SENSOR-BASED SIGNER INDEPENDENT CONTINUOUS ARABIC SIGN LANGUAGE RECOGNITION

by

Mohamed Hassan

A Thesis Presented to the Faculty of the
American University of Sharjah
College of Engineering
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Mechatronics Engineering

Sharjah, United Arab Emirates

May 2017

Approval Signatures

We, the undersigned, approve the Master's Thesis of Mohamed Hassan.

Thesis Title: Sensor-Based Signer Independent Continuous Arabic Sign Language Recognition

Signature **Date of Signature** (dd/mm/yyyy)

_____ _____

Dr. Khaled Assaleh
Professor, Department of Electrical Engineering
Thesis Advisor

_____ _____

Dr. Tamer Shanableh
Professor, Department of Computer Science and Engineering
Thesis Co-Advisor

_____ _____

Dr. Tarik Ozkul
Professor, Department of Computer Science and Engineering
Thesis Committee Member

_____ _____

Dr. Mamoun Abdel-Hafez
Professor, Department of Mechanical Engineering
Thesis Committee Member

_____ _____

Dr. Lotfi Romdhane
Director, Mechatronics Engineering Graduate Program

_____ _____

Dr. Mohamed El-Tarhuni
Associate Dean for Graduate Affairs and Research
College of Engineering

_____ _____

Dr. Richard Schoephoerster
Dean, College of Engineering

_____ _____

Dr. Khaled Assaleh
Interim Vice Provost for Research and Graduate Studies

Acknowledgements

I would like to thank all the people who contributed to this work. First and foremost, I want to thank my academic advisors, Dr. Khaled Assaleh and Dr. Tamer Shanableh, for their continued help and support. I will be forever grateful to both of you.

I am sincerely thankful for Dr. Lotfi Romdhane, the director of the Mechatronics Engineering program, for his unlimited support. He was always there whenever I needed help. I am forever grateful to all the professors in the program. I am also thankful for the American University of Sharjah for sponsoring my M.Sc degree studies.

Last but not least, all appreciation goes to my family, friends, and colleagues for believing in me.

This thesis is dedicated to my beloved parents . . .

Abstract

The deaf community relies on sign language as the primary means of communication. For the millions of people around the world who suffer from hearing loss, interaction with hearing people is quite difficult. The main objective of Sign language recognition (SLR) is the development of automatic SLR systems to facilitate communication with the deaf community. SLR as a whole is considered a relatively new area. Arabic SLR (ArSLR) specifically did not receive much attention until recent years. This work presents a comprehensive comparison between two different recognition techniques for continuous ArSLR, namely a Modified k-Nearest Neighbor (MKNN) which is suitable for sequential data and Hidden Markov Models (HMMs) techniques based on two different toolkits. Additionally, in this thesis, two new ArSL datasets composed of 40 Arabic sentences are collected using Polhemus G4 motion tracker and a camera. An existing glove-based dataset is employed in this work as well. The three datasets are made publicly available to the research community. The advantages and disadvantages of each data acquisition approach and classification technique are discussed in this thesis. In the experimental results chapter, it has been shown that data acquisition using only the motion tracker results in accurate sentence recognition similar to that generated by the glove-based acquisition system. The modified KNN solution is inferior to HMMs in terms of the computational time required for classification. Moreover, the performance of Polhemus G^4 and RASR on multiple users is examined and promising results have been achieved.

Search Terms: ***Arabic Sign Language Recognition, pattern classification, feature extraction, Motion detectors.***

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ArSLR Arabic Sign Language Recognition

DCU Device Configuration Utility

GT^{2}K Georgia tech gesture toolkit

MKNN Modified K-nearest-neighbors

RASR RWTH Aachen University Open Source Speech Recognition Toolki

SL Sign Language

SLR Sign Language Recognition

WER Word Error Rate

Chapter 1: Introduction

Sign language recognition (SLR) is closely related to speech recognition (SR). Therefore, most of the analysis and classification techniques used in SLR have been borrowed from the speech recognition literature which has been established decades ago and has reached an adequate level of maturity. SLR, on the other hand, is still a new but active area of research. Since sign language is primarily a set of gestures, it is similarly affected by the advances in gesture recognition. Nonetheless, not all techniques of gesture and speech recognition are adequate for SLR. Over the years SLR has developed its own literature.

SL has its own set of grammar rules, which is different from the spoken language grammar. It is worth noting that SL is not universal. American Sign Language (ASL) is different from Arabic Sign Language (ArSL) which is different from Chinese Sign Language (CSL) [1]. Some SLs have more in common with spoken languages than others; Sign Exact English (SEE) is a good example, but it is not popular [2].

Compared to other gestures, SL is the most structured one. It has a large set of signs where each sign has a specific meaning. The majority of signs are associated with words while some are for finger spelling. For instance, American Sign Languages (ASL) has approximately 6000 signs [2].

Data availability is one of the main struggles facing researchers in SLR. The number of publicly available datasets is quite limited both in terms of quantity and quality. Manually annotated datasets are severely scarce. Moreover, since sign language is not universal, some sign languages (e.g., English and Chinese) have more datasets available than others (e.g., Arabic). Some publicly available datasets are in [3–5].

Another issue in SL is co-articulation or epenthesis which is also encountered in speech recognition. It means that a sign is affected by the signs before and after it. It is a well-known problem in speech recognition, but in SL it happens over a longer period and affects different aspect of the sign at the same time. This poses a lot of troubles in continuous recognition. Yang and Sarkar used conditional random fields (CRF) to detect co-articulation in SL [6]. Other approaches for handling co-articulation can be

found in [7, 8]. The way in which each person performs signs might be different; this is another problem known as signer dependency.

Signs of SL consist of two components: manual and non-manual features. Manual features are hand position, orientation, shape and trajectory. Non-manual features are body movement and facial expressions. Most of the information is conveyed through manual features [9], thus most of the researches focused only on them [10]. Non-manual features can form a sign by themselves, but mostly they can modify the meaning of the manual features; raising eyebrow indicates a question. Lip shape and head pose are popular non-manual features as well.

The two main approaches for SLR are: vision-based and sensor-based approaches. Vision-based SLR uses cameras only to capture gestures (signs). It has the advantage of user friendliness since the user is not required to wear any devices such as data gloves or motion trackers. However, the computational cost is normally high for this approach. Moreover, it can be quite sensitive to variations in the background or changing illumination conditions.

Sensor-based approaches, however, make use of wearable devices to capture the signs in a more accurate manner than vision-based systems. Although it might not be as convenient as the vision-based systems, it comes with a huge improvement in recognition accuracy. Additionally, its data acquisition and processing requirements are modest. Gloves and motion trackers are the most commonly used sensors for sensor-based SLR. Usually the measurements of those sensors are accurate enough; thus no feature extraction techniques are needed. They allow researchers to focus on the recognition problem. The drawback of using sensors is that the signer is required to wear those sensors, which can be annoying. Another issue is the cost; accurate gloves and motion trackers are still expensive. Before selecting a glove, the reader is encouraged to look at [11], where a comparison between the different gloves available in the market and the technologies behind them is presented. An older survey was done by Sturman and Zeltzer in [12] where they discussed some of the common applications of gloves in different fields. A summary of the comparison between the vision-based and sensor-based approaches is shown in Table 1.

Table 1: A Comparison between the vision-based and sensor-based approaches.

	Vision-based approach	Sensor-based approach
Tools	**Camera**	**Sensors (wearable devices)**
Feature extraction	**Essential**	**Not Essential**
Computational power	**High**	**Low**
Recognition rate	**Low**	**High**
Cost	**Low**	**High**
Manual features capturing	✓	✓
Non – manual features capturing	✓	✗

In general, any glove contains a number of sensors to measure the bending at each finger. Different technologies are used to develop those sensors: piezoresistive, Hall Effect, and fiber optics are examples. The accuracy of those sensors and their distribution over the hand is what specifies the overall accuracy. Some gloves need calibration for each user; this factor must be in mind before buying a glove.

Some gloves come with an embedded accelerometer, gyroscope, and magnetometer. Hence it can capture both the hand shape using the bend sensors as well as hand position and orientation. DG5-VHand is an example of such gloves [13]. Other gloves contain only bend sensors thus additional sensors are required to capture hand motion and orientation; an example of such is CyberGlove [14]. In such cases motion trackers could be used alongside the gloves to record hand trajectory. The reader is referred to [15] for an overview of the available motion trackers, their technologies and applications.

The rest of this thesis is organized as follows. Section 1.1 presents a short survey of the current state-of-the-art in SLR. A description of the datasets used and their collection procedures is in Chapter 2, followed in Chapter 3 by a detailed explanation of the feature extraction techniques used and the adopted classification algorithms. Results of the experiments are presented in Chapter 4. Concluding remarks and future works are discussed in Chapter 5.

1.1. Literature Review

SLR can be at the level of alphabets, isolated words, or continuous words. The data collection can be a vision-based approach or a sensor-based one. This section tries to provide an overview of the state-of-the-art techniques in SLR on the three aforementioned levels using both vision-based and sensor-based approaches.

A recent and thorough survey of SLR was done by Cooper *et al.* [9]. They covered the key components of SLR, and discussed the pros and cons of the different types of data available. The manual and non-manual components of signs were also explored, as well as the recent researches in the area. The survey discussed some of the current research frontiers such as: continuous recognition, signer independency, the work towards combining different modalities of sign, and the development of unconstrained real-life SLR systems. In [16], a more recent survey with a focus on Indian sign language (ISL) is introduced.

Recognition of alphabets is in general easier than recognizing words. Usually alphabets are static gestures, this allows the use of conventional classification and clustering techniques. Color gloves were used to collect data of ArSL alphabets from multiple users, where adaptive Neuro-Fuzzy Inference System (ANFIS) was the recognition approach [17] . The same data and feature extraction techniques were used by Assaleh *et al.* [18], but they used polynomial classier and reported better results than the previous ANFIS approach. In [19], depth camera was used as the input device for real time recognition of ASL alphabets.

The problem of co-articulation is not present in recognition of isolated gestures, which makes it simpler than continuous sign recognition but tougher than alphabets recognition due to its dynamic nature. Oz *et al.* [20] collected a dataset of 50 isolated right handed words of ASL. After extracting some global features, artificial neural networks were used for classification. The system was tested on multiple users as well as on new words, and they reported accuracy of 90%. Different spatio-temporal feature-extraction techniques were used in [21] for recognizing isolated ArSL words. Accuracy of 97% was reported upon using K nearest neighbor (KNN) classifier. Their proposed feature extraction and classification yielded results comparable to conventional HMM.

HMMs are the most commonly used classification technique in SLR. For instance, Gaussian Hidden Markov Model (GHMM) was used on the SIGNUM database [22]. Multilayer perceptron (MLP) features in addition to appearance-based features extracted directly from the videos were used achieving word error rate (WER) of 11.9%. Then Principal Component Analysis (PCA) was used for dimensionality reduction. The same group investigated combining different sign modalities for the same database [23]. They studied different combinations of five modalities, and were able to decrease WER to 10.7%. A promising approach of end-to-end embedding of a Convolutional Neural Network (CNN) into an HMM was recently proposed in [24]. In [25], HMMs were used to model the hand trajectory for large isolated Chinese SLR. They claim to achieve better performance compared to normal coordinate features with HMM.

In [26], Kong and Ranganath presented promising results in terms of signer independency. Their system was tested on new signs as well as new users. Accuracies of 95.7% and 86.6% respectively were reported. They used a segmentation algorithm proposed in their previous work [27]. A major contribution in signer independency was done by Koller and colleagues [28], where they worked on two publicly available large vocabulary databases representing lab-data (SIGNUM database:25 signers, 455 sign vocabulary, 19k sentences) and unconstrained real-life sign language (RWTH-PHOENIX-Weather database: 9 signers, 1081 sign vocabulary, 7k sentences). The earlier works of Gao *et al.* [29] and Fang *et al.* [30] are also examples of research on signer independency and large vocabulary. Fang *et al.* tried to tackle co-articulation by modeling the transition between signs using transition-movement models (TMMs).

In vision-based SLR, hand tracking is still a challenge especially in unconstrained environment where the background is cluttered and illumination conditions vary. Several researches have tried to tackle this issue with the use of Kinect [31–34]. Kinect simplifies hand tracking by providing depth and color data simultaneously. In [35], a survey of the state-of-the-art of gesture recognition as it was in 2007 was presented. Object tracking methods and techniques were surveyed by Yilmaz *et al.* [36]. A detailed survey of vision-based systems for capturing the human motion was done by Moeslund and Granum [37]. Non-manual features like facial expressions received a lot of attention in recent years in computer vision as well. In [38], a survey of algorithms

for face detection in images was presented. A comprehensive study of face recognition was presented in [39, 40]. Analysis of facial expression recognition is in [41]. Head pose estimation was studied in [42]. Recently gaze estimation was studied in [43]. Many depth cameras are now available in the market with a variety of prices and accuracies. ZED and Kinect are the most commonly used ones. A team from Microsoft research Asia has developed Kinect-based SLR system and has reported very promising results [32]. Zafrulla *et al.* compared their old copycat system which used a colored glove and embedded accelerometer to a system based in Kinect in [33].

