131 words and 31 manual alphabet of KSL. Experiments conducted with single signers achieved an average recognition rate of 80.1%. Waldron and Kim [61] developed a two-stage artificial neural network (ANN) system making use of a data glove mounted with a Polhemus sensor for data collection. In the first stage, phonemes are recognised using back propagation (BP) network consisting of four modules, one for each phonemes (hand shape, location, orientation and movement). Concatenated phonemic vectors are passed to the next stage in which signs are recognised using two different techniques, BP network and self-organising network (SON). Experiments are conducted in a multiple-signer environment, not necessarily signer-independent, and BP network achieved a recognition rate of 86.2% whereas that of SON was 78%. Huang and Huang [28] developed a system which applies 3D Hopfield neural network (HNN) to classify features collected by 2D model-based tracking. Their system recognise 15 sign gestures with a rate of 91%. Yang et al. [63] employed time delay neural networks (TDNN) to classify 2D motion trajectories of hands extracted from video using motion segmentation, skin segmentation and geometric analysis.

Due to its success in speech recognition, and similarities with the problem of speech recognition and SLR, HMM has been being used widely in SLR since mid 90's. HMM was used by systems having different data collection methods since it is applicable to a large variety of temporal data due to its ability of automatic segmentation of data during training and recognition. In some systems, HMM was employed to classify features extracted directly from video [18, 54] while some systems made use of it to classify data collected from a data glove [58]. Ouhyoung and Liang [51] developed a system consisting of 3 functional stages, posture analysis, gesture matching

and sentence matching. They employed HMM in posture analysis state to select possible postures corresponding to the input gathered from a data glove. After candidate postures are found HMM is applied to match this postures to the predefined gestures. Then, forward-backward procedure is applied by adding probabilities of gesture sequences into the process to estimate predefined sentences. Kim et al. [34] proposed a method for continuous KSL recognition using a colour camera. Their method creates a fuzzy partitioning using speed and change of speed of the hands during utterances of signs and implements a state automata using the created partitioning to segment isolated words from continuous sentences. Then, they employed HMM to classify isolated words. Average recognition rate of their method was 94.0% in the experiments done with 15 predefined sentences.

Although HMM is a good choice for SLR, it imposes some limitations. Requirement of a large training set and the inability of weighting features according to their importance are some of these limitations. It is possible to overcome these disadvantages of HMM by combining it with other methods. Vogler and Metaxas [59] coupled HMM with computer vision techniques in order to improve its performance by extracting geometric primitives of signs and taking advantage of assigning weight to the features relative to their importance. In another work of Vogler and Metaxas [60], Parallel HMM (PaHMM) was presented as an SLR method to surmount scalability issues of HMM in large lexicon databases. The idea behind it is that processes evolving in time independently from each other create independent output. PaHMMs employ one HMM for each independent process where state probabilities and outputs of these HMMs are not influenced from each other. In the experiments conducted with 99 sentences over a vocabulary of 22 signs, PaHMM performed 84.85% and 94.23% in sentence and word recognition, respectively, while regular HMM performed 80.81% and 93.27%. These results show that PaHMMs outperform regular HMMs in recognition accuracy besides scalability.

During the evolution of SLR, methods other than NNs and HMMs were also proposed. Ong et al. [50] presented a multi-class classifier based on sequential pattern trees for SLR. Their classifier performed 55.4% in signer-independent experiments conducted with a dataset consisting of 40 German signs captured by Kinect® sensor and 74.1% in signer-dependent experiments conducted using a video dataset of

982 Greek signs. Kadous [32] compared instance-based learning (IBL) and decision trees with different set of features extracted using a data glove. He, then, synthesised the best-performing features and obtained an accuracy of 80% for IBL and 55% for decision trees.

In most of the studies, classification methods which require external training were preferred. Although these methods are useful because of the small computational cost during the run-time, adding new samples to the lexicon requires complicated training phases to be repeated and the models to be created again. Using instance-based learning methods dispels the costly training phase; however, recognition on large databases is likely to result in high computational cost during classification which may prevent real-time recognition of signs on large databases.

3.3 Finger Spelling

All studies which have been reviewed so far proposed methods for SLR using word level classification. It is also possible to communicate letter by letter instead of word by word because each letter has an equivalent sign. Consequently, some SLR systems have focused on recognising finger spelled letters to generate words. Sign language letters are different from sign language words in terms of the features conveying the meaning. While words are described using 4 manual and 1 non-manual feature (see Section 2.2) letters are described using only 2 manual features, hand shape and orientation. Discrimination between letters are provided especially by finger constellation. To this end, SLR systems recognising sign language letters examined the hands to obtain hand shape, hand orientation and finger constellation.

Uebersax et al. [56] recognised letters by classifying the input data acquired from a depth camera using a classifier based on average neighbourhood margin maximisation (ANMM), depth difference and hand rotation. Words were assigned a score which is the sum of the confidences of the estimated letters. Word estimation occurs when the difference between the scores of the first most likely word and the second most likely word exceeds a predefined threshold. A word lexicon database is also employed to correct possible errors in the letter recognition step. Feris et al. [15] modified a colour

camera by placing four flashes on the left, right, up and down side of it. Their purpose was to use hand images illuminated by different light sources to detect depth edges which is discriminative in recognising letters. The problem with their design is that 5 photos, one for ambient illumination and one for each of 4 flashes, are taken for each frame which is impossible when the hand moves fast, which is common in utterances of signs. Another gesture recognition method using shadow was developed by Segen and Kumar [52]. Differently, their method made use of only one light source to estimate predefined 4 hand gestures which are very simple compared to sign language letters. Jerde, Soechting and Flanders [30] employed a glove with embedded sensors to detect degrees of 17 joints in the hand. Then, they proposed a method based on bio-mechanical and neuromuscular constraints for reduction in number of joints and compared their method to PCA. Their method did much better than PCA resulting in very small impairment in classification besides performance improvement due to reduced dimensionality in data.

Because the set of gestures in sign language alphabet is much smaller than the sign language lexicon, focusing on recognition of signs letter by letter seems a logical thing to do. However, it is a non-trivial task since recognition of letters requires more detailed investigation of hands compared to the recognition of words. Moreover, constraints imposed by some proposed methods are not applicable in real world situations. Above all, although communicating letter by letter is preferred in situations that the sign corresponding to an intended word is unknown, it is not convenient and preferred for whole communication. Thus, SLR systems intended to use only manual sign alphabet are not well-suited way of building a communication channel for deaf people.

Table 3.1: A list of SLR systems. (ABBRV: CR: Continuous sign recognition, SL: Sentence level recognition, WL: Word level recognition, SD: Signer-dependent experiments, SI: Signer-independent experiments, FS: Finger spelling recognition)

Author	**Data Acquisition**	**Classification**	**Dataset**	**Results**
Waldron and Kim [61]	Data glove and Polhemus sensor	BP Network[1], SON[2]	14 ASL words	86%[1], 84%[2]
Kadous [32]	PowerGlove	IBL	95 Auslan words	80%
Vogler and Metaxas [58]	Magnetic sensors and computer vision	HMM	53 ASL words	87.71% (CR)
Hernandez-Rebollar et al. [23]	DataGlove and accelerometers	A 3-level hierarchical classifier	26 ASL alphabet gestures	96.3%
Brashear et al. [7]	Accelerometers and hat-mounted camera	HMM	5 ASL gestures	90.48%
Holden et al. [26]	Colour-coded gloves	HMU	22 Auslan words	95%
Zhang et al. [67]	Multi-coloured gloves	TMDHMM	439 Chinese SL words	92.5%
Huang and Huang [28]	Skin segmentation	HNN	15 Taiwanness SL words	91%
Holden et al. [25]	Skin segmentation with the snake algorithm	HMM	163 Auslan words	97% (SL)

Zieren and Kraiss [68]	Multiple hypothesis	HMM	232 BSL words[3], 221 BSL words[4], 18 BSL words from 6 signers[5]	99.3%[3] (SD), 44.1%[4] (SI), 87.8%[5]
Starner and Pentland [54]	Hat-mounted camera	HMM	40 ASL words	97% (CR)
Vogler and Metaxas [59]	3 orthogonally-placed cameras	HMM	53 ASL words	87.71% (CR)
Muñoz-Salinas et al. [47]	MLSHI	SVM	10 defined gestures	86.83%
Grzeszcuk et al. [19]	Stereo camera system	Statistical moments	6 defined gestures	96%
Lang, Block and Rojas [36]	Kinect® skeleton tracking	HMM	25 German SL words	97%
Ong et al. [50]	Kinect® skeleton tracking	SP Trees	40 German SL words	55.4% (SI)
Kadir et al. [31]	Boosting 2 weak classifiers	A two-stage classifier	164 BSL words	92%
Wong and Cipolla [62]	Motion gradient orientation images	Bayesian classifier	10 defined gestures	90%