Gloves and motion trackers were used as input devices in many researches. CyberGloves and Flock of Birds 3-D motion tracker were used to collect data of 50 isolated right handed words of ASL in [20]. Gao *et al.* [29] used two CybeGloves and 3 Pohelmus trackers. A similar approach using CyberGlove and Polhemus FASTRACK system was used in [26].

Until recently Arabic Sign Language Recognition (ArSLR) has not received much attention. A survey of the contributions in ArSLR up to 2014 using both sensor-based and vision-based approaches can be found in [44]. The majority of the literature is concerned with isolated sign recognition. For instance, a system based on adaptive neuro fuzzy inference system (ANFIS) networks was proposed by Al-Jarrah and Halawani to recognize 30 Arabic alphabets. They managed to achieve an accuracy of 93.55% [45]. A vision-based posture recognition called AndroSpell was proposed in [46] where the authors made use of a camera phone to recognize 10 postures with 97% accuracy.

In [47], Shanableh and Assaleh proposed a vision-based user-independent system capable of recognizing 23 Arabic sign language words with average classification rate of 87.0%. Their dataset was collected by 3 signers, where each one was asked to repeat each sign 50 times over 3 different sessions. They end up with 150 repetitions for all gestures. The signers wore colored gloves during the data collection process. Mohandes *et al.* [48] used Power Glove to collect a dataset of 120 words; they used some statistical features and Support Vector Machine (SVM) as a classifier. CyberGloves and two hand-tracking devices were used in [49] to build a similar system, so the accuracy jumped to 99.6%.

It was not until 2010 that the first continuous ArSLR system was proposed by Assaleh *et al.* [50]. They used their novel spatio-temporal feature-extraction technique [21] and reported 6.0% WER on a dataset of 40 sentences. A modified version of K-Nearest Neighbor (MKNN) was proposed by Tubaiz *et al.* in [51] and tested on the same dataset but collected using DG5-VHand data gloves instead of the camera. Their system achieved 2.0% WER. The data collected in [51] was used by Tuffaha *et al.* [52] where modified polynomial classifier was proposed. Similar feature extraction phase was implemented. In addition to mean and standard deviation, covariance, entropy and uniformity were appended to raw data. The paper reported 85.0% sentence recognition rate.

Multichannel Electromyography (EMG) sensors have been also used for SLR. A recent study fused accelerometer data with EMG sensors measurements, and used the intensity of EMG signal to automatically detect signs boundaries [53], see similar work in [54–58]. Vlasic *et al.* introduced a wearable motion capture device based on ultrasonic and inertial measurement and they reported good performance in the everyday environment [59].

Leap motion controller (LMC) has also gained some publicity lately. It depends on IR cameras and LEDs to capture the motion. It observes a distance of a meter in a hemispherical shape. The controller is connected to PC through USB. Data is analyzed by a software provided by the company. Potter *et al.* tested LMC on Australian Sign Language (Auslan) recognition [60]. In [61] features extracted from Kinect and LMC were combined for better recognition. Mohandes used LMC for recognizing Arabic Signs alphabets using Multilayer Perceptron (MLP) neural networks with the Naive Bayes classifier [62].

1.2. CyberGlove

As defined in [11], a glove-based system is a system worn on the hand, equipped with an array of sensors and the necessary electronics for data acquisition and processing. The goal of using a glove is to capture hand shape and configuration. Gloves available in the market differ in terms of the technology used, number of sensors, sensors

distribution, sensors precision and communication medium. Figure 1 shows the different joints in a human hand. Gloves try to capture the amount of pending in as many as possible of these joints. In 1992, Virtual Technologies Inc started selling CyberGlove

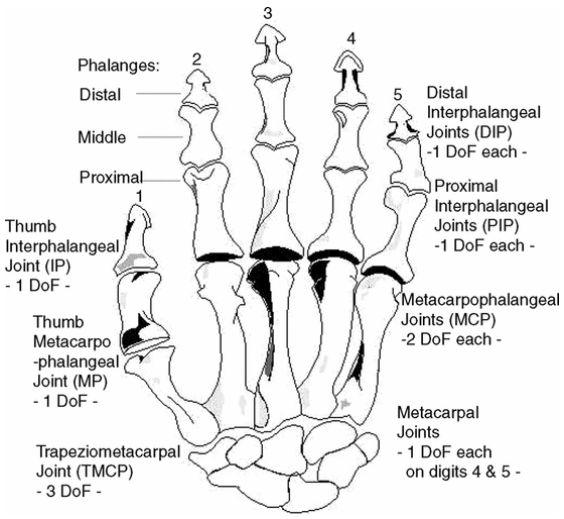

Figure 1: Joints in a human hand [11].

which was designed originally by James Kramer from Stanford. CyberGlove is considered one of most accurate gloves available today [11]. It comes in two versions; one with 18 sensors and another with 22 sensors. Both versions use piezoresistive sensors. In the 18 sensors version, there are two bend sensors on each finger plus four abduction sensors, in addition to sensors measuring the thumb crossover, palm arch, wrist flexion, and wrist abduction/adduction. Additional four sensors to measure DIP joints flexion are added in the 22 sensor version. There are also wired and wireless gloves. Wireless gloves provide more freedom to the user and they can work within a range of 100 feet. The gloves come with user-friendly Device Configuration utility (DCU) shown in Figure 2 through which hand shape can be constantly monitored. It also shows the readings of each sensor as shown in Figure 3. These gloves also come with a set of APIs and well-documented programmer guide.

Calibration is required for each user of the glove. Calibration compensates the variations in hand formation, thickness and fingers lengths. It also provides joint angles by converting the sensors' voltages. DCU is used for calibration where the user is asked to perform specific hand configurations and then the program automatically

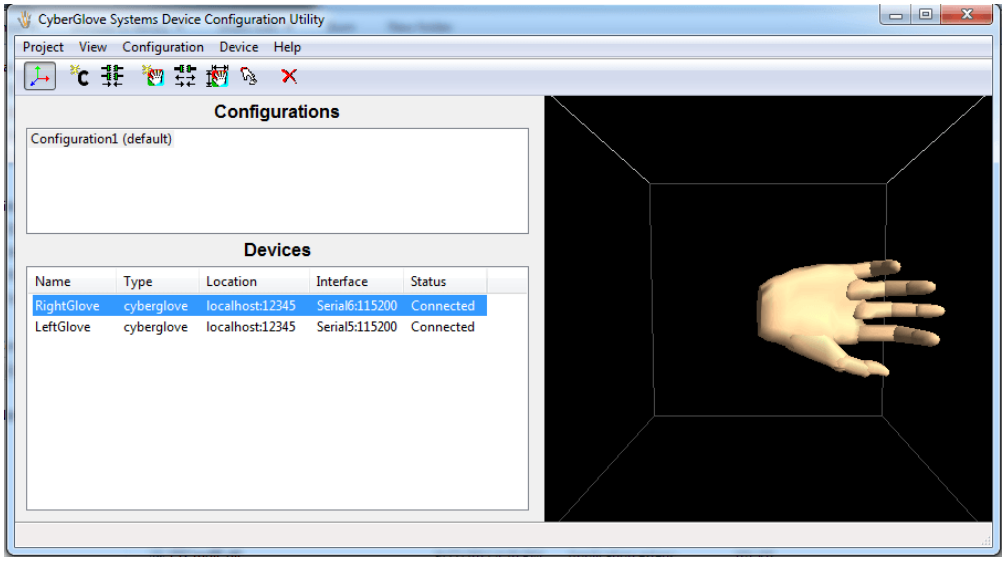

Figure 2: CyberGlove Device Configuration Utility

adjusts gains and offsets for each sensor. Better results can be achieved by manually adjusting the gains and offsets. A motion tracker can be added to the system by mounting the tracker sensors on top of the gloves. The DCU can be used to monitor the tracker data as well.

1.3. Polhemus G^4

Polhemus G^4 is 6 DOF wireless tracking system. The system contains the following components:

- Sensors which detect the electromagnetic field generated by the source. The position and orientation are measured from the center of the sensors.
- Source, which is a 4-inch cube, generates the necessary electromagnetic (EM) field for tracking the sensors. The closer the sensors are to the source, the more accurate the measurements are. Maximum motion tracking area is around 15 feet, but it can be extended by using more than one source with each one working in a different frequency.
- Hub, the sensors are connected to lightweight hub worn by the user which in turn sends the data to the PC via RF module. Wired operation is possible as well.
- RF communication is provided using RF/USB module connected to PC. The frame rate is 120 samples per second.

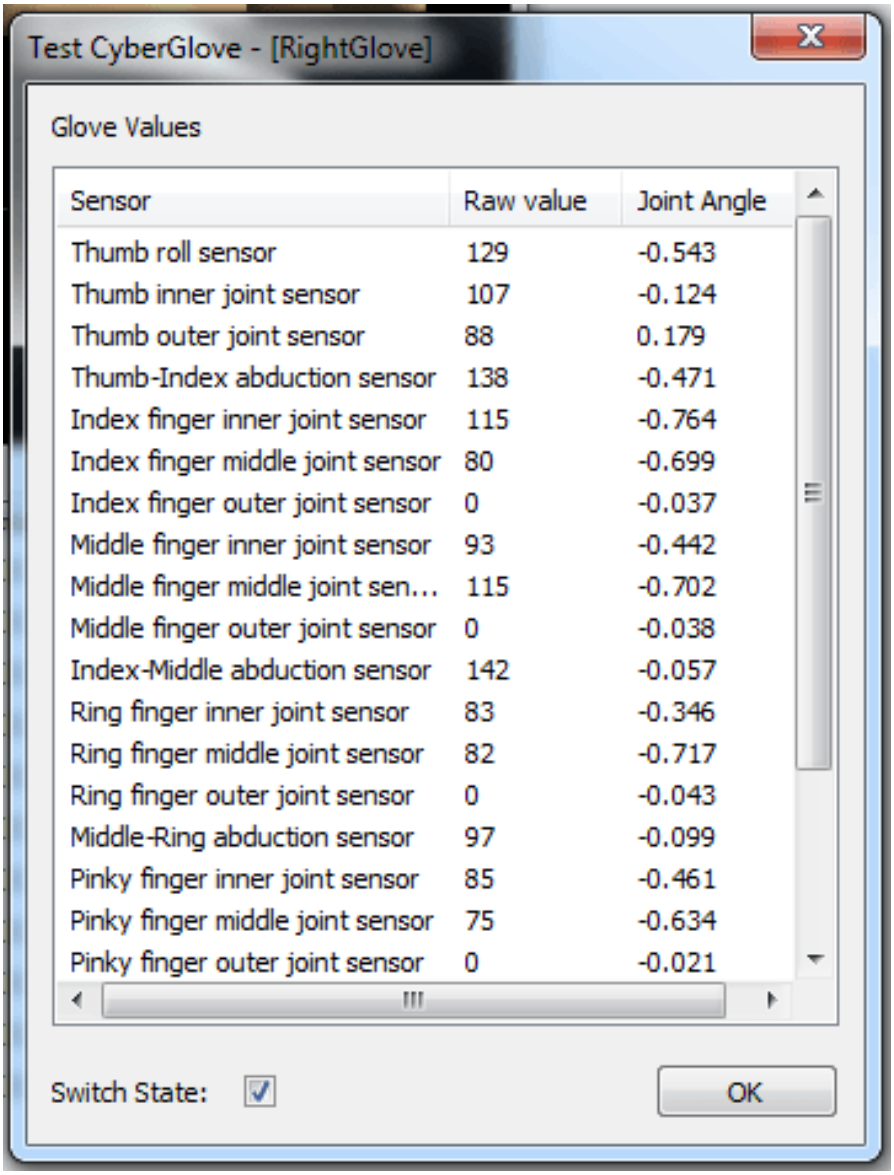

Figure 3: Glove measurements

The origin with respect to which the position and orientation will be measured can be the center of one of the sources or any actual location. Components are shown in Figure 4 and Figure 5.

For a better performance, it is recommended to operate within a range 2 to 6 feet (0.6 to 1.83m) from the source. Special attention should be paid to placing the source, as it should not be placed on a metal or close to magnetic distorter. The system comes with a handy software for monitoring and recording the sensors. More details are available in [63].