Zahedi et al. [65]	Raw video	HMM	50 ASL words	82.8%
Cooper and Bowden [10]	3 different classifiers	A 2-level classifier	164 BSL words	74.3%
Kelly et al. [33]	Hand postures from video	SVM	10 static gestures[6], 23 ISL letters[7]	91.8%[6], 97.3%[7]
Kim et al. [35]	Data glove	FMMNN	25 KSL words	25%
Lee et al. [37]	CyberGlove and Polhemus sensor	FMMNN	131 KSL words	80.1%
Yang et al. [63]	Motion and skin segmentation	TDNN	40 ASL words	93.42%
Kim et al. [34]	Fuzzy partitioning using speed	HMM	15 KSL sentences	94% (SL)
Vogler and Metaxas [60]	MotionStar 3D tracking	PaHMM	99 sentences over 22 signs	84.9% (SL), 94.2% (WL)
Uebersax et al. [56]	Depth camera	ANMM	56 ASL words	87.8% (FS)
Feris et al. [15]	Colour camera with four flashes	Nearest-neighbour	ASL alphabet except 'J' and 'Z'	96% (FS)
Jerde et al. [30]	CyberGlove with sensors	Discriminant analysis	26 ASL letters	95% (FS)

CHAPTER 4

SLR USING REDUCED RESOLUTION TRAJECTORIES OF JOINTS

In this chapter, the proposed method is described in detail. In Section 4.2, the overview of the system is given. Then, details of the collection and the normalisation of the sign data are described in Section 4.3. In Section 4.4, how the distance between two sign graphs is measured is explained. Then, details of the classification of the sign graphs based on measured distances are described in Section 4.5.

4.1 System Overview

The proposed system consists of a Kinect® sensor as input device and a computer to perform necessary computations. Figure 4.1 shows the overview of the recognition process. Firstly, sequences of skeletons for an isolated test sign are obtained from the Kinect® sensor. Then, the positions of the joints are normalised into joint paths using grid structure and they constitute a sign graph. After normalisation is done, the distance of the test sign to all of the signs in the training set are calculated using DTW distance of their sign graphs. Note that, the training set is stored as the set of sign graphs to get rid of unnecessary computations in the run time. Lastly, k-NN classification is applied to choose the k closest neighbours of the test sign and assign its label based on the label of the most voted one. Next sections describe each of these steps in detail.

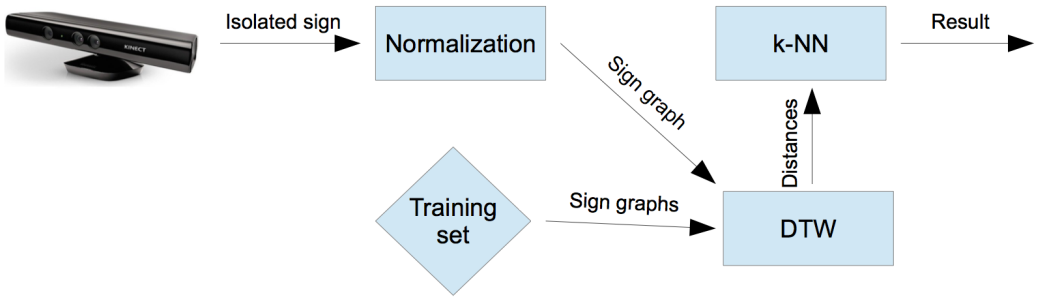


Figure 4.1: Overview of the recognition process

4.2 Gathering Sign Data

In the scope of this thesis, automatic SLR has been studied using human skeleton data tracked using Microsoft™ Kinect® device. Although there are different drivers processing raw data gathered from Kinect® and tracking different joints of human body, all of them tracks necessary joints (right and left hands and elbows) used for this study. Drivers return sequences of frames each of which contains a human skeleton data consisting of an array of joints which describes the position of human body joints in the 3D space (see Section 2.1 for frame rates and the list of the joints). That is, movement of human body joints in 3D space gathered frame by frame using MS Kinect® is used as the sign data in this thesis.

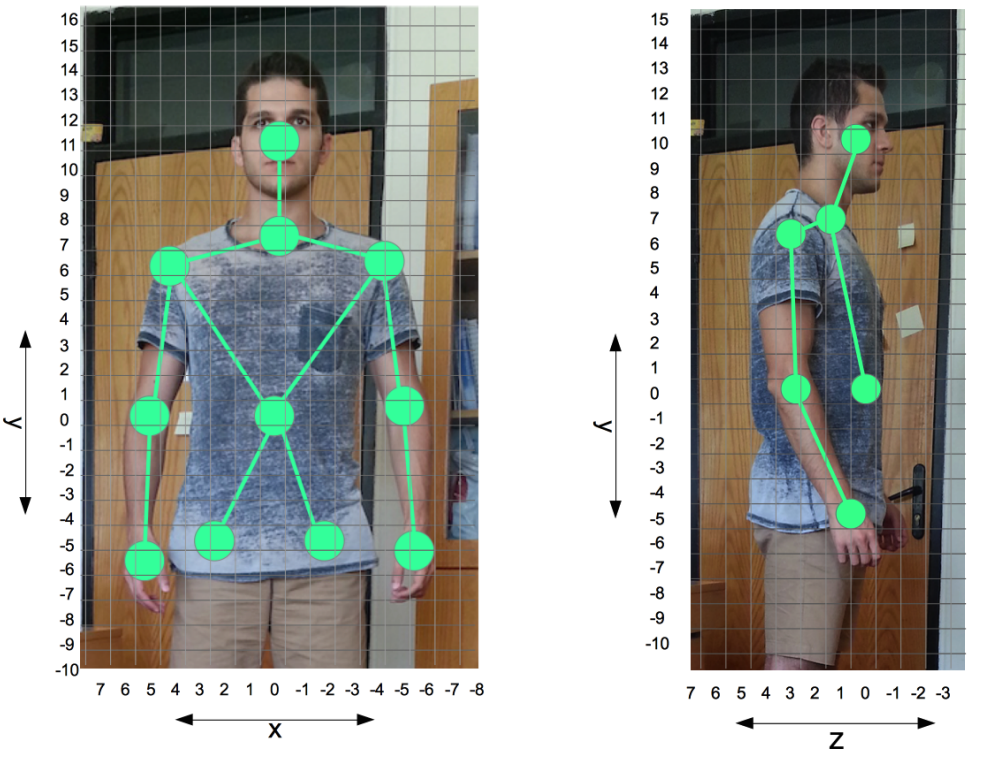
4.3 Normalising Sign Data with Enumerated Grid Structure

Because recognition process is done using the location and the movement of the joints, inter-signer differences are crucial for recognition of signs. Physical properties of signers like height, length of arms, length of neck, length between shoulders, etc. cause data gathered from different signers show non-standard characteristics. To get rid of this normalisation can be used which standardise data by placing samples in a predefined range. In our case, normalisation helps decreasing the influence of inter-signer differences on the recognition process.

Normalisation method applied in the scope of this study differs from what Lei Chen et al. [9] applied to make data invariant to both shifting and scaling. Making sign

data invariant to shifting may cause that recognition process cannot get benefit of one of the most discriminative features i.e., location. Therefore, using a normalisation method which does not make sign data invariant to shifting while normalising location is preferred. Another problem with the normalisation is making sign data totally invariant to scaling. Besides handling inter-signer differences in the size of movements, it may cause sign data lose sequences of locations, another discriminative feature of signs. As an example, think about two signs which have the same type of movement but different inception and conclusion positions for the active hand. Making data of these signs invariant to scaling induces locations of the active hands to have the same value during the production of the signs and causes these two signs no longer have a discriminative feature.

A novel normalisation method is proposed in this thesis to get rid of those problems arising from standard normalisation methods. It divides signing space of signers into grids and describes the trajectory of a joint as a sequence of grids (see Figure 4.2).



(a) Grid structure in x, y directions

(b) Grid structure in z, y directions

Figure 4.2: Grid structure of the signing space

Definition 4.1. *A **joint path** is a sequence* $J = (j_1, ..., j_{|J|})$*, denoted by* $\mathcal{P}$*, where*

$j_i = (j_{x_i}, j_{y_i}, j_{z_i})$ *for* $i \in [1 : |J|]$ *and* $j_{x_i}, j_{y_i}, j_{z_i} \in \mathbb{N}$ *represents a grid with its* x, y *and* z *positions and* $|J|$ *is the number of frames. Joint path for a joint contains grids through which that joint passes during the production of a sign.*

The following properties of the grid structure make it an appropriate normalisation method for automatic SLR:

1. The signer space is divided into grids based on the position of the signer so that differences in signer position according to Kinect® device have no influence on the recognition process.