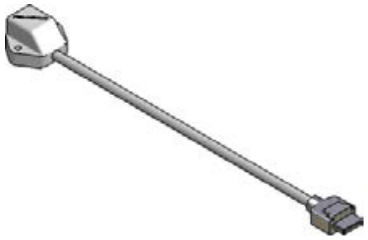

Figure 4: G4 Sensor and source

Figure 5: G4 RF USB and Hub

1.4. DG5-VHand

DG5-VHand has five embedded bend sensors - one on each finger- and one 3-axes accelerometer as shown in Figure 6. Bend sensors, which are also known as flex sensors, measure the amount of bend on the sensor and express it as electrical resistance. As the amount of bending increases, the resistance increases. DG5-VHand also has the advantage of the Bluetooth interface which makes the movement less restricted. A summary of its specifications is in Table 2.

Table 2: Specifications of the DG5-VHand Glove 2.0.

Number of Sensors	5 proprietary flex sensors 3 degrees of integrated tracking
Resolution	10 bits
Output Interface	Bluetooth Interface
Software Bundle	C++ SDK
Price (per hand)	$750.0

Figure 6: DG5-VHand Glove 2.0 [64].

Table 3: A Comparison between the CyberGlove and the DG5-VHand.

	CyberGlove	DG5 V-Hand
Number of sensors	**18 - 22**	**5**
Precision (bits)	**8**	**10**
Motion capturing	✘	✓
Technology	**Piezo-resistive**	**Fiper optics**
Calibration	✓	✓

The five flexion measurements for thumb, index, middle, ring, and little fingers are expressed as bend percentages in the range (0.0% - 100.0%) where 0.0% represents no bend and 100.0% represents maximum bend. The 3-axes accelerometer measurements represent the hands instantaneous accelerations in x,y, and z direction. Those measurements are between $-2g$ and $+2g$.

Figure 7 shows the DG5-VHand software used to establish communication with the glove. The data can be visually examined through the software and it can be saved in Microsoft Excel Worksheet (.xls) for further processing. Each row of the file starts with a time stamp in milliseconds, followed by five bend measurements, then three acceleration measurements. A Summary of the comparison between the CyberGlove and the DG5-VHand is shown in Table 3.

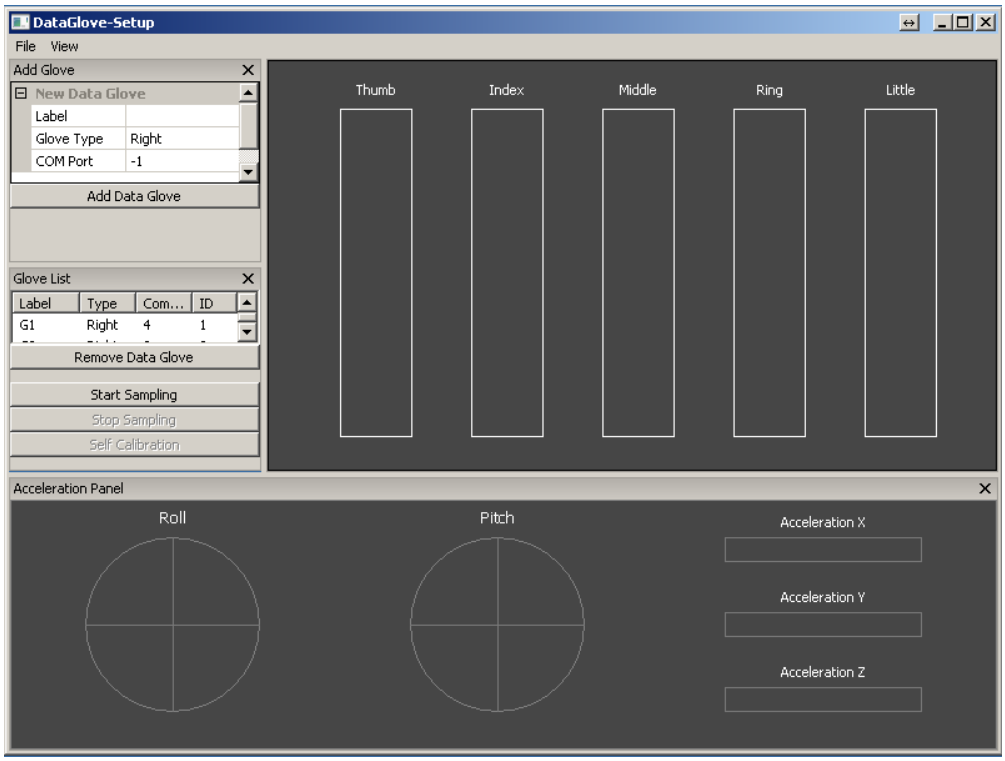

Figure 7: DG5-VHand software.

Chapter 2: Data Collection

Two datasets are collected in this work; both of which are composed of 40 Arabic sign language sentences created from 80 words lexicon. Each sentence was repeated 10 times by each user. The list of sentences is shown in Table4. An existing dataset which was collected using DG5-VHand data gloves [51] was also used, and is referred to henceforth as dataset 1. The DG5-VHand data glove comes with five bend sensors; one for each finger. It also has an embedded accelerometer. Figure 8 shows a female signer wearing the DG5-VHand data gloves during the collection of Dataset 1. Dataset 2 is collected in this work using two Polhemus G4 motion trackers which provide 6 measurements; the Cartesian position coordinates (x,y,z) and the Euler angles: Azimuth, elevation and roll(a, e, r). Two male users contributed in collecting this dataset where each one was asked to perform each sentence 10 times. Figure 9 and Figure 10 show photos of the two male signers wearing the CyberGloves and Polhemus G4 motion trackers during the collection of Dataset 2. Dataset 3 is also collected in this work using a camera only; no wearable sensors is used during the collection of this dataset. Table 5 summarizes the specification of the three datasets. The first two datasets are expected to result in higher recognition accuracy due to the use of accurate sensors, the third dataset has the advantage of being more user-friendly since the user is not required to wear any device.

Both of our classification approaches; namely HMMs and MKNN are supervised classification algorithms, thus labeling is required. We need to assign a label to each feature vector; this label will specify to which class (sign) this feature vector belongs. Normally a sentence contains multiple signs; hence it is necessary to define the boundaries between successive signs prior to labeling. This is a difficult task, because it is not clear where each sign starts and ends. It is also difficult to specify which parts of the temporal data are not parts of a sign. It is easier to define boundaries in Automatic Speech Recognition (ASR) by a decline in speech volume. However, it is more involved in SLR. Consider, for example, the case of sign 'X' which ends with the right hand raised to the chin level followed by sign 'Y' which starts with the right hand at the waist level. The movement from the end of sign 'X' to the beginning of sign 'Y' is not

Table 4: List of sentences.

Arabic Sentence with English Meaning	Hands Used	Arabic Sentence with English Meaning	Hands Used
ذهبت الى نادي كرة القدم I went to the soccer club	Both	عندي أخوين I have two brothers	Right
انا احب سباق السيارات I love car racing	Both	ما اسم ابيك؟ What is your father's name?	Right
اشتريت كرة ثمينة I bought an expensive ball	Both	كان جدي مريضا في الامس Yesterday my grandfather was sick	Right
يوم السبت عندي مباراة كرة قدم On Saturday I have a soccer match	Both	مات ابي في الامس Yesterday my father died	Right
في النادي ملعب كرة قدم There is a soccer field in the club	Both	رأيت بنتا جميلة I saw a beautiful girl	Right
غدا سيكون هناك سباق دراجات There will be a bike race tomorrow	Both	صديقي طويل My friend is tall	Both
وجدت كرة جديدة في الملعب I found a new ball in the field	Both	انا لا اكل قبل النوم I do not eat close to bedtime	Both
كم عمر اخيك؟ How old is your brother?	Both	اكلت طعاما لذيذا في المطعم I ate delicious food at the restaurant	Both
اليوم ولدت امي بنتا My mom had a baby girl today	Right	انا احب شرب الماء I like drinking water	Both
اخي لا يزال رضيعا My brother is still on breastfeeding	Both	انا احب شرب الحليب في المساء I like drinking milk in the evening	Both
ان جدي في بيتنا My grandfather is at our home	Both	انا احب اكل اللحم اكثر من الدجاج I like eating meat more than chicken	Both
اشترى ابني كرة رخيصة My kid bought an inexpensive ball	Both	اكلت جبنة مع عصير I ate cheese and drank juice	Both
قرأت اختي كتابا My sister read a book	Both	يوم الأحد القادم سيرتفع سعر الحليب Next Sunday the price of milk will go up	Both
ذهبت امي الى السوق في الصباح My mother went to the market this morning	Both	أكلت زيتونا صباح الامس Yesterday morning I ate olives	Right
هل اخوك في البيت؟ Is your brother home?	Both	سأشتري سيارة جديدة بعد شهر I will buy a new car in a month	Both
بيت عمي كبير My brother's house is big	Both	هو توضأ ليصلي الصبح He washed for morning prayer	Both
سيتزوج اخي بعد شهر In one month my brother will get married	Both	ذهبت الى صلاة الجمعة عند الساعة العاشرة I went to Friday prayer at 10:00 o'clock	Both
سيطلق اخي بعد شهرين In two months my brother will get divorced	Both	شاهدت بيتا كبيرا بالتلفاز I saw a big house on TV	Both
اين يعمل صديقك؟ Where does your friend work?	Both	في الامس نمت عند الساعة العاشرة Yesterday I went to sleep at 10:00 o'clock	Both
اخي يلعب كرة سلة My brother plays basketball	Both	ذهبت الى العمل في الصباح بسيارتي I went to work this morning in my car	Both

Table 5: Datasets and equipment used.

Datasets	Equipment	No. of Users	No. of Sentences	No. of Repetitions	Total No. of Sentences
Dataset 1	DG5-VHand data gloves	1	40	10	400
Dataset 2	Polhemus G4 motion trackers	2	40	10	800
Dataset 3	Camera	1	40	10	400

Figure 8: Collection setup of Dataset 1 [51].

Figure 9: Collection setup of Dataset 2.

a part of either sign. This transitional movement is known as movement epenthesis. In general, if it spans few feature vectors, it can be compensated for by HMMs, otherwise it should be labeled separately and should not be part of the learned model of any sign.

Several works chose to model movement epenthesis explicitly [65]. For instance, Gao *et al.* [66] used data gloves to find the end and start points of each sign in their database. By clustering them into three general clusters using the temporal

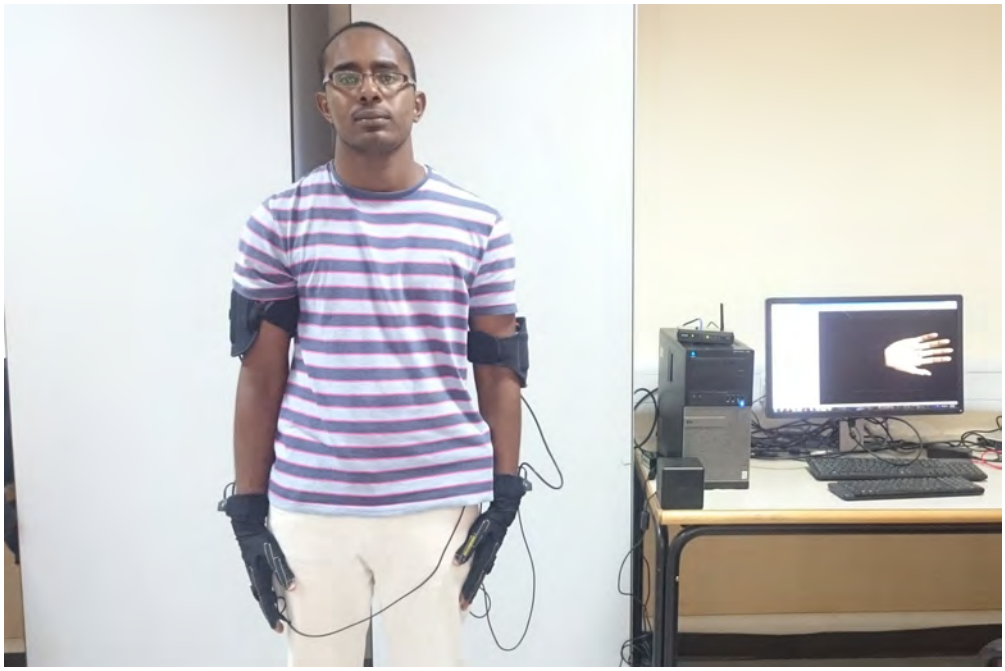

Figure 10: Collection setup of Dataset 2.

clustering algorithm Dynamic Time Warping (DTW), they managed to achieve 90.8% recognition accuracy on 750 test sentences. A different approach was proposed by Yang *et al.* in [67]. They proposed an adaptive threshold model to differentiate between signs and movement epenthesis.

In the labeling phase, all the sensor readings belonging to each word in a sentence are labeled accordingly. For manual labeling in vision-based SLR, a human can decide the boundaries by examining the videos visually. However, for sensor-based SLR, the word boundaries cannot be determined visually. Therefore, a camera was used in the data collection phase and it was synchronized with the gloves and tracker recordings in order to detect word boundaries.

Chapter 3: Theory

3.1. Feature Extraction

Minimal feature extraction is required for sensor-based data because it contains direct measurements of hand trajectory and configuration. On the other hand, vision-based data requires extensive feature extraction to get such features. Below is a discussion of the feature extraction techniques used for the sensor-based datasets (dataset 1 and dataset 2) and the vision-based dataset (dataset 3).

3.1.1. Feature extraction for sensor-based datasets. As mentioned before, dataset 1 was collected using DG5-VHand Glove. The DG5-VHand provides five flexion measurements in the range (0.0% - 100.0%) followed by 3- axes accelerometer measurements in the range of $-2g$ and $+2g$. Prior to applying any classification techniques, it is recommended to normalize all features. Normalization allows different features to contribute equally to the classification decision. Normalization is required for dataset 2 as well, since the CyberGlove measurements and the Polhemus G^4 measurements have different ranges. There are different types of normalization techniques available in the literature, Z-score normalization is probably the most common one. The Z-score could be formulated as in (1).

$$z = \frac{x-u}{\sigma} \tag{1}$$

where:

- x is the raw feature.
- z is the normalized feature.
- σ is the standard deviation of x.
- u is the mean of x.

It could be easily shown that the normalized feature z has a mean $u = 0$ and a standard deviation $\sigma = 1$. An example of applying the Z-score on two different features is shown in Figure 11. We can see how close the two features have become after the normalization.

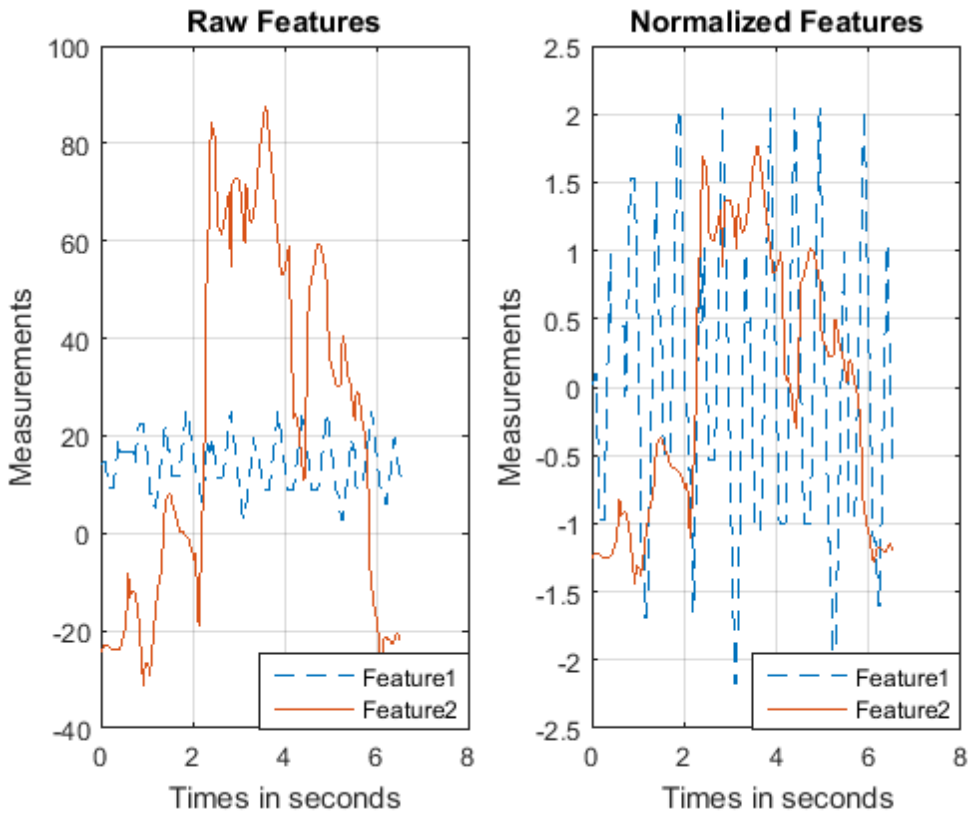

Figure 11: Z-score Normalization.

After normalization, some meaningful features are extracted out of the data. Our approach is to extract window-based statistical features because they provide information about the context, which in turn helps in classifying our temporal data. Those features are then appended to the raw feature vectors to form the final augmented feature vectors. The size of the window is varied via trial and error to achieve the highest performance. The classification systems were tested on raw data as well as on raw data augmented with the extracted statistical features.

Statistical features used included mean, standard deviation, covariance, entropy and uniformity. All features are extracted using the sliding window approach. The purpose of using a sliding window is to capture contextual information. In Chapter 4 we show that this greatly enhanced the classification accuracy. Equation (2) shows the calculation of the window-based mean for a window size of w.

$$\tilde{x}_i = \frac{1}{w} \sum_{k=i-\frac{w-1}{2}}^{i+\frac{w-1}{2}} x_k \tag{2}$$

where:

- w is an odd number denoting the window size.
- i is the current feature vector.

Standard deviation measures the dispersion from the average sample. Equation (3) shows the calculation of the window-based standard deviation. In fact, Equation (3) calculates what is better known as sample standard deviation since only part of the population is considered.

$$s_i = \left(\frac{1}{w-1} \sum_{k=i-\frac{w-1}{2}}^{i+\frac{w-1}{2}} (x_k - \tilde{x}_i)^2 \right)^{\frac{1}{2}} \tag{3}$$

To further enrich the feature vector, we add the covariance. Covariance is a measure of the strength of the correlation between two random variables. For two real-valued random variables x and y, the covariance is defined as the expectation of the product of their deviations from their mean values as shown in (4).

$$cov(x,y) = E((x-E(x))(y-E(y))) \tag{4}$$

The covariance matrix of two random vectors $X \in \mathbb{R}^n$ $Y \in \mathbb{R}^m$ is defined by (5).

$$cov(X,Y) = E((X-E(X))(Y-E(Y))^T) \tag{5}$$

Our window-based covariance can be then formulated as in (6)

$$cov_i(X,Y) = \frac{1}{w-1} \sum_{k=i-\frac{w-1}{2}}^{i+\frac{w-1}{2}} (x_i - E(X))(y_i - E(Y)) \tag{6}$$

Entropy (H) is a measure of the amount of information. It was introduced first by the father of information theory, Claude Shannon. He defined it by (7).

$$H = -\sum_{i=1}^{n} P(x_i) log_2(P(x_i)) \tag{7}$$

Equation (8) shows the window-based entropy.

$$H_i = -\sum_{k=i-\frac{w-1}{2}}^{i+\frac{w-1}{2}} P(x_k) log_2(P(x_k)) \tag{8}$$

Uniformity is another feature that has been extracted and it describes how uniform the data is, as shown in (9). Window-based uniformity is shown in (10).

$$Uniformity = \sum_{i=1}^{n} P(x_i) log_2(P(x_i)) \tag{9}$$

$$Uniformity_i = \sum_{k=i-\frac{w-1}{2}}^{i+\frac{w-1}{2}} log_2(P(x_k)) \tag{10}$$

Appending all those features to the raw sensor data greatly increased the dimensionality of the feature vector. For example, the DG5-VHand raw feature vector consists of 16 features, the augmented feature vector consists of 200 features:

- 16 original sensor readings.
- 16 means of a window of FVs.
- 16 sample standard deviations of a window of FVs.
- 120 covariance of a window of FVs.
- 16 entropy of a window of FVs.
- 16 uniformity of a window of FVs.

3.1.2. Feature extraction for vision-based dataset. Vision-based datasets require extensive feature extraction to get meaningful features. Numerous techniques have been used in the literature and it can be a challenge to select the appropriate features. The proposed feature extraction approach is explained by the block diagram in Figure 12.

The first stage of the pipeline is computing pixel-based difference for successive images (frames). Assuming typical frame rate, the two images will have almost identical background, thus the difference will be an image that only depicts the motion between the two images. The image differences are then converted into binary images

Figure 12: Feature extraction block diagram for vision-based dataset.

by applying an appropriate threshold. The threshold is given by (11).

$$\mathrm{TH} = \mu + x\sigma \tag{11}$$

where:

- μ is the mean pixel intensity of the image difference.
- σ is the corresponding standard deviation.
- x is a weighting parameter

x is to be empirically determined based on subjective evaluation whose criterion is to retain enough motion information and discard noisy data. Figure 13 shows an illustration of performing image difference followed by thresholding.

Next, a 2D Discrete Cosine Transform (DCT) is applied to the binary image differences. The 2-D DCT is given by (12).

$$F(u,v) = \frac{2}{\sqrt{MN}} C(u)C(v) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) cos\left(\frac{\pi u}{2M}.(2i+1)\right) cos\left(\frac{\pi v}{2N}.(2j+1)\right) \tag{12}$$

where:

- $F(u,v)$ is DCT coefficient at row u and column v of the DCT matrix.
- N, M are the dimensions of the input image f.
- $C(u)$ is a normalization factor equal to $\frac{1}{\sqrt{2}}$ for $u = 0$ and 1 otherwise.

Figure 14 shows a 2D DCT of a thresholded image difference. DCT is quite popular in signal and image processing due to its energy compaction property. In Figure 14, most of the information is represented by the top left coefficients. Thus a zig-zag scanning is used to select the most important coefficients only. Figure 15 explains the zig-zag scanning fashion.

The top left DCT coefficients are zigzag scanned to form a 1D feature vector. The number of DCT coefficients in the vector is known as the DCT cutoff value.

Figure 13: Thresholded image differences [50].

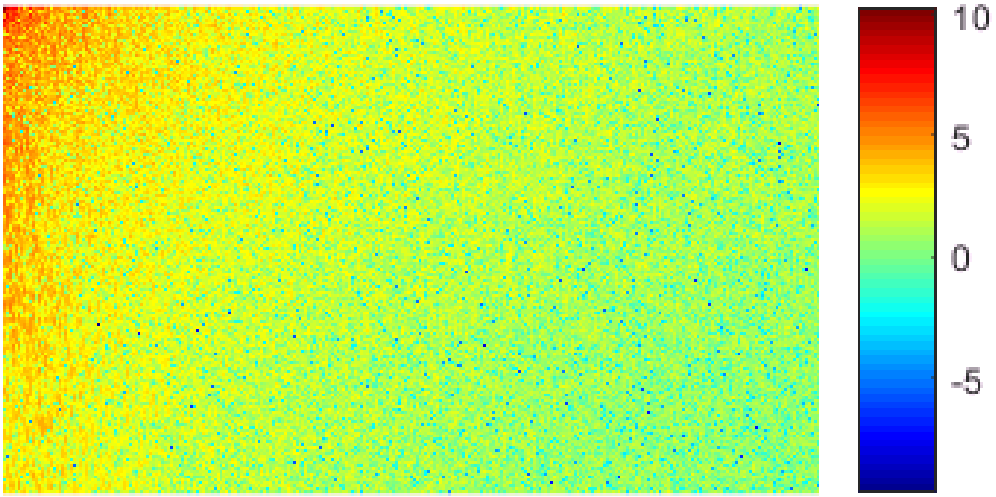

Figure 14: 2D Discrete Cosine Transform (DCT).

3.2. Classification

Three different classification approaches are used in this work; a modified KNN suitable for classifying sequential data, and two different HMM toolkits. A review of each is presented next.

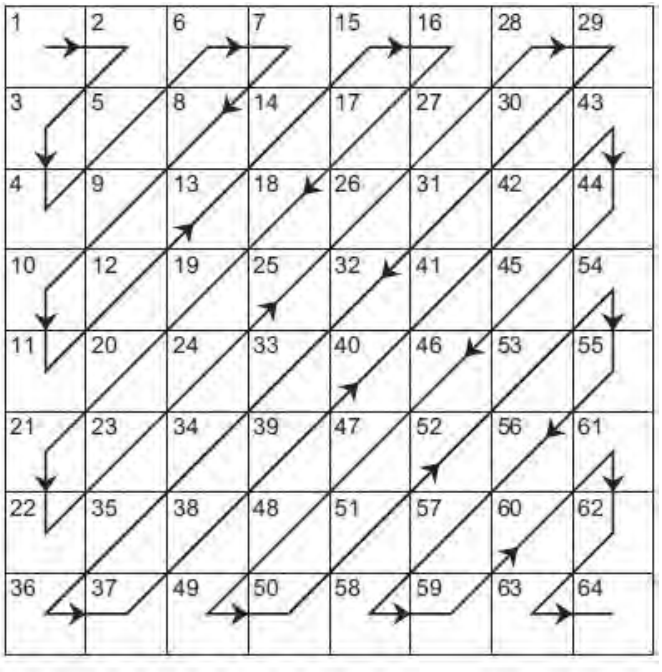

Figure 15: zig-zag scanning [68].