2. The enumeration of grid cells is done based on a point on the body of the signer which ensures that location parameter of signs is invariant to inter-signer differences while variations of different signs in location parameter are preserved.

3. The size of the grid cells are adjusted according to length of arms of the signer which reduces influence of physical properties of signers on the movement parameter of signs.

4. Using grid structure to describe path of the joints reduces resolution of the movements. Reducing resolution helps recognition system define an acceptable error rate on the location and the movement of signs. Changing the size of the grids, it is possible to increase or decrease the defined acceptable error rate.

After normalisation, a sign is represented as a set of joint paths each of which keeps track of trajectory of a joint during the production of that sign. Because joint paths for a sign are obtained from the frames of a single skeleton, the number of frames (number of elements in a joint path sequence) is the same for all joint paths belonging to a single production of a sign. Though, it might have different values for distinct production of the same signs by the same signer. As a result, training and test data extracted from skeletons will possibly have some non-linear variations in time. Therefore, this thesis uses DTW which is an adequate technique for classifying trajectories of joints in the manner that it can evaluate joint paths independent of non-linear variations in the characteristics of the data.

4.4 A DTW Based Distance Determination

Various methods are available for aligning two time-dependent sequences which differs by an unknown shift in time. Cross-correlations and DTW are some examples of these methods. Cross-correlations is effectively used in signal processing for searching a small sequence in a larger one or finding the best alignment between two sequences in x-axis. Although it is well-suited to align two time-dependent sequences one of which is shifted linearly in time, it may align sequences having non-linear variations incorrectly. On the other hand, DTW is designed to align two time-dependent sequences having non-linear variations in time. Because, in SLR, samples from different subjects differ from each other non-linearly in time, DTW is the most convenient method to align these samples in time. Therefore, DTW is used to determine a measure of distance between two joint paths in order to get rid of non-linear variations in time.

The measure of distance between two joint paths is the cost of aligning them optimally using DTW. Let $U = (u_1, ..., u_N)$ and $Z = (z_1, ..., z_M)$ be two joint paths of length N and M, respectively. Then a distance function, $d : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{N}_{\geq 0}$, can be defined. Computation of d is the same as DTW. Let D be the accumulated cost matrix obtained from local cost matrix C of size $N \times M$ where $C(n, m) = c(u_n, z_m)$ is the distance between two elements (grid cells) of sequences U and Z. Then,

$$d(U, Z) = D(N, M). \tag{4.1}$$

Distance between grid cells can be calculated using L_1 distance, i.e. Manhattan distance, L_2 distance, i.e. Euclidean distance or sum of squared differences (SSD). Equation 4.2, Equation 4.3 and Equation 4.4 show the calculation of L_1 distance, L_2 distance and SSD, respectively.

$$c(a, b) = ||a, b||_{L_1} = (a_x - b_x) + (a_y - b_y) + (a_z - b_z), \tag{4.2}$$

$$c(a, b) = ||a, b||_{L_2} = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2}, \tag{4.3}$$

$$c(a, b) = ||a, b||_{SSD} = (a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2, \tag{4.4}$$

where a and b are two grid cells. The best distance measure will be chosen based on the experiments.

Because DTW is symmetric, the distance function d is also symmetric, i.e. the value returned does not change when the order of the parameters changes. Since the measure returned by d is the distance between the joint paths, it is called the dissimilarity function. It means that the lower the value it returns is, the more similar the joint paths are. The distance between the joint paths is used in determining the similarity between the signs which consist of a set of the joint paths. Normally, a single sign produced by a signer can be represented by the path of the joints that are listed in Section 2.1. However, only a small subset of them is used for recognition in the scope of this thesis, because some of the joints are non-discriminative for SLR and bring nothing more than computational cost.

Although all discriminative features of sign language structure are related to hands and face of the signers, elbows are also included in the set of joints which are used for determining similarity of signs. The reason why elbows show discriminative characteristics in distinct signs is that elbows are passively involved in some signs by being referenced by the active hand and that their position and movement have a connection with the palm orientation feature, that is signs differing in palm orientation probably differs also in location and movement of elbows. Therefore, not only trajectories of the hands but also that of elbows are included in the recognition process. Then, a sign data, is represented as set of joint paths.

Definition 4.2. *A **sign graph**, denoted by $\mathcal{S}$, is defined to be the set of joint paths extracted from skeleton data obtained during the production of a sign by a signer. It is a set of joint paths of hands and elbows.*

$$\mathcal{S} = \{\mathcal{S}^{rh}, \mathcal{S}^{lh}, \mathcal{S}^{re}, \mathcal{S}^{le}\}, \tag{4.5}$$

where $\mathcal{S}^{rh}, \mathcal{S}^{lh}, \mathcal{S}^{re}, \mathcal{S}^{le} \in \mathcal{P}$ represent joint paths of right hand, left hand, right elbow and left elbow, respectively.

Although sign graphs are not related to graph theory, it is named as a graph since it is a set of joint paths and these paths form a graph-like structure when connected to each other. Without a loss of generality, the set of joint paths representing a sign is

referred as sign graph in the rest of this thesis.

In order to classify sign graphs using k-NN, we need to define a dissimilarity measure between them. Let $V = \{V^{rh}, V^{lh}, V^{re}, V^{le}\}$ and $G = \{G^{rh}, G^{lh}, G^{re}, G^{le}\}$ be two sign graphs where $V^{rh}, V^{lh}, V^{re}, V^{le}, G^{rh}, G^{lh}, G^{re}, G^{le} \in \mathcal{P}$; then, a dissimilarity function, $\Delta : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{N}_{\geq 0}$ can be defined. To compute dissimilarity between two sign graphs, the joint path dissimilarity function, d, is used as follows:

$$\Delta(V,G) = \begin{cases} \infty & \text{if V, G different-handed.} \\ d(V^{rh}, G^{rh}) + d(V^{re}, G^{re}) & \text{if V, G are right-handed.} \\ d(V^{lh}, G^{lh}) + d(V^{le}, G^{le}) & \text{if V, G are left-handed.} \\ (d(V^{lh}, G^{lh}) + d(V^{le}, G^{le}) + & \\ \quad d(V^{rh}, G^{rh}) + d(V^{re}, G^{re}))/2 & \text{if V, G are both-handed.} \end{cases} \tag{4.6}$$

Because the dissimilarity function for joint paths is symmetric, the dissimilarity function for sign graphs is also symmetric. Being a symmetric function should not be confused with being able to recognise a sign independent of that the signer is left-handed or right-handed. It only guarantees that the function returns the same value independent of the order of parameters, sign graphs in this case. Note that the dissimilarity function Δ is infinite when both signs are one-handed but different hands are used or one of the signs is one-handed and the other one is two-handed. Thus, when it is given two sign graphs of the same sign produced by left-handed and right-handed signers, the distance between them is returned as infinity.

4.5 KNN Based Classification

Having a dissimilarity measure for sign graphs, we can classify them using k-NN. In our case, aim is to determine label of a sign graph. Classification is performed in two stages. In the first stage, k closest examples are chosen from the training set based on their distances to test example, measured using sign graph dissimilarity function, Δ. In the second stage, two different approaches are applied. The first approach returns the r most voted labels among the k chosen training examples while in the second approach, the r most weighted labels are returned such that the closer a training example is the more weight its label has.

Let $T = t_1, ..., t_k$ be the set of k closest training samples to the test sample q where $t_i, q \in \mathcal{S}$ for $i : [1 : k]$, $b_i \in$ be the label of t_i and $H = (h_1, ..., h_{|H|})$ be the set of labels. Then, the vote or the weight of each label $h_j \in H$ is calculated as:

$$\text{vote}(h_j) = \sum_{i=1}^{k} e(h_j, b_i), \tag{4.7}$$

$$\text{weight}(h_j) = \sum_{i=1}^{k} \frac{1}{\Delta(q, t_i)} e(h_j, b_i), \tag{4.8}$$

where $j \in |H|$, Δ is sign graph dissimilarity function and e is a function which return 1 if its parameters match, 0 otherwise. After the vote or the weight of each label is determined, labels are sorted based on these values and topmost r labels are returned. For $r = 1$, it is classical k-NN which returns only one result. For $r > 1$, more than one result is returned to improve recognition rate.