3.2.1. K-nearest-neighbors (KNN). K-nearest-neighbors (kNN) is one of the simplest classification techniques that should always be one of the first choices to consider especially when little or nothing is known about the data. KNN is a non-parametric classification technique since it makes no assumption about the distribution. Non-parametric methods have the ability to fit different types of distribution and they are easier to implement. But they normally require a lot more training data than parametric methods.

KNN depends on the Euclidean distance between a test sample and the training samples. For instance, consider x^i to be a test sample with p features $(x_1^i, x_2^i, x_3^i, ..., x_p^i)$ and we have a training set of N samples $(x^1, x^2, x^3, ..., x^N)$. Y is the set of labels for all training examples $(y^1, y^2, y^3, ..., y^N)$. The Euclidean distance between the test sample x^i and a random training sample x^j is given by (13).

$$d(x^i, x^j) = \sqrt{(x_1^i - x_1^j)^2 + (x_2^i - x_2^j)^2 + ... + (x_p^i - x_p^j)^2} \tag{13}$$

Consider the case of two classes problem $(y^j = 0)$ *or* $(y^j = 1)$ $\forall$ j. and two-dimensional data $p = 2$. KNN partition the space into Voroni cells with each cell labeled based on the training point it contains as shown in Figure 16.

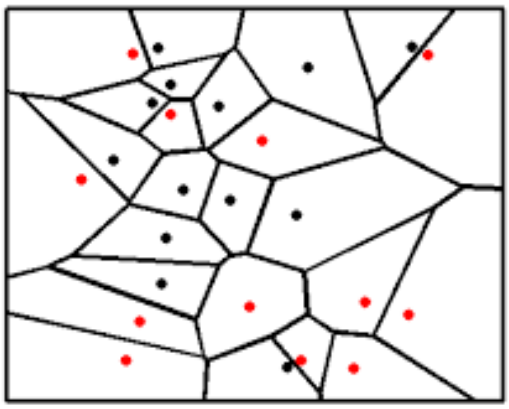

Figure 16: KNN Voronoi diagram, from [69].

The diagram shows multiple training samples from the two classes (red or black circles) and a Voroni cell R around each sample. The cell encapsulates all points that are closer to this training sample than they are to any other training sample in the training set. R is defined by (14).

$$R_i = \{\mathbf{x} \in \mathbb{R}^p : d(\mathbf{x}, \mathbf{x}_i) \leq d(\mathbf{x}, \mathbf{x}_m), \quad \forall \quad i \neq m\} \tag{14}$$

There is no actual training in KNN; it is only required to store the p features of each training sample $(x_1^j, x_2^j, x_3^j, ..., x_p^j)$ and its label and y^j. For classification, an unlabeled test sample x^i is classified by assigning it with the most frequent label among the nearest k training samples, where k is a constant defined by the user. An illustration of the KNN is in Figure 17. The example is again of a two classes problem; the labels are either a green square or a blue rectangle. The data is of a dimension of two $p = 2$ and the user-defined constant k is 3. In classification, the unlabeled circle will be labeled as a green square because there are two green squares and only one blue rectangle within the nearest 3 neighbors.

3.2.1.1. Modified K-nearest-neighbors (MKNN). In previous work, a modification on the K-Nearest Neighbors (KNN) classifier was proposed to make it suitable for classifying sequential data [51]. The modified algorithm was called the Modified KNN or MKNN for short. The core modification is to consider the context prior to predicting the label of each feature vector. The approach used in this thesis was to replace the

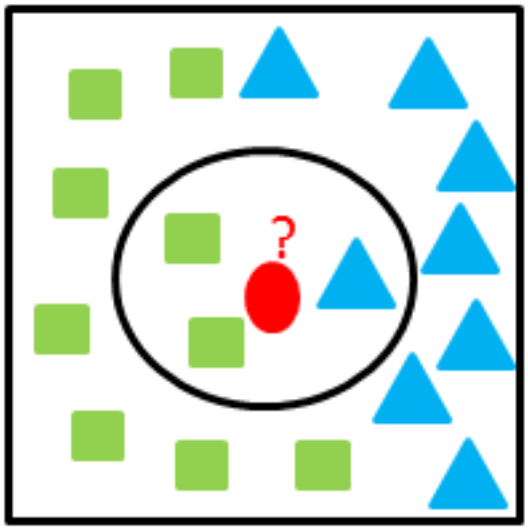

Figure 17: KNN Example.

predicted label by the most common label in a surrounding window of labels. After predicting all labels of a given sentence, each label was replaced with the statistical mode of its surrounding labels. For example, if the statistical mode window is of size 5 and k (number of nearest neighbors) is 3 then 5*3 labels are considered in predicting the label of a feature vector. The window size is referred to in this case as ModeW. An illustration of the KNN for ModeW of 3 and k of 3 is shown in Figure 18.

For each class of label L, $g(L)$ is the number of neighbors of the k nearest neighbors that belong to class L. $g(L)$ can be formulated as in (15).

$$g(L) = \sum_{i=1}^{k} \delta\left(L, \text{label}_i(\text{FV}_t)\right) \tag{15}$$

where:

$$\delta\left(L, \text{label}_i(\text{FV}_t)\right) = \begin{cases} 1, & \text{if } \text{label}_i(\text{FV}_t) = L \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The class label of the ith neighbor of the feature vector acquired at time t FV_t is given by (17).

$$\text{label}_i(\text{FV}_t) = \underset{\forall \text{FV}_i \in T}{\arg\min} \{\text{FV}_t - \text{FV}_i\} \tag{17}$$

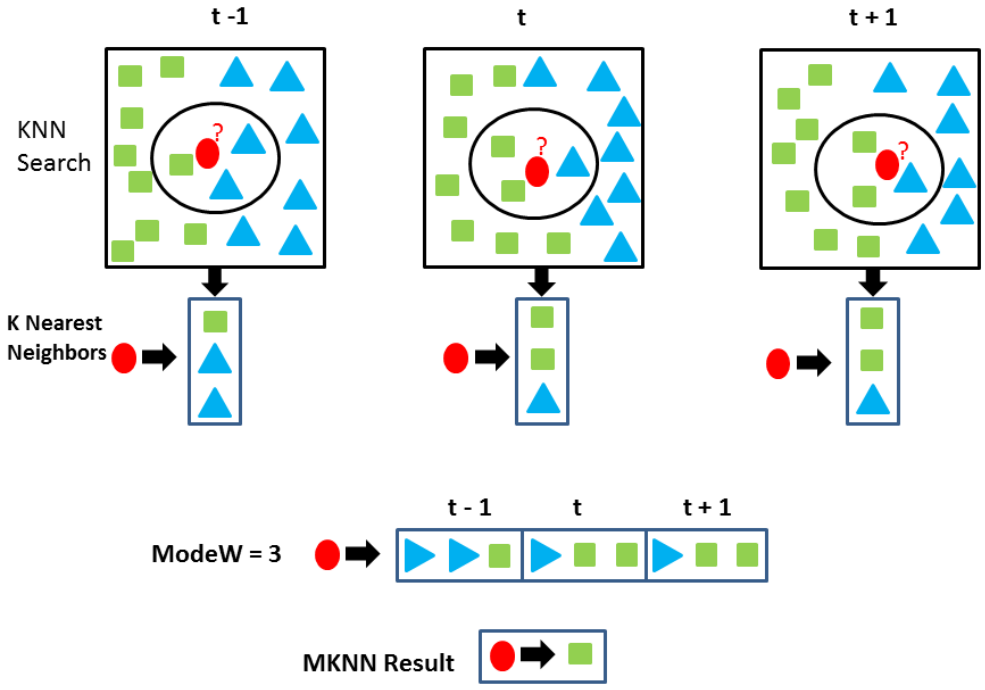

Figure 18: Modified KNN with the statistical mode approach.

In our MKNN the class label L^* is found as in (18).

$$L^* = \underset{L}{\operatorname{argmin}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} \sum_{i} \delta\left(L, \text{label}_i(\text{FV}_{t+j})\right) \tag{18}$$

After predicting a label for each feature vector, similar labels are grouped to form a sign language word.

The algorithm is implemented in MATLAB. First the dataset was divided into 70% for training and 30% for testing. The MATLAB function knnsearch was used to implement the standard KNN algorithm. Given a test sentence with T observations $FV_1, FV_2, FV_3, ..., FV_T$, where each observation FV_t is the feature vector at time t. The knnsearch function is used to find the distance from each observation FV_t to all training instances. The calculated distances are then sorted in an ascending order to specify the k nearest samples. For the MKNN implementation, the k nearest neighbors of the previous and future observations are found similarly. Finally the current observation label is the most common label in a surrounding window of previous, current and future labels as shown previously in Figure 18.

3.2.2. Hidden Markov Models (HMMs). In probabilities, a system can be described by random variables which can change over time. The collection of those variables is called a random process, or simply a process. The values of those random variables at any point of time describe the system at that point. Markov assumption is the assumption that the future state of the process depends only on the current state. A process that satisfies the Markov assumption is called a Markov process.

As defined by Rabiner and Juang [70] "An HMM is a doubly stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols". HMM is popular in many areas; speech processing, natural language processing and gesture recognition are some examples. It was named after the Russian mathematician Andrei Markov in the early twentieth century [71], but the theory of HMMs was developed by Baum and his colleagues in the 1960s [72]. A brief introduction to HMM is presented in this section. For more details, the reader is referred to [70] and [73–76].

3.2.2.1. Discrete HMMs. HMM is defined by the tuble:

$$\lambda = (A, B, \pi) \tag{19}$$

Let S be the set of states labels, and V is the set of observation labels:

$$S = (s_1, s_2, ..., s_N) \tag{20}$$

$$V = (v_1, v_2, ..., v_M) \tag{21}$$

Let Q be a state sequence of length T, and the corresponding observations is O:

$$Q = q_1, q_2, ..., q_T \tag{22}$$

$$O = o_1, o_2, ..., o_T \tag{23}$$

Transition matrix A contains the probability of the system moving from state i to state j:

$$A = [a_{ij}], a_{ij} \quad = \quad P(q_t = s_j | q_{t-1} = s_i) \tag{24}$$

Observation matrix B stores the probability of observation k produced by state i:

$$B = [b_i(k)], b_i(k) \quad = \quad P(o_t = v_k | q_t = s_i) \tag{25}$$

The initial probability of system being in specific state initially is defined by π:

$$\pi = [\pi_i], \pi_i \quad = \quad P(q_1 = s_i) \tag{26}$$

Markov assumption can be written as in (27):

$$P(q_t | q_1^{t-1}) \quad = \quad P(q_t | q_{t-1}) \tag{27}$$

Another assumption is that the observation at time t depends only on the current hidden state. It does not depend on the previous states or observation:

$$P(o_t | o_1^{t-1}, q_1^t) \quad = \quad P(o_t | q_t) \tag{28}$$

Given HMM, three different problems can be of interest; namely evaluation, decoding and learning.

3.2.2.1.1. Evaluation. The evaluation problem is to estimate the likelihood (probability) of a sequence of observation to be emitted by a giving HMM. The problem is to compute $p(o|\lambda)$.

Probability of observations O given a hidden state sequence Q is:

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(o_t | q_t, \lambda) \quad = \quad b_{q_1}(o_1) b_{q_2}(o_2) ... b_{q_T}(o_T) \tag{29}$$

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} ... a_{q_{T-1} q_T} \tag{30}$$

$$P(O|\lambda) = \sum_{Q} P(O|Q,\lambda) P(Q|\lambda) = \sum_{q_1 \cdots q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) ... \tag{31}$$

Calculating the above equation is exponential in T. The forward algorithm is used to reduce the complexity of calculation.

3.2.2.1.2. Decoding. The decoding which is the most common problem is to find the most likely sequence of hidden states given a sequence of observations. The Viterbi algorithm can be used to solve this problem.
The algorithm starts by defining $\delta_t(i)$ which is the probability of the best path to state i in time t.

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1 q_2 ... q_t = s_i, o_1, o_2, ..., o_t | \lambda) \tag{32}$$

The algorithm then evolves as follows:

- Initialization:

$$\delta_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N, \psi_1(i) = 0 \tag{33}$$

- Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t), 2 \leq t \leq T, 1 \leq j \leq N \tag{34}$$

$$\psi_t(i) = arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], 2 \leq t \leq T, 1 \leq j \leq N \tag{35}$$

- Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \tag{36}$$

$$P_T^* = arg \max_{1 \leq i \leq N} [\delta_T(i)] \tag{37}$$

- The optimal sequence can be found though backtracking using the equation below:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T-1, T-2, ..., 1 \tag{38}$$

3.2.2.1.3. Learning. Learning is finding HMM given a set of observations and the corresponding hidden states. Maximum likelihood estimation (MLE) is used to find the parameters as explained in Equations 39-41:

$$a_{ij} = P(t_i|t_j) = \frac{Count(t_i, t_j)}{Count(t_i)} \tag{39}$$

$$b_j(k) = P(o_k|t_j) = \frac{Count(o_k, t_j)}{Count(t_j)} \tag{40}$$

$$\pi_i = P(q_1|t_i) = \frac{Count(q_1 = t_i)}{Count(q_1)} \tag{41}$$

3.2.2.2. Continuous HMMs. In discrete HMM, observations are confined to a set of discrete symbols. $b_j(k)$ is the probability of emitting symbol k at state j, but in continuous HMM the observations are continuous symbols. Here we define $b_j(x)$ where x is a continuous variable.

The observation distribution on the continuous range can take different forms. Gaussian mixtures are probably the most common distribution. It can be defined as in (42):

$$b_j(x) = \sum_{k=1}^{M} c_{jk} \mathcal{N}[x, \mu_{jk}, \sigma_{jk}^2] \tag{42}$$

where:

c_{jk}: mixture weight.

$\mathcal{N}$: Gaussian(normal) density.

μ_{jk}, σ_{jk}^2: mean and covariance of the Gaussian distribution.

3.2.2.3. Implementation. Hidden Markov Models (HMMs) are widely used for sequential data classification in general, and for speech recognition in particular. They are also adopted for SLR and gesture recognition. The majority of SLR toolkits are

developed originally for speech recognition then adopted for SLR. CMU Sphinx [77], HTK toolkit [78], Julius [79], Kaldi [80] and RASR [81] are all examples of open source speech recognition toolkits.

Some researchers created toolkits specifically for gesture recognition, while others modified versions of previously released speech recognition toolkits. For instance, the Georgia tech gesture toolkit GT^2K [82] was created based on a popular speech recognition toolkit known as HTK to provide tools that support gesture recognition research. Additionally, although RASR toolkit was originally developed for speech recognition, it has proved to be flexible and could be easily adapted for different applications such as SLR [83] [28] and optical character recognition [84]. An example of a toolkit created specifically for gesture recognition is the gesture recognition toolkit GRT [85] created by Gillian and Paradiso in 2014 with an emphasis on real-time recognition.

The GT^2K and RASR are selected for our work because they are adequate for SLR and have been used before in similar applications.

The GT^2K toolkit was created based on the HTK to provide tools that support gesture recognition research. It can be used for training models in both real-time mode and off-line mode. To use the toolkit, the user must build the gesture models, specify appropriate grammar, and provide labeled examples for training. The tool will then train models for each gesture. The trained models are used for recognition of new data. More details and examples are available in [82].

The RWTH Aachen University Open Source Speech Recognition Toolkit (RASR), on the other hand, is an open source version of a speech recognition toolkit developed by a group from RWTH Aachen University. Modular design is used for flexibility. It allows most components to be decoupled and replaced at runtime. It comes with comprehensive documentation, examples and tutorials. RASR proved to be applicable for real-life applications; recently it has been used for numerous large vocabulary speech recognition systems by research groups all over the world [86] [87] [88] [89]. The toolkit proved to be adequate for SLR as reported in [28] and [83].

The backbone of the toolkit is a generic data processing framework called the flow network module. The flow network consists of a number of basic data processing

units called nodes connected together by links. Each node is responsible of data manipulation task, such as loading, storing, and caching of data. Flow networks are defined in XML files and then created at runtime based on that definition. They are primarily used for feature extraction and alignment processing, i.e assigning each feature vector to HMM state.

The toolkit supports strict left-to-right HMM topologies. All HMMs have the same number of states, except silence which is modeled by a single state. The Gaussian mixture model is used to model the emission probability. It uses the standard maximum likelihood estimation as well as discriminative training using the minimum phone error (MPE) [90] for Gaussian mixtures estimation. The toolkit itself does not have a module for the estimation of language models; nonetheless the decoder supports N-gram language models in the ARPA format generated by other toolkits.

The toolkit is available for download on [91]. It is published under an open source license, called RWTH ASR License which grants free usage including re-distribution and modification for non-commercial use. In [92], step by step examples, several tutorials and training recipes are available.

Chapter 4: Experimental Results

In this section, we discuss the classification results achieved on the sensor-based datasets and the vision-based dataset. Throughout this section, the results of the three classification tools described in Section 3.2 will be discussed. Namely MKNN, GT^{2}K and RASR. Results are reported in terms of word recognition rate and sentence recognition rate. Word recognition rate is given by (43).

$$\text{Word Recognition Rate} = 1 - \frac{D+S+I}{N}. \tag{43}$$

Where D is the number of deletions, S is the number of substitutions, I is the number of insertions, and N is the total number of words. Sentence recognition rate is the ratio of correctly recognized sentences to the total number of sentences. A sentence is considered to be correctly recognized if and only if all words in this sentence have been correctly recognized without any word being inserted, substituted, or deleted.

4.1. Sensor-based Datasets

There are several parameters that govern the accuracy of recognition using HMM. The most important parameters are the number of states used to represent each gesture and the number of Gaussian mixtures per each state. The effect of these two parameters on the recognition rates of raw data as well as augmented data is shown in Figures 19-22.

Figure 19 and Figure 20 show that increasing the number of states to a certain point enhances the recognition rates. Any increment beyond that point decreases the accuracy or at the best cases, the accuracy saturates. The number of sub-words used to model each sign language word is increased by increasing the number of states.

Emission distribution for each state is modeled by a mixture of Gaussians. Having more mixtures -to a certain extent- allows to fit the actual distribution better, which is apparent by the general trend of the increased accuracy when increasing the number of mixtures as shown in Figure 21 and Figure 22. However, the more mixtures used,

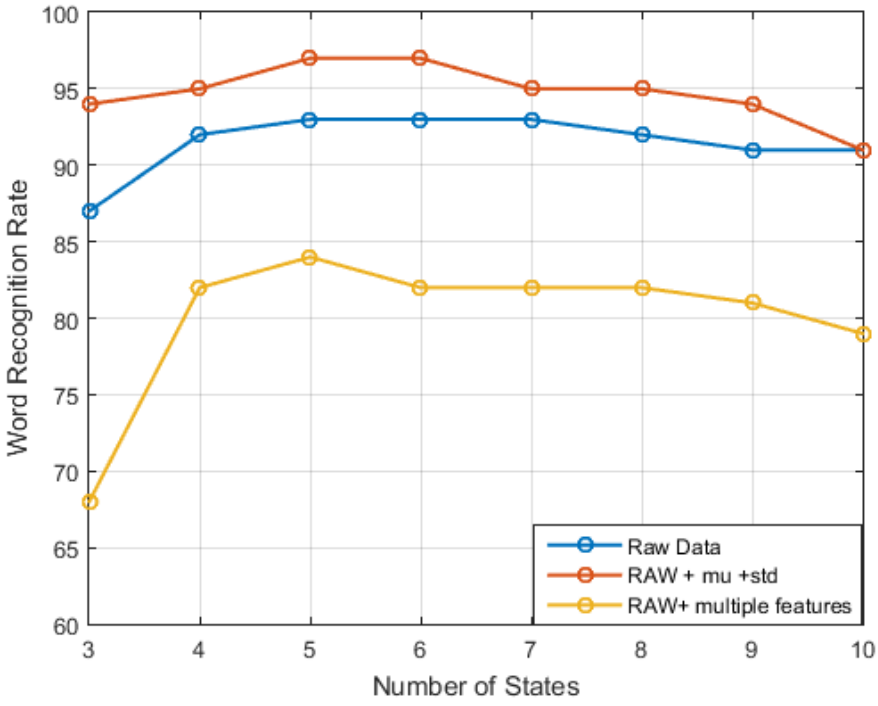

Figure 19: Effect of the number of states on word recognition rate for DG5-VHand dataset.

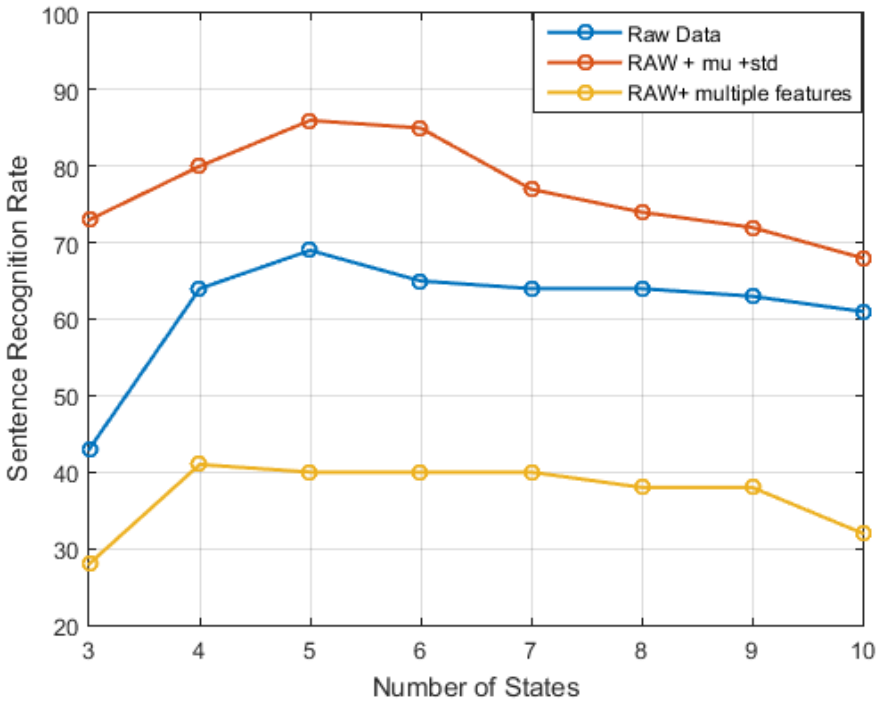

Figure 20: Effect of the number of states on sentence recognition rate for DG5-VHand dataset.

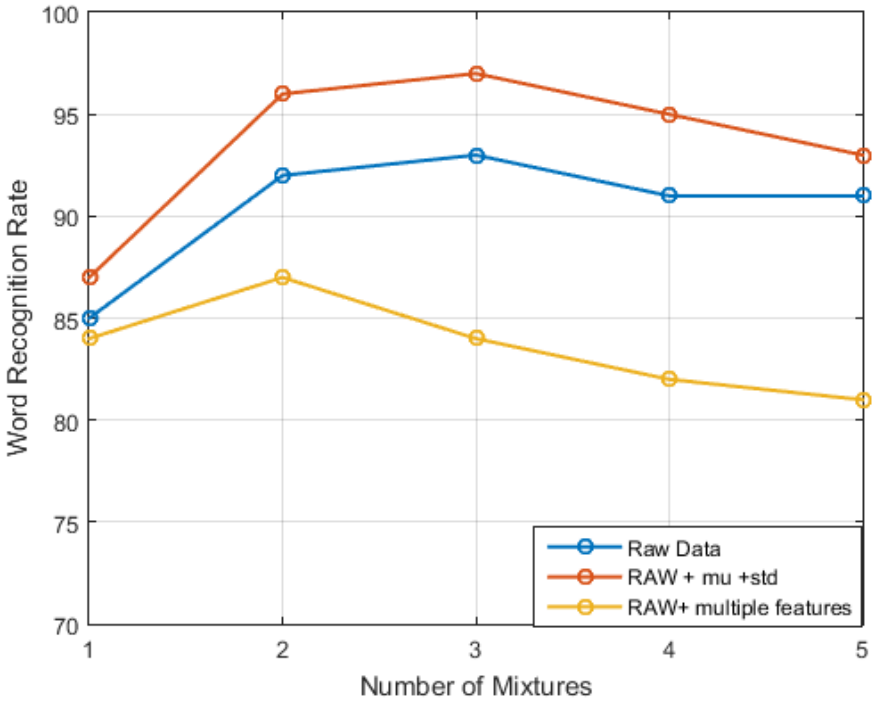

Figure 21: Effect of the number of mixtures on word recognition rate for DG5-VHand dataset.