Since k-NN is a lazy learner, all computation is deferred until run-time, which means that there is no learning phase before run-time. Although the lazy learner characteristic of k-NN would cause excessive memory usage and high computational cost for large training sets, it does not yield such problems in our case due to the condensed structure of sign graphs. As sign graphs are obtained by extracting necessary information from raw sign data, they occupy less memory compared to raw sign data. Moreover, training set of sign graphs can be saved into and loaded from disc. Hence, processed training data can be loaded into memory before classification takes place and this reduces the computation cost of the classification process. In addition, the condensed structure of sign graphs also helps in reducing computational cost by improving performance of distance function, d.

The choice of k is determined according to the results of the experiments. Experiments are performed with different k values and the one giving the best results is chosen.

Algorithm 4.1 shows the whole process to determine the label of a test example using the training set consisting of sign graphs.

Algorithm 4.1: SLR Recognition

Input: Test sample q, Training set T of length $|T|$ where $T\,[i] \in \mathcal{S}$

Output: Label of q

```
// Variables
```
$S : Array[q.numberOfFrames]$ `// Sign graph of` q

$K : Array[k]$ `// Stores` k `closest sample`

```
// Normalization of q
```
for $i \leftarrow 1$ **to** $q.numberOfFrames$ **do**

  foreach *joint* j *in* (rh, re, lh, le) **do**

    $S[i]_x^j \leftarrow (int)(q[i]_x^j - basepoint_x)/gridSize$

    $S[i]_y^j \leftarrow (int)(q[i]_y^j - basepoint_y)/gridSize$

    $S[i]_z^j \leftarrow (int)(q[i]_z^j - basepoint_z)/gridSize$

  end

end

```
// Choose the k closest example
```
for $i \leftarrow 1$ **to** T **do**

  if $\Delta(S, T[i]) <$ *any of elements in* K **then**

    $K.insert(T[i])$ `// replaces an element`

  end

end

```
// Compute the votes and the weights of the labels
```
for $i \leftarrow 1$ **to** k **do**

  `// Increase votes or add weight`

  vote [`Label`$(T[i])$] $\leftarrow$ vote [`Label`$(T[i])$] + 1

  weight [`Label`$(T[i])$] $\leftarrow$ weight [`Label`$(T[i])$] + $(1/\Delta(S, T[i]))$

end

```
// Sort in descending order
```
`SortDesc` (vote)

`SortDesc` (weight)

```
// Return the label having the most vote or weight
```
labels $\leftarrow$ vote $(1 : r)$ `// or`

labels $\leftarrow$ weight $(1 : r)$

CHAPTER 5

EXPERIMENTS AND RESULTS

In this chapter, details of the experiments are given. In Section 5.1, the dataset used in the experiments is introduced. In Section 5.2, optional parameters of the methods used in this thesis are analysed to find the optimal values. Then, results of the experiments performed with the chosen parameters are given in Section 5.3. Note that, all results are given as the percentage of the recognition rate. Recognition rate gives the ratio of the number of classifications in which the correct result is between returned r results to the number of total classifications performed.

5.1 Dataset: DGS Kinect® 40

The dataset (also used by Ong et al. [50]) consists of German Sign Language signs captured using Kinect® sensor. Sign data are stored as frame-by-frame skeletons of a signer. OpenNI® framework is used for skeleton tracking. Therefore, movements of the joints described in Section 2.1 in 3D space can be extracted easily from the set. In the scope of this study, joint paths of the hands and the elbows are extracted from skeleton data to generate sign graphs.

The dataset contains 40 signs, a mixture of similar and dissimilar signs, each of which is produced 5.6 times on average by 13 different signers.[1] Since signers of different age and physical properties generated the data set, it includes examples with inter-signer differences. Moreover, all the signers in this set are non-native signers which

[1] The dataset contains signs from 15 different signers, actually. Because data of 2 signers have inconsistencies in arm length during the utterances of some signs, they are excluded from the dataset.

causes the set to have intra-signer differences. In other words, distinct production of the same sign by the same signer may also have some differences. Containing complex data in terms of number of signs and properties of signers, this data set is considered as an appropriate one to test the proposed method.

5.2 Analysing the Parameters of The Proposed Method

Some choices should be made for the values of variable parameters through the way of sign recognition using the proposed method. The parameters having the most noticeable influence on recognition process are the value of k in k-NN and the size of the grid cells which is used in the normalisation of the raw skeleton data into the joint paths. Another parameter is the way of computing votes in k-NN. Available choices for this parameter are voting by counting and voting by using weight of the neighbours. All of these parameters are decided based on the experimental results.

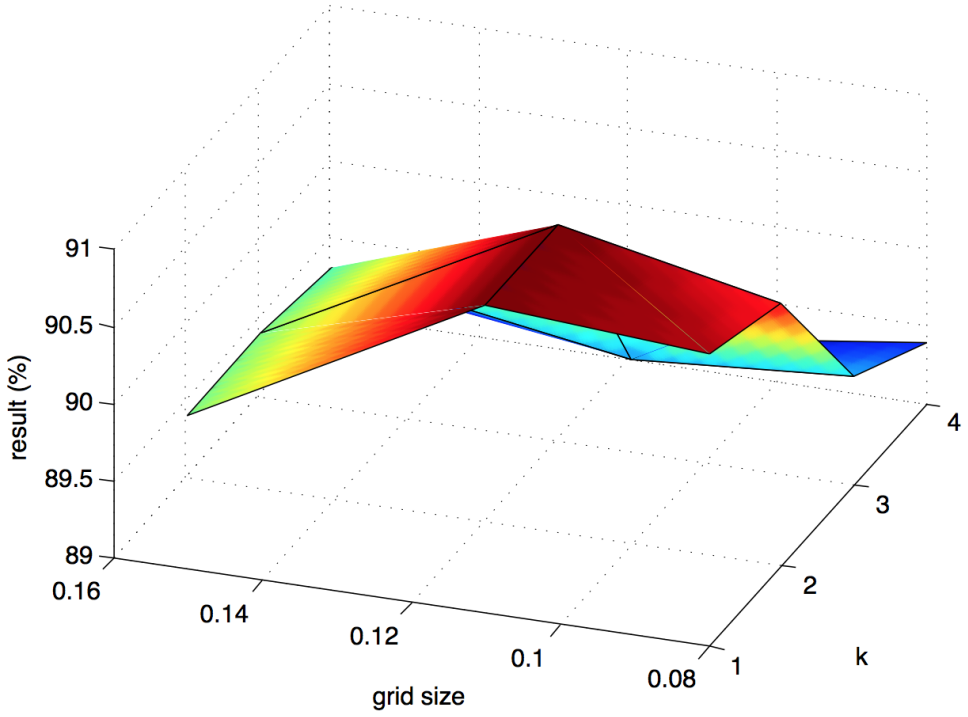


Figure 5.1: Visualisation of the signer-dependent results with different k and grid size values

Table 5.1: Average results of the signer-dependent experiments and standard deviations for different values of k and grid size

Grid size	k	Result (%)	Std. deviation (%)
0.08	1	90.9	4.4
0.08	2	90.7	4.4
0.08	3	89.7	4.5
0.08	4	89.4	4.6
0.11	1	91.0	3.8
0.11	2	91.0	3.8
0.11	3	89.6	4.2
0.11	4	89.2	4.6
0.15	1	90.0	3.6
0.15	2	90.0	3.6
0.15	3	88.9	4.1
0.15	4	88.2	4.0

Because the influence of the value of k and the grid size on the recognition process is not mutually exclusive, experiments are performed by changing the value of the both to find the most appropriate combination of them. Values of k and grid size are decided separately for signer-dependent and signer-independent experiments. Table 5.1 shows the average results of signer-dependent experiments conducted with different k and grid size values. It also gives the standard deviation of the signer-based results which shows how much results differ from each other as different signers become the test object. Table 5.2 shows the same thing for the signer-independent experiments. Figure 5.1 and Figure 5.2 visualise Table 5.1 and Table 5.2, respectively. The lower the standard deviation is the higher consistency the recognition process has. Measure of grid size is given as the ratio of its side length to the arm length of the signer. Therefore, it is not constant, but changes proportionally to the length of the signer. According to the results, there exists a gap close to 3 percent between the best and the worst combinations of the value of k and the grid size for both the signer-dependent and the signer-independent experiments. This shows how important choosing the right values for these parameters is. In the signer-dependent experiments, the best results are obtained when grid size $= 0.11$ and $k = 1$. In the signer-independent experiments, the best results are obtained when grid size $= 0.11$ and $k = 10$.