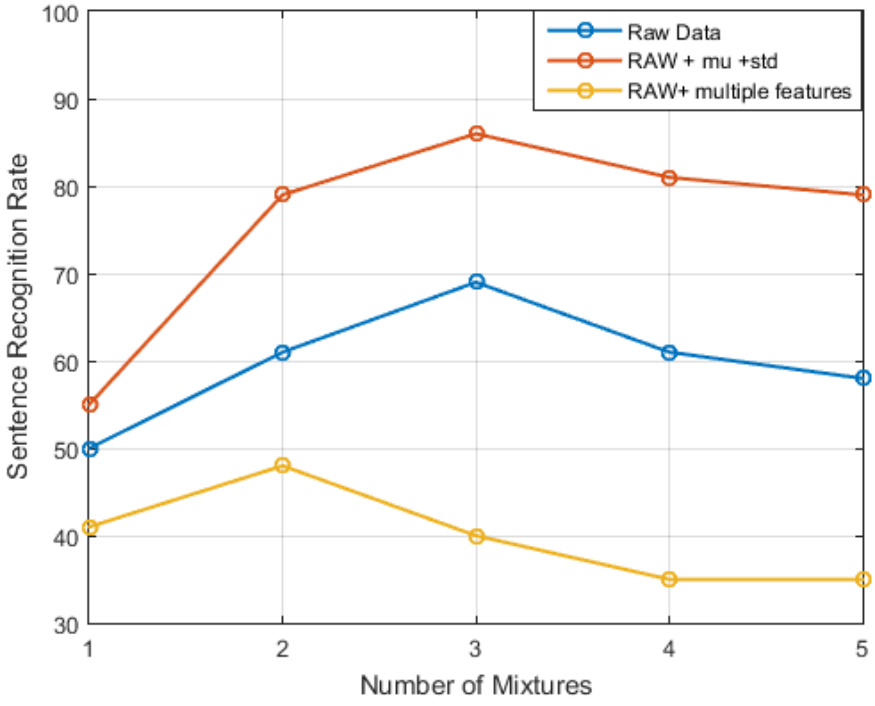

Figure 22: Effect of the number of mixtures on sentence recognition rate for DG5-VHand dataset.

Table 6: Best HMM recognition rates for dataset 1.

Data	No. States	No. mixtures	Word	Sentence
DG5-VHand Raw	5	3	93.0	69.0
DG5-VHand + mu + std	5	3	97.0	86.0
DG5-VHand + multiple features	4	2	86.0	49.0

the more data needed to fit those mixtures. This is apparent by the deterioration in recognition rates for higher number of mixtures.

Best recognition results have been achieved by augmenting the raw sensor data with window-based means and standard deviations. Those statistical features capture contextual information; hence enhance the recognition rates. Adding more statistical features also adds more information but it makes the feature vector much longer which decreases the accuracy once again.

A summary of the best recognition rates for the DG5-VHand is shown in Table 6. The table lists the sentence and word recognition rates for the raw DG5-VHand dataset as well as the augmented datasets. The best number of states and best number of mixture of Gaussians are also stated for each dataset.

As mentioned in Section 3, the Polhemus G^4 provides 6 measurements; the Cartesian position coordinates (x,y,z) and the Euler angles: Azimuth, elevation and roll (a, e, r). Figure 23 and Figure 24 shows HMM results for raw sensor data as well as the augmented data. Again, a window-based approach is used to extract the mean and standard deviations. A similar pattern of increased classification accuracy while increasing the number of HMM states can be noticed. The deterioration for a number of states higher than 6 is obvious as well. Highest word and sentence recognition rates achieved for raw Polhemus G4 tracker dataset are 93.0% and 67.0% respectively. Substantial improvement has been achieved by augmenting raw data with statistical features; word and sentence recognition rates jumped to 97.0% and 85.0% respectively.

The MKNN algorithm has been tested previously on the DG5-VHand dataset [51]. Here we show the results of testing it on the tracker data; both raw and augmented. In MKNN, a sliding window is used as a post-process to replace each predicted label with the statistical mode of its surroundings. The effect of varying this mode window

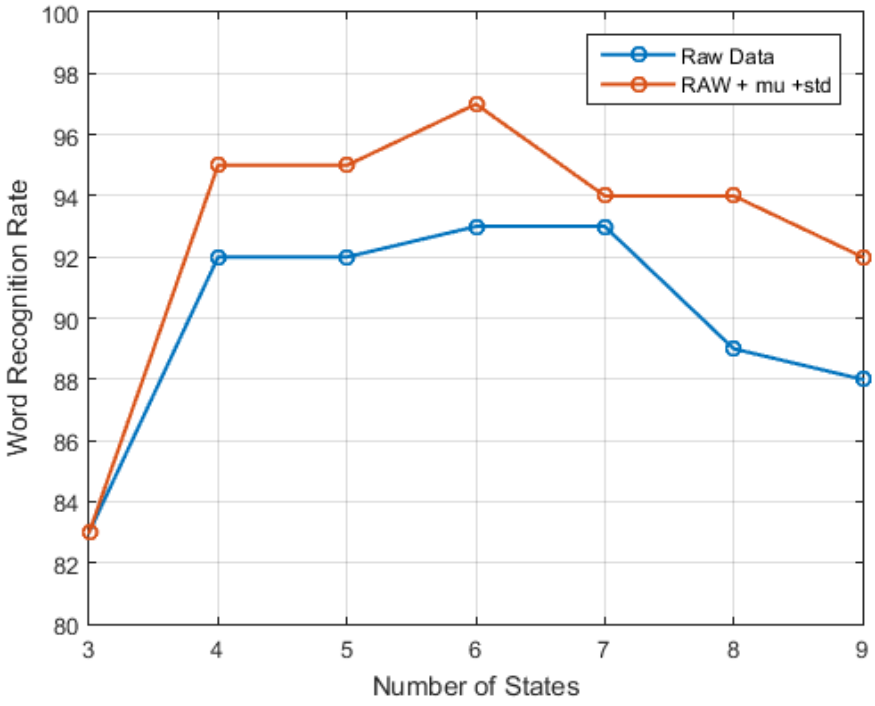

Figure 23: Effect of the number of states on word recognition rate for Polhemus G4 tracker dataset.

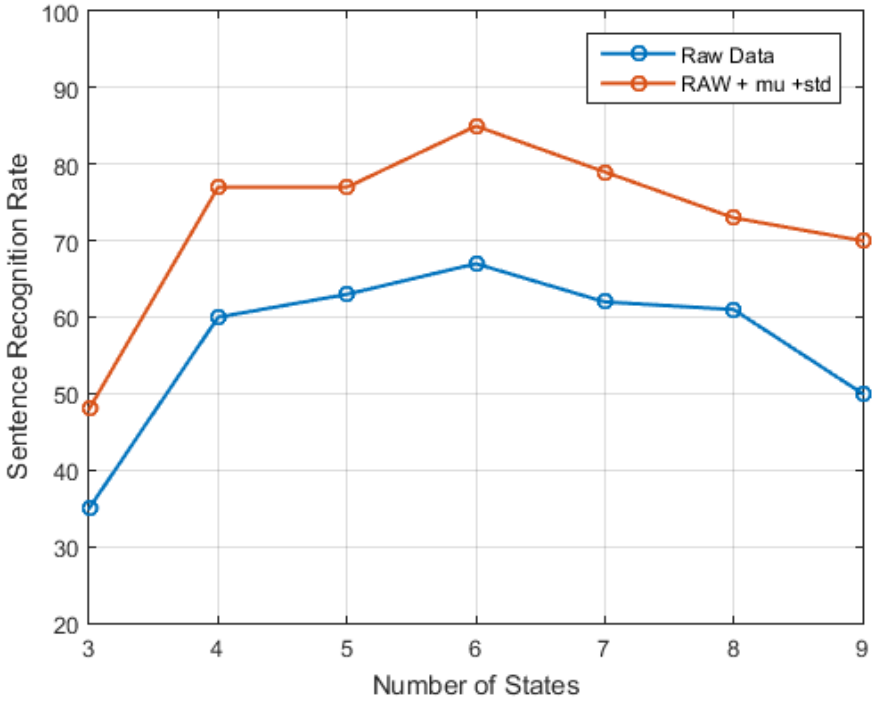

Figure 24: Effect of the number of states on sentence recognition rate for Polhemus G4 tracker dataset.

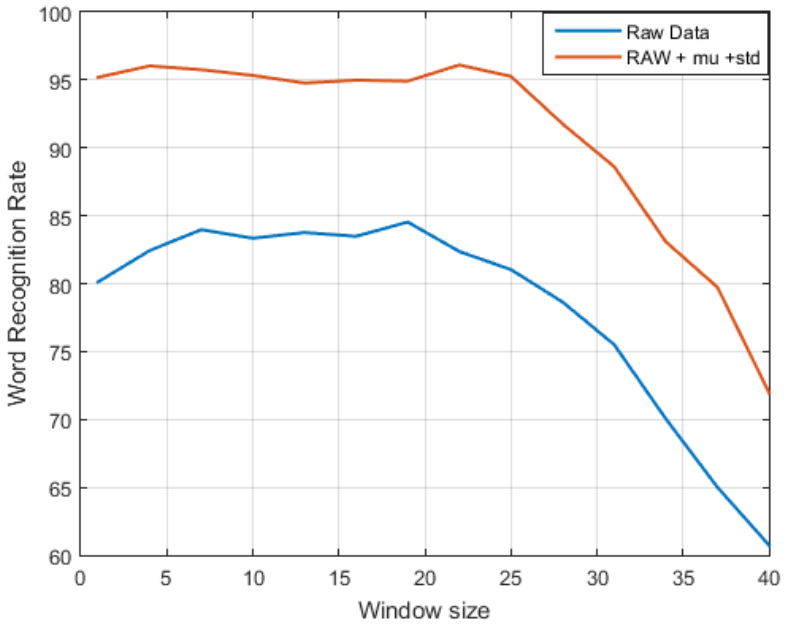

Figure 25: Effect of the window size on word recognition rate for Polhemus G4 tracker dataset.

(ModW) size on classification accuracy is shown in Figure 25 and Figure 26 . In general, the increment in window size enhances the recognition rate as it captures more contextual information. Classification rates will saturate after any further window size increment, and it will decrease rapidly for very wide window size. It happens because large windows will include FVs belonging to other sign language words and will therefore reduce the accuracy of the classifier.

Table 7 summarizes the best word recognition rates achieved for all sensor-based datasets using MKNN and HMM. Table 8 lists the correspondent sentence recognition rates. The most important conclusion is that while Polhemus G4 tracker measures only the position and orientation of the hand, it achieves comparable recognition rates to those of the DG5-VHand dataset. It is also apparent that augmenting raw sensor data with statistical features greatly enhanced the classification accuracy. By examining Table 8, the superiority of MKNN to HMM in terms of sentence classification accuracy can be deduced. It is almost the opposite for word recognition rate where HMM outperformed MKNN in 3 out of 4 datasets.

We move on comparing the performance of two HMM toolkits (RASR and GT^{2}K) on manually labeled datasets. By manually labeled we mean that word bound-

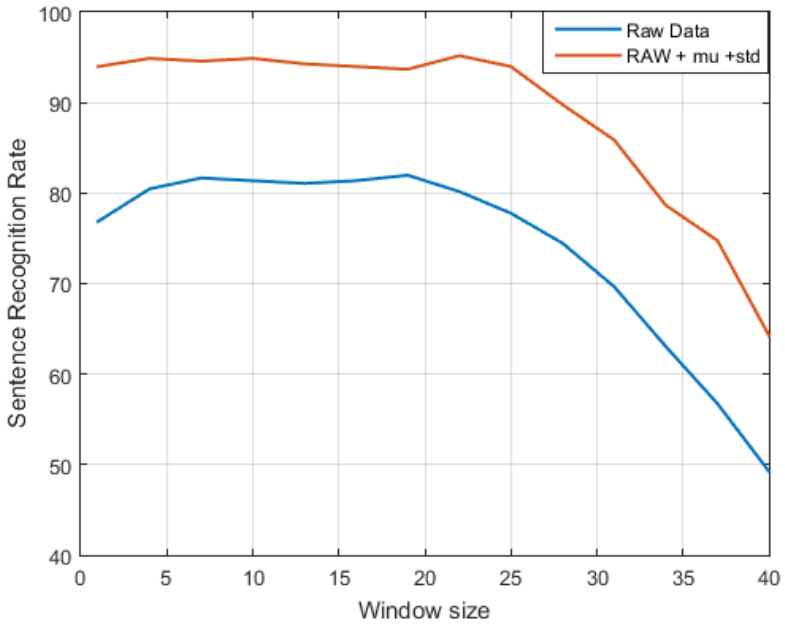

Figure 26: Effect of the window size on sentence recognition rate for Polhemus G4 tracker dataset.

Table 7: Word Recognition rates.

Data	HMM	MKNN
DG5-VHand Raw	93	85
DG5-VHand + mu + std	97	98
Tracker Raw	93	84
Tracker + mu + std	97	97

aries are manually annotated by a human. Both datasets (tracker-based and DG5-VHand-based) are augmented with the statistical features as explained in Section3.1, namely window-based means and standard deviations. These features are used to cap-

Table 8: Sentence Recognition rates.