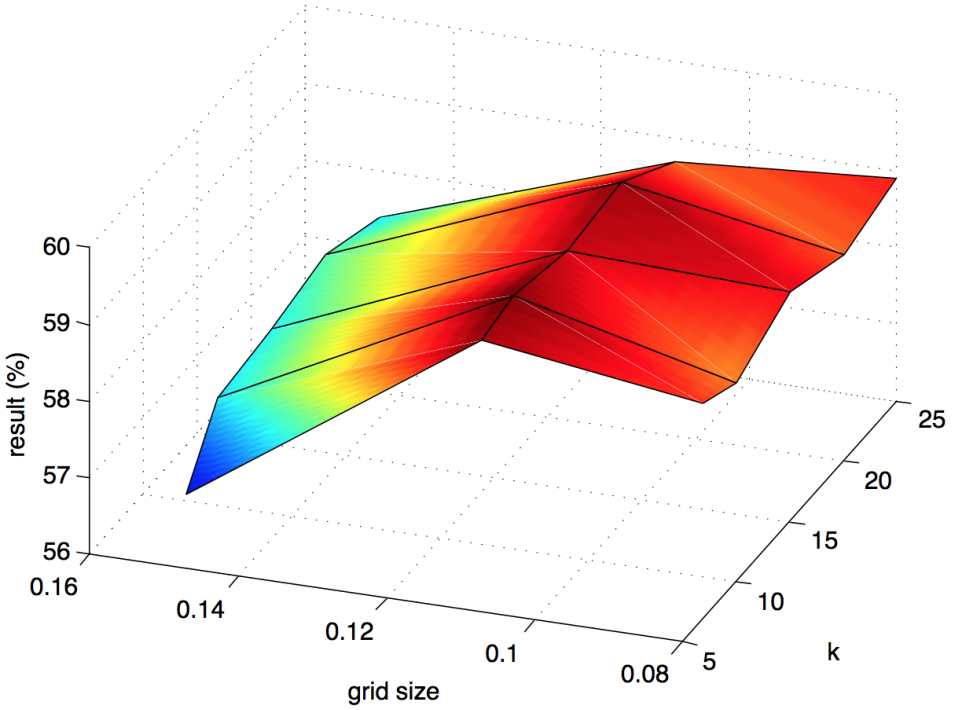


Figure 5.2: Visualisation of the signer-independent results with different k and grid size values

The other parameter to be decided is the method to be used (Equation 4.7 or 4.8) for deciding label of a sign after nearest neighbours are determined. In Equation 4.7, all k examples have the same effect on determining the label independent of distance. On the other hand, in Equation 4.8, the weight of each sample decreases as the distance between it and the test sample increases. Results in the Table 5.2 are obtained using Equation 4.8. To decide which of them yields better results, top 5 combinations in Table 5.2 are chosen and experiments are repeated with the first equation (see Figure 5.3).

Table 5.3 compares the results and the standard deviations obtained by using functions in Equation 4.7 and 4.8 in the second phase of k-NN. The weighted results are obtained with the latter equation. It is clearly seen that the performance decreases when count function is applied. On the other hand, standard deviations do not show any stable tendency for increasing or decreasing; that is, equations used in the second phase of k-NN have no specific effect on standard deviation of signer-based results. Taking these experiments into consideration, it can be concluded that using weight

Table 5.2: Average results of the signer-independent experiments and standard deviations for different values of k and grid size

Grid size	k	Result (%)	Std. deviation (%)
0.08	7	58.8	6.5
0.08	10	58.6	6.8
0.08	15	59.0	7.0
0.08	20	58.7	6.8
0.08	25	58.9	7.4
0.11	7	59.2	7.0
0.11	10	59.3	6.6
0.11	15	59.1	6.9
0.11	20	59.2	6.7
0.11	25	58.7	7.4
0.15	7	56.6	7.1
0.15	10	57.4	7.6
0.15	15	57.5	7.2
0.15	20	57.7	7.2
0.15	20	57.4	6.5

function instead of count function is much more desirable.

Table 5.3: Results and std. deviations of the experiments performed with weight and count functions. Top 5 grid size-k combinations in the Table 5.2 are used.

		Result (%)		Std. deviation (%)	
Grid size	k	Weight	Count	Weight	Count
0.11	10	59.3	58.1	6.6	6.6
0.11	20	59.2	58.1	6.7	7.1
0.11	7	59.2	58.1	7.0	6.5
0.11	15	59.1	58.4	6.9	6.7
0.08	15	59.0	57.9	7.0	7.0

Besides the grid size, the value of k and the voting function, the distance measure between grid cells should also be decided based on the experiments. Three different distance measures are used in the experiments, Manhattan distance (L_1 distance), Euclidean distance (L_2 distance) and sum of squared differences (SSD). Equation 4.2, Equation 4.3 and Equation 4.4 show the calculation of these distance measures, respectively. Figure 5.4 compares the distance measures for the best 5 combinations

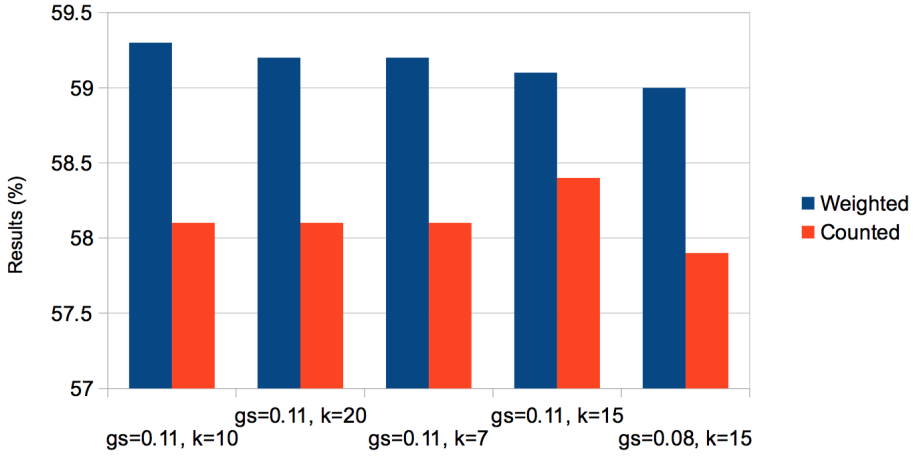


Figure 5.3: Visualisation of Table 5.3

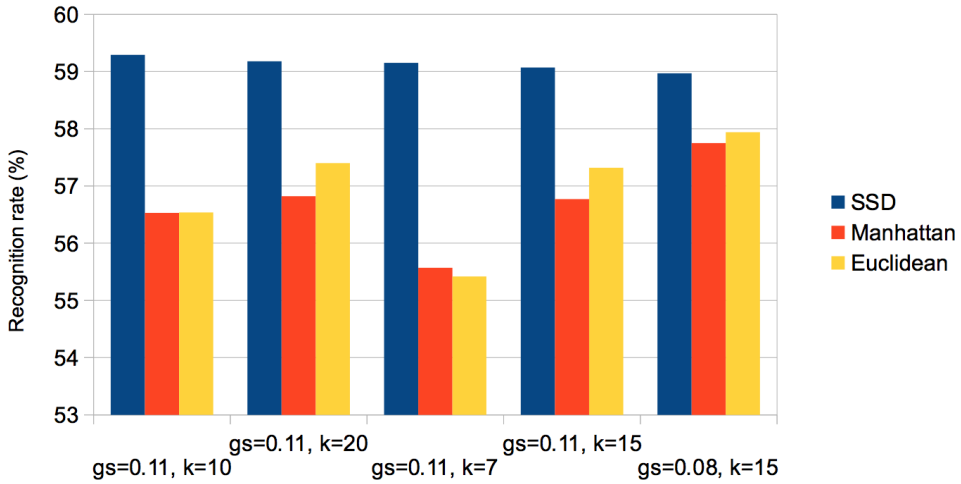


Figure 5.4: Comparison of two distance measures for the best 5 combinations of grid size and value of k

of grid size and value of k. According to the experiments, SSD performed better than Manhattan distance and Euclidean distance. Therefore, SSD is used as the distance measure between grid cells in the scope of this thesis.

5.3 Results

Because the data set consists of signs captured from different signers, it is possible to perform both signer-dependent and signer-independent experiments. Signer-dependent experiments evaluate the recognition performance of the system in situ-

ation where intra-signer differences are available while signer-independent experiments test the system for the situations where both intra-signer and inter-signer differences are available. The details of the experiments are given in the following subsections.

5.3.1 Signer-Dependent Experiments

In signer-dependent experiments, a test sample obtained from a signer is classified using the training set consisting of samples from that signer. It measures performance of the proposed method for the systems which is trained and used by only one signer (see Table 5.4). Experiments are conducted separately for each signer.

Table 5.4: Results (recognition rates) of the signer-dependent experiments for each signer (%)

P1	P2	P3	P4	P5	P6	P7
90.4	94.7	91.5	91.8	87.5	94.2	87.7

P8	P9	P10	P11	P12	P13	Avg.
90.5	96.6	81.8	91.8	94.4	90.8	91.0

It can be seen that the recognition rate for signer 10 is 10% below average. Because the dataset contains signs performed by non-native signers, it has high intra-signer differences. As a result, recognition performance of some signers may stay under the average. If the system were trained with native signers, recognition rates would be higher. Still, having average recognition rate of 91% for 40 signs, it can be concluded that the proposed method is convenient to be used with signer-dependent systems.

5.3.2 Signer-Independent Experiments

Signer-independent experiments are more suitable to the real world situations since the aim of an SLR system is to establish communication with different signers. In the signer-independent experiments, samples obtained from a single signer are reserved as test set and excluded from the remaining data set, known as the training set. This type of experiments are referred to as signer-based experiments in this study. Signer-

Table 5.5: Comparison of the recognition rates (%) of the signer-independent experiments for $r = 1$

	Markov Chain [12]	SP-Boosting [48]	SP Tree [50]	Sign Graph
P1	60.4	65.1	64.2	46.3
P2	50.2	49.6	50.0	64.1
P3	55.9	57.9	52.9	53.5
P4	54.1	56.6	57.1	60.7
P5	38.9	41.8	48.2	59.1
P6	62.3	64.2	65.2	61.1
P7	46.0	53.6	54.7	57.6
P8	53.1	53.5	54.6	66.7
P9	53.8	70.4	75.2	-
P10	50.0	52.2	50.2	64.7
P11	42.8	44.5	44.2	53.0
P12	52.7	56.5	57.9	65.2
P13	50.3	53.7	55.2	67.4
P14	38.5	44.2	45.7	51.5
Avg.	50.6	54.6	55.4	59.3
Std. dev.	7.13	8.2	8.4	6.6

based experiments are repeated for each signer to compute the average recognition rate. Performing signer-based experiments for each signer is a good example of cross-validation technique. Results in Table 5.2 and Table 5.3 are obtained by averaging results of the signer-based experiments for all signers. Standard deviations are also computed according to these experiments.

The dataset was also used by Eng-Jon Ong et. al. [50] to test their Sequential Pattern Tree classifier. They also conducted experiments by processing the data using SP-Boosting proposed by Ong and Bowden [48] and Markov Chain proposed by Cooper et al. [12] to compare their results. Thus, results of the proposed method in this study are compared to the results of their experiments. The experiments are performed in the same way, i.e. signer independent and with different values of r, using the same data set. Remember that r is the number of results the classifier returns.

Table 5.5 and Table 5.6 show average recognition rates and recognition rates of each signer-based experiment for Sign Graph, SP Tree [50], SP-Boosting [48] and Markov Chain [12] for different values of r. Beside recognition rates, standard deviations of

Table 5.6: Comparison of the recognition rates (%) of the signer-independent experiments for $r = 4$

	Markov Chain [12]	SP-Boosting [48]	SP Tree [50]	Sign Graph
P1	88.2	91.0	88.2	73.7
P2	80.6	85.4	85.4	92.2
P3	92.6	95.0	88.2	78.5
P4	79.5	85.4	85.8	77.1
P5	68.5	80.8	78.9	82.7
P6	89.7	92.6	91.6	85.6
P7	75.7	84.6	85.9	84.4
P8	84.0	91.0	93.3	90.1
P9	83.3	95.2	96.2	-
P10	84.9	88.3	84.9	86.8
P11	73.7	84.1	79.9	77.5
P12	78.7	90.3	94.7	87.5
P13	81.4	86.9	89.5	86.7
P14	72.1	84.1	82.3	80.0
Avg.	80.9	88.2	87.5	83.4
Std. dev.	6.9	4.4	5.3	5.7

the results of the signer-based experiment are also given.

Compared to other methods, Sign Graph, the proposed method, does better in $r = 1$ case. It outperforms SP Tree by %4, SP-Boosting by %5 and Markov Chain by %9. Furthermore, standard deviation values show that the Sign Graph method has more tolerance to inter-signer differences. One possible reason why Sign Graph performs better is its approach for handling movement of hands in detail. Another reason is the effect of DTW on decreasing inter-signer differences in time dimension. Despite better results of Sign Graph method in $r = 1$ case, recognition performances of SP-Boosting and SP Tree methods pass that of Sign Graph by %5 and %4, respectively, in $r = 4$ case. But, returning more than one result is inconsistent with the nature of Automatic Sign Language Recognition. In addition, it may not be feasible for the real world situations. Therefore, Table 5.5 is more meaningful to compare performance of the given classification methods.

Signer-independent experiments are also performed with smaller training sets to assess recognition performance as the number of training samples decreases. To this

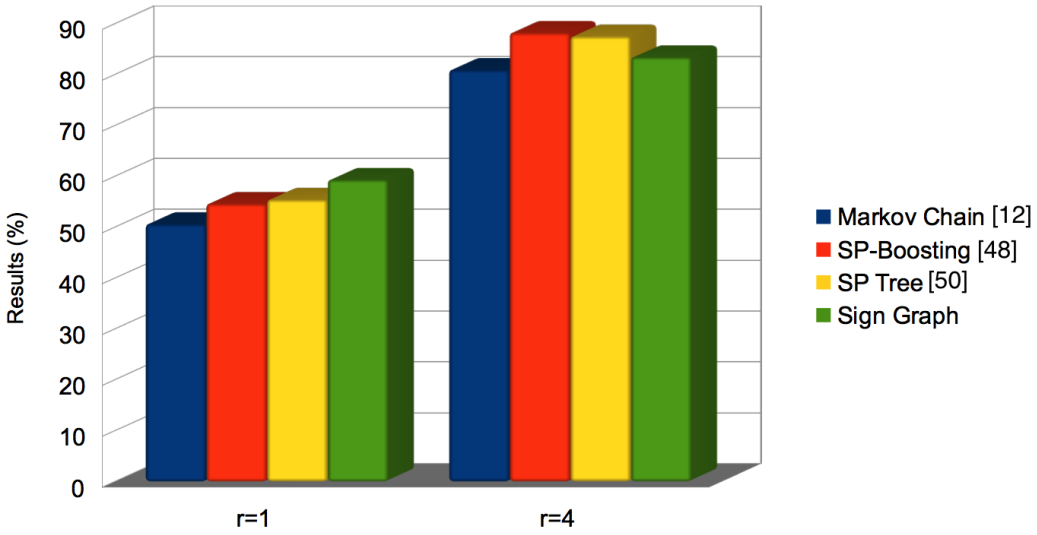


Figure 5.5: Avg. results of the signer-independent experiments with different r values

end, number of utterances of each sign by each signer is decreased gradually and the experiments are executed 7 times in each step. Random samples are extracted from signer's data in each execution for better generalisation of the results. This type of experiments are also called 7-fold cross validation. The value of k is equal to 10 and the grid size is equal to 0.11 for all of the experiments. Table 5.7 shows the recognition rates with different number of samples per sign for each user and Figure 5.6 visualise

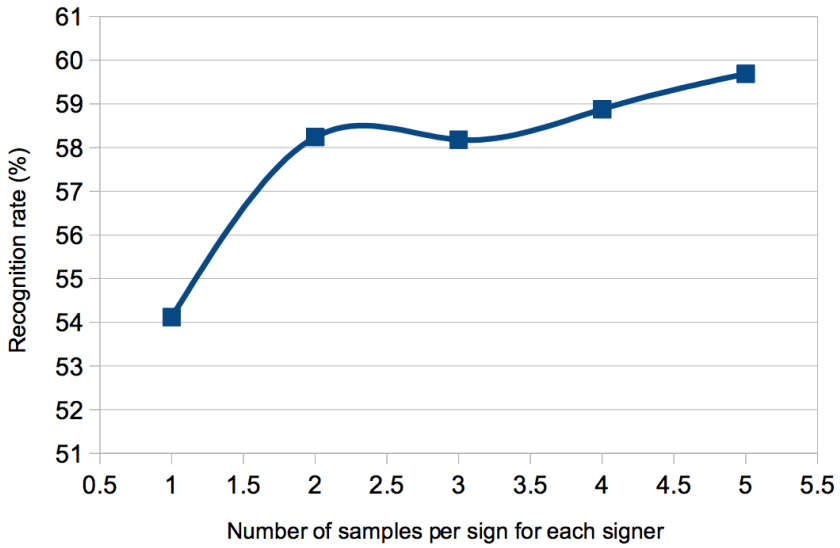


Figure 5.6: Recognition rates (%) with different number of samples per sign for each user

the change in recognition rates. Results show that the recognition performance of the system is affected slightly from the decrease in the number of training samples. The greatest gap between recognition rates occurs when the number of samples decrease to 1 from 2. The recognition rate is still acceptable when the number of samples per sign for each user is equal to 1. Therefore, it can be concluded that the system does not require large training sets to operate. It can perform well even when the sufficient number of the training samples are not available.

Table 5.7: Recognition rates (%) with different number of samples per sign for each user

Number of samples per sign for each signer	Recognition rate (%)
1	54.12
2	58.24
3	58.18
4	58.88
5	59.69

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this chapter, a summary of the proposed method is given and it is assessed by considering its strengths and weaknesses. In Section 6.1, the proposed method is summarised and advantages of it are described. In Section 6.2, weaknesses of the proposed method are described and what can be done in the future to get rid of these weaknesses are described.

6.1 Summary and Advantages

This thesis presented a novel approach for SLR which makes use of Kinect®. Using Kinect® sensor to capture sign gestures yields a system which requires more useful and simpler environmental setup compared to the systems using data gloves or combination of cameras. Kinect® is a ready-to-use sensor which can detect users without need of explicit calibration and start tracking immediately whereas other sensors, generally, require complex environmental setup and calibration. Some systems even require recalibration for separate users. In this manner, use of Kinect® sensor makes the proposed method advantageous in terms of system setup and mobility. The proposed system also gets benefit of Kinect® in terms of accuracy. Through its embedded depth camera, colour camera and infrared sensor, Kinect® provides locations of skeleton joints accurately. While mobility and ease of use are preferred properties, they do not have direct influence on recognition process. On the other hand, accuracy of the input device is crucial for the success of the recognition process. Another benefit of using Kinect® is real-time tracking of body joints. It would not be possible

to recognise signs in real-time unless the input device was not able to track skeletons in real-time. Employing Kinect® sensor as input device leads to easy environmental setup and accurate and real-time tracking. At the same time, it provides less detail compared to data gloves used by some SLR systems. Lack of information about position and constellation of the fingers affects recognition process negatively since hand shape is one of the most discriminative parameters of signs. As a future work, it is possible to improve the present system to estimate hand shape using the integrated depth and video camera of Kinect® without losing advantages of it.

In the scope of this thesis, normalisation of signs are done by dividing the signing space into grids. Then, a sign is described as a vector of sequences where each sequence contains the grid cells which a joint passes through during the utterance of a sign. Since the proposed normalisation method is robust to the changes in the arm length of the signers, the system is not affected by physical differences of the users. Moreover, using it the system gains the ability to compensate differences in the utterances of a sign due to different signing styles of the users. Besides, normalisation makes the system invariant to the shifting, that is the position of the signer according to the sensor. In real world situations, it is important for an SLR system to perform well independent of the signer. The proposed system achieves that by being invariant to the changes that arise when different signers use the system.

The proposed method employs k-NN, an instance-base learning (IBL) method, for the classification process. IBL methods have some advantages and disadvantages compared to other classification methods. One advantage of them is the absence of the explicit training step. The system gets rid of complex implementation of the training phase by using an IBL method. Moreover, enhancing the training set is much more easy than the classification methods which require explicit training since the models and the structures do not need to be recreated. Another advantage of k-NN over other classification methods is the size of the required training set. While methods like HMM and ANN need large training sets to create effective models and structures, k-NN can operate well on relatively small training sets by assigning optimal value to k. On the other side, deferring all computation until run time, use of IBL methods with very large training sets may cause excessive memory usage and high computational cost. As a result, recognising signs in real time may not be possible. However,

experiments show that the recognition performance of our method is effected slightly from the decrease in the number of training samples. Consequently, memory usage and computational cost will not be a problem since there is no need for large databases to obtain adequate recognition performance.

6.2 Limitations and Future Work

The system is not intended to replace a sign language interpreter due to some drawbacks. The first of them is the fact that the system expects isolated signs as input. Signers require to take their hands down between consecutive signs. It is not something that an interpreter can expect from the signers. This issue can be prevailed by improving the system to detect end and start points of the signs and recognise continuous signs. The other drawback is the fact that inter-signer differences cause the recognition performance of the system reduce dramatically. In other words, when a signer who is not participated in training becomes the test object the recognition rate decreases. The system does well in a signer-dependent environment with recognition rate of 91.0% while this number reduces to 59.3% in a signer-independent environment. It is possible to overcome this issue by adding a hand shape recogniser to the system. Moreover, tracking wrists and hands separately may also increase the recognition performance (Because the dataset contains only positions of the hands it was not possible to do that in the scope of this thesis.). Although the aim of the study was not to develop a commercial sign language recogniser, the proposed system can be used as a basis for a commercial SLR system.

REFERENCES

[1] World federation of the deaf. `http://wfdeaf.org/faq`. Last checked August 14 2014 20:07.

[2] Kinect for windows sdk beta launch. `http://channel9.msdn.com/Events/KinectSDK/BetaLaunch`, June 2011. Last checked July 24 2014 12:07.

[3] S. Akyol and P. Alvarado. Finding relevant image content for mobile sign language recognition. In *IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA), Rhodes*, pages 48–52, 2001.

[4] C. K. L. Anshul Gupta, Roberta Cozza. Market share analysis: Mobile phones, worldwide, 4q13 and 2013. Technical report, Gartner, Inc., February, 2014.

[5] G. Awad, J. Han, and A. Sutherland. A unified system for segmentation and tracking of face and hands in sign language recognition. In *18th International Conference on Pattern Recognition, 2006 (ICPR).*, volume 1, pages 239–242. IEEE, 2006.

[6] R. Battison. *Lexical Borrowing in American Sign Language.* ERIC, 1978.

[7] H. Brashear, T. Starner, P. Lukowicz, and H. Junker. Using multiple sensors for mobile sign language recognition. 2003.

[8] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language tv broadcasts. In *Proceedings of the 19th British Machine Vision Conference*, pages 1105–1114. BMVA Press, 2008.

[9] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2005.

[10] H. Cooper and R. Bowden. Large lexicon detection of sign language. In *Human–Computer Interaction*, pages 88–97. Springer, 2007.

[11] H. Cooper, B. Holt, and R. Bowden. Sign language recognition. In *Visual Analysis of Humans*, pages 539–562. Springer, 2011.

[12] H. Cooper, N. Pugeault, and R. Bowden. Reading the signs: A video based sign dictionary. In *International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 914–919. IEEE, 2011.

[13] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos. Comparing gesture recognition accuracy using color and depth information. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, page 20. ACM, 2011.

[14] C. Eisler. Starting february 1, 2012: Use the power of kinect for windows to change the world. `http://blogs.msdn.com/b/kinectforwindows/archive/2012/01/09/kinect-for-windows-commercial-program-announced.aspx`, January 2012. Last checked July 24 2014 12:16.

[15] R. Feris, M. Turk, R. Raskar, K. Tan, and G. Ohashi. Exploiting depth discontinuities for vision-based fingerspelling recognition. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04).*, pages 155–155. IEEE, 2004.

[16] H. Fillbrandt, S. Akyol, and K.-F. Kraiss. Extraction of 3d hand shape and posture from image sequences for sign language recognition. In *International Workshop on Analysis and Modeling of Faces and Gestures.*, pages 181–181. IEEE Computer Society, 2003.

[17] K. Fujimura and X. Liu. Sign recognition using depth image streams. In *7th International Conference on Automatic Face and Gesture Recognition (FGR).*, pages 381–386. IEEE, 2006.

[18] K. Grobel and M. Assan. Isolated sign language recognition using hidden markov models. In *International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation.*, volume 1, pages 162–167. IEEE, 1997.

[19] R. Grzeszcuk, G. Bradski, M. H. Chu, and J.-Y. Bouguet. Stereo based gesture recognition invariant to 3d pose and lighting. In *Proceedings of Conference on Computer Vision and Pattern Recognition.*, volume 1, pages 826–833. IEEE, 2000.

[20] S. Hadfield and R. Bowden. Generalised pose estimation using depth. In *Trends and Topics in Computer Vision*, pages 312–325. Springer, 2012.

[21] Y. Hamada, N. Shimada, and Y. Shirai. Hand shape estimation under complex backgrounds for sign language recognition. In *Proceedings of Sixth International Conference on Automatic Face and Gesture Recognition.*, pages 589–594. IEEE, 2004.

[22] J. Han, G. Awad, and A. Sutherland. Automatic skin segmentation and tracking in sign language recognition. *Computer Vision, IET*, 3(1):24–35, 2009.

[23] J. L. Hernandez-Rebollar, R. W. Lindeman, and N. Kyriakopoulos. A multi-class pattern recognition system for practical finger spelling translation. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, page 185. IEEE Computer Society, 2002.

[24] E. Holden and R. Owens. Segmenting occluded objects using a motion snake. In *The 6th Asian Conference on Computer Vision*, pages 342–347, 2004.

[25] E.-J. Holden, G. Lee, and R. Owens. Australian sign language recognition. *Machine Vision and Applications*, 16(5):312–320, 2005.

[26] E.-J. Holden and R. Owens. Visual sign language recognition. In *Multi-Image Analysis*, pages 270–287. Springer, 2001.

[27] S.-j. Hong, N. A. Setiawan, and C.-w. Lee. Real-time vision based gesture recognition for human-robot interaction. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 493–500. Springer, 2007.

[28] C.-L. Huang and W.-Y. Huang. Sign language recognition using model-based tracking and a 3d hopfield neural network. *Machine vision and applications*, 10(5-6):292–307, 1998.

[29] K. Imagawa, S. Lu, and S. Igi. Color-based hands tracking system for sign language recognition. In *Proceeding of Third IEEE International Conference on Automatic Face and Gesture Recognition.*, pages 462–467. IEEE, 1998.

[30] T. E. Jerde, J. F. Soechting, and M. Flanders. Biological constraints simplify the recognition of hand shapes. *Transactions on Biomedical Engineering.*, 50(2):265–269, 2003.

[31] T. Kadir, R. Bowden, E.-J. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *BMVC*, pages 1–10, 2004.

[32] M. W. Kadous. Machine recognition of auslan signs using powergloves: Towards large-lexicon recognition of sign language. In *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*, pages 165–174, 1996.

[33] D. Kelly, J. McDonald, and C. Markham. A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters*, 31(11):1359–1368, 2010.

[34] J.-B. Kim, K.-H. Park, W.-C. Bang, and Z. Z. Bien. Continuous gesture recognition system for korean sign language based on fuzzy logic and hidden markov model. In *Proceedings of International Conference on Fuzzy Systems (FUZZ-IEEE'02).*, volume 2, pages 1574–1579. IEEE, 2002.

[35] J.-S. Kim, W. Jang, and Z. Bien. A dynamic gesture recognition system for the korean sign language (ksl). *Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.*, 26(2):354–359, 1996.

[36] S. Lang, M. Block, and R. Rojas. Sign language recognition using kinect. In *Artificial Intelligence and Soft Computing*, pages 394–402. Springer, 2012.

[37] C.-S. Lee, Z. Bien, G.-T. Park, W. Jang, J.-S. Kim, and S.-K. Kim. Real-time recognition system of korean sign language based on elementary components. In *Proceedings of the Sixth International Conference on Fuzzy Systems.*, volume 3, pages 1463–1468. IEEE, 1997.

[38] M. Lewis, G. F. Paul, and C. D. F. Simons. *Ethnologue: Languages of the World*. SIL International, Dallas, Texas, seventeenth edition, 2014.

[39] S. K. Liddell. Structures for representing handshape and local movement at the phonemic level. *Theoretical issues in sign language research*, 1:37–65, 1990.

[40] S. K. Liddell. *Grammar, gesture, and meaning in American Sign Language*. Cambridge University Press, 2003.

[41] S. K. Liddell and R. E. Johnson. American sign language: The phonological base. *Sign language studies*, 64(1):195–277, 1989.

[42] X. Liu and K. Fujimura. Hand gesture recognition using depth data. In *Proceedings of Sixth International Conference on Automatic Face and Gesture Recognition.*, pages 529–534. IEEE, 2004.

[43] H. Matsuo, S. Igi, S. Lu, Y. Nagashima, Y. Takata, and T. Teshima. The recognition algorithm with non-contact for japanese sign language using morphological analysis. In *Gesture and Sign Language in Human-Computer Interaction*, pages 273–284. Springer, 1998.

[44] Microsoft Corporation. *Kinect for Windows SDK Managed Code Reference*.

[45] Microsoft Corporation. *Human Interface Guidelines*, 1.8 edition, 2013.

[46] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[47] R. Muñoz-Salinas, R. Medina-Carnicer, F. J. Madrid-Cuevas, and A. Carmona-Poyato. Depth silhouettes for gesture recognition. *Pattern Recognition Letters*, 29(3):319–329, 2008.

[48] E. Ong and R. Bowden. Learning sequential patterns for lipreading. In *Proceedings of the 22nd British Machine Vision Conference*, pages 55–1, 2011.

[49] E.-J. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Proceedings of Sixth International Conference on Automatic Face and Gesture Recognition.*, pages 889–894. IEEE, 2004.

[50] E.-J. Ong, H. Cooper, N. Pugeault, and R. Bowden. Sign language recognition using sequential pattern trees. In *International Conference on Computer Vision and Pattern Recognition (CVPR).*, pages 2200–2207. IEEE, 2012.

[51] M. Ouhyoung and R. Liang. A sign language recognition system using hidden markov model and context sensitive search. In *Proceedings of the ACM symposium on virtual reality software and technology*, pages 59–66, 1996.

[52] J. Segen and S. Kumar. Shadow gestures: 3d hand pose estimation using a single camera. In *Conference on Computer Vision and Pattern Recognition.*, volume 1. IEEE, 1999.

[53] A. Shamaie and A. Sutherland. Hand tracking in bimanual movements. *Image and Vision Computing*, 23(13):1131–1149, 2005.

[54] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *Transactions on Pattern Analysis and Machine Intelligence.*, 20(12):1371–1375, 1998.

[55] W. C. Stokoe. Sign language structure: An outline of the visual communication systems of the american deaf. *Studies in linguistics: Occasional papers (No. 8)*, 1960.

[56] D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool. Real-time sign language letter and word recognition from depth data. In *International Conference on Computer Vision Workshops (ICCV Workshops).*, pages 383–390. IEEE, 2011.

[57] C. Valli. *Linguistics of American sign language: An introduction*. Gallaudet University Press, 2000.

[58] C. Vogler and D. Metaxas. Adapting hidden markov models for asl recognition by using three-dimensional computer vision methods. In *International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation.*, volume 1, pages 156–161. IEEE, 1997.

[59] C. Vogler and D. Metaxas. Asl recognition based on a coupling between hmms and 3d motion analysis. In *Sixth International Conference on Computer Vision.*, pages 363–369. IEEE, 1998.

[60] C. Vogler and D. Metaxas. Parallel hidden markov models for american sign language recognition. In *The Proceedings of the Seventh International Conference on Computer Vision.*, volume 1, pages 116–122. IEEE, 1999.

[61] M. B. Waldron and S. Kim. Isolated asl sign recognition system for deaf persons. *Transactions on Rehabilitation Engineering.*, 3(3):261–271, 1995.

[62] S.-F. Wong and R. Cipolla. Real-time interpretation of hand motions using a sparse bayesian classifier on motion gradient orientation images. In *BMVC*, 2005.

[63] M.-H. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *Transactions on Pattern Analysis and Machine Intelligence.*, 24(8):1061–1074, 2002.

[64] M. Zahedi, P. Dreuw, D. Rybach, T. Deselaers, and H. Ney. Geometric features for improving continuous appearance-based sign language recognition. In *BMVC*, volume 3, pages 1019–1028, 2006.

[65] M. Zahedi, D. Keysers, T. Deselaers, and H. Ney. Combination of tangent distance and an image distortion model for appearance-based sign language recognition. In *Pattern Recognition*, pages 401–408. Springer, 2005.

[66] M. Zahedi, D. Keysers, and H. Ney. Appearance-based recognition of words in american sign language. In *Pattern recognition and image analysis*, pages 511–519. Springer, 2005.

[67] L.-G. Zhang, Y. Chen, G. Fang, X. Chen, and W. Gao. A vision-based sign language recognition system using tied-mixture density hmm. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 198–204. ACM, 2004.

[68] J. Zieren and K.-F. Kraiss. Robust person-independent visual sign language recognition. In *Pattern recognition and image analysis*, pages 520–528. Springer, 2005.