Data	HMM	MKNN
DG5-VHand Raw	69	82
DG5-VHand + mu + std	86	97
Tracker Raw	67	82
Tracker + mu + std	85	97

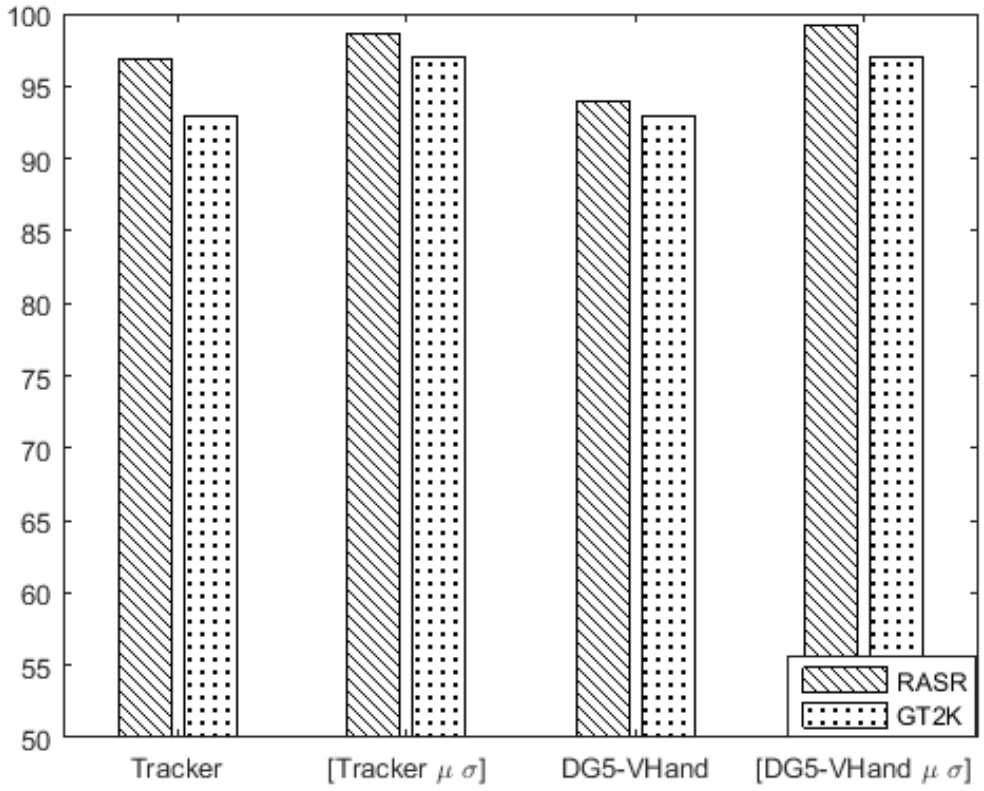

Figure 27: Word recognition rates of manually labeled sensor-based datasets.

ture contextual information. The other features are not used since it was found that they greatly increase the feature vector length which in turn decreases the accuracy.

Figure 27 and Figure 28 plot the classification accuracies for the sensor-based data using the two HMMs toolkits. The classification results are presented for both raw data and raw data augmented with statistical features. It is apparent from the classification results that RASR performance is better than that of the GT^{2}K. This holds for all tests shown with no exception. For instance RASR sentence recognition rate for the augmented DG5-VHand dataset was 96.7% while it was only 86.0% when GT^{2}K was used. We also note that the motion tracker proved to be more accurate than the DG5-VHand glove, however, the augmented DG5-VHand data surpasses the augmented tracker data. A summary of all recognition rates is presented in Table 9. Note that the average of word and sentence recognition rates shown in the table also confirms the superiority of RASR over GT^{2}K.

The performance of RASR on automatically generated labels has been investigated in this work as well. Auto labeling refers to the use of a tool to automatically

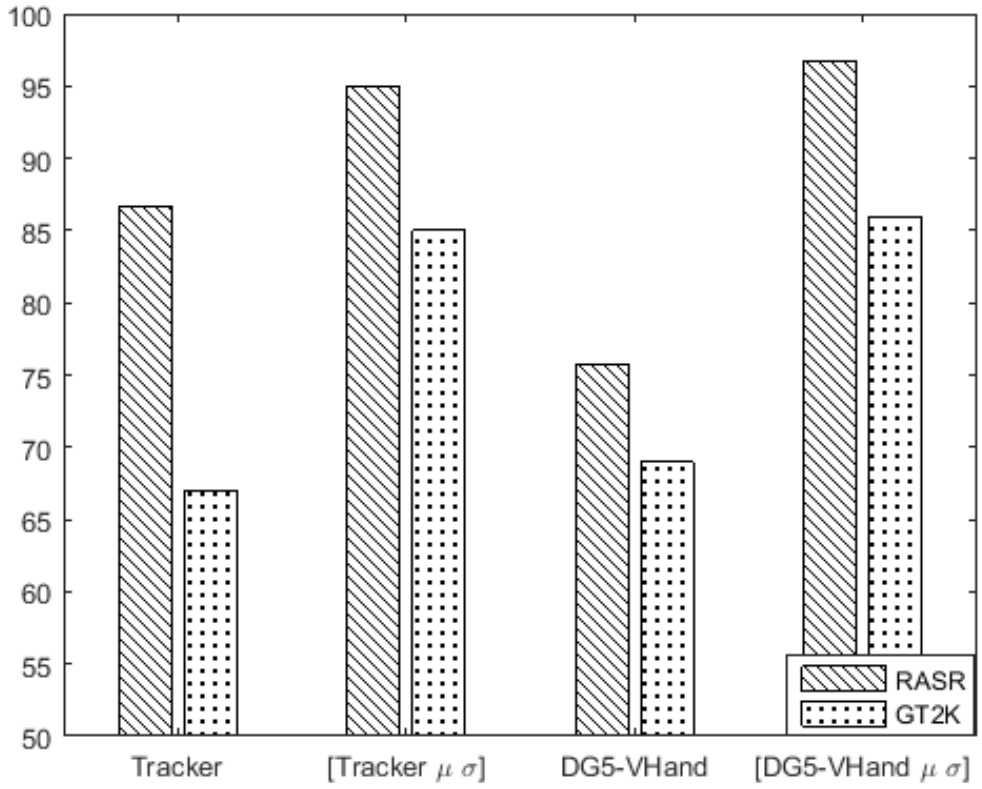

Figure 28: Sentence recognition rates of manually labeled sensor-based datasets.

Table 9: RASR and GT^{2}K comparison on manually labeled datasets

Dataset	RASR		GT^{2}K	
	Word	Sentence	Word	Sentence
Tracker, raw data	96.88	86.67	93.00	67.00
Tracker, augmented raw data	98.64	95.00	97.00	85.00
DG5-VHand, raw data	94.00	75.80	93.00	69.00
DG5-VHand, augmented raw data	99.20	96.70	97.00	86.00
Average	97.18	88.54	95.00	76.75

estimate word boundaries. This could be done using RASR alignment module which automatically assigns each feature vector to an HMM state. This is advantageous because it allows the recognition of sentence-level labeled datasets where only sentence boundaries are annotated. This feature is of high practical gain since manual labeling is a daunting task. Naturally this gain comes at the expense of less accuracy as it could be seen in the results plotted in Figure 29 and Figure 30 where manually labeled datasets always result in higher recognition rates. The accuracy of the auto labeling depends on the accuracy of the raw sensor readings. Since tracker data is highly precise, recognition

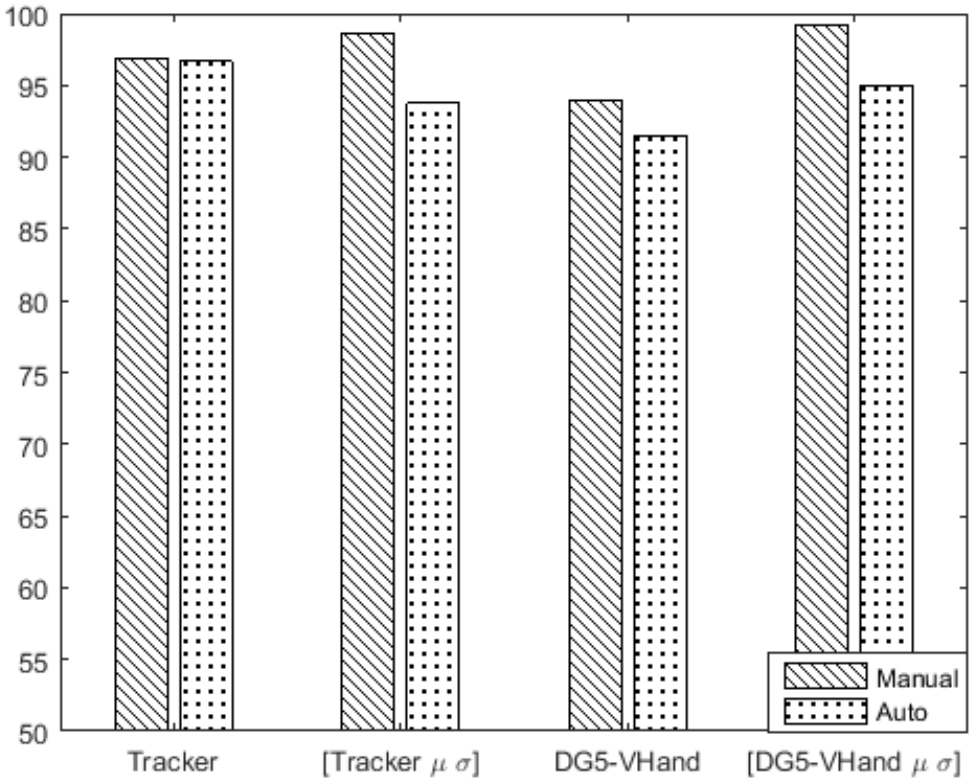

Figure 29: Word recognition rates of auto and manually labeled sensor-based datasets.

rates of its manual and auto labeled dataset were almost the same which were around 96% for both. We have seen previously that augmenting the manually labeled datasets with the statistical features enhanced the accuracy, but this is not the case for auto labeled datasets. This is due to the sliding window approach used for statistical features extraction; it blurs word boundaries making its automatic detection less accurate.

In the following experiments the results of RASR against MKNN classification solution are compared. RASR turned out better in terms of word recognition rate as shown in Figure 31. On the other hand, In comparison to existing work, Figure 32 shows that the MKNN surpasses RASR in 3 out of 4 tests in terms of sentence recognition rates which goes up to 97% for both augmented datasets.

The computational time for each classification approach is listed in Table 10. Results were recorded from 64-bit PC, 4.00 GB RAM, Intel Core i5, running Ubuntu 14.04. Again RASR was superior achieving the least classification time of 2.03 seconds, it was closely followed by the GT^2K. However considering both train and test time, our MKNN is advantageous since it does not require training.

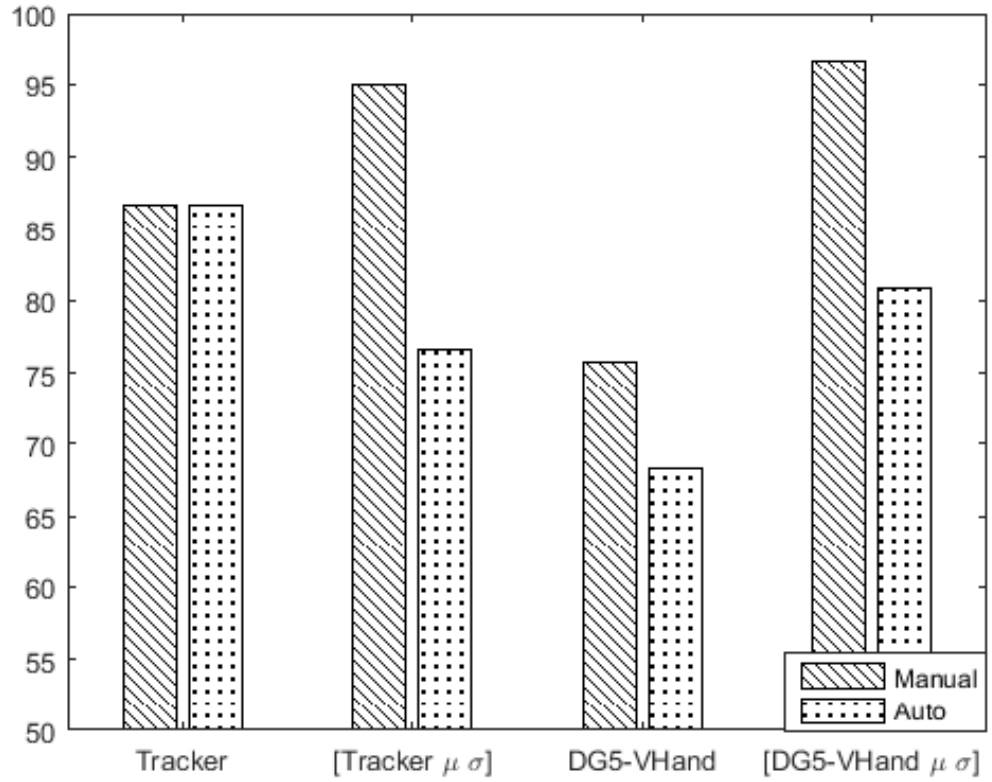

Figure 30: Sentence recognition rates of auto and manually labeled sensor based datasets.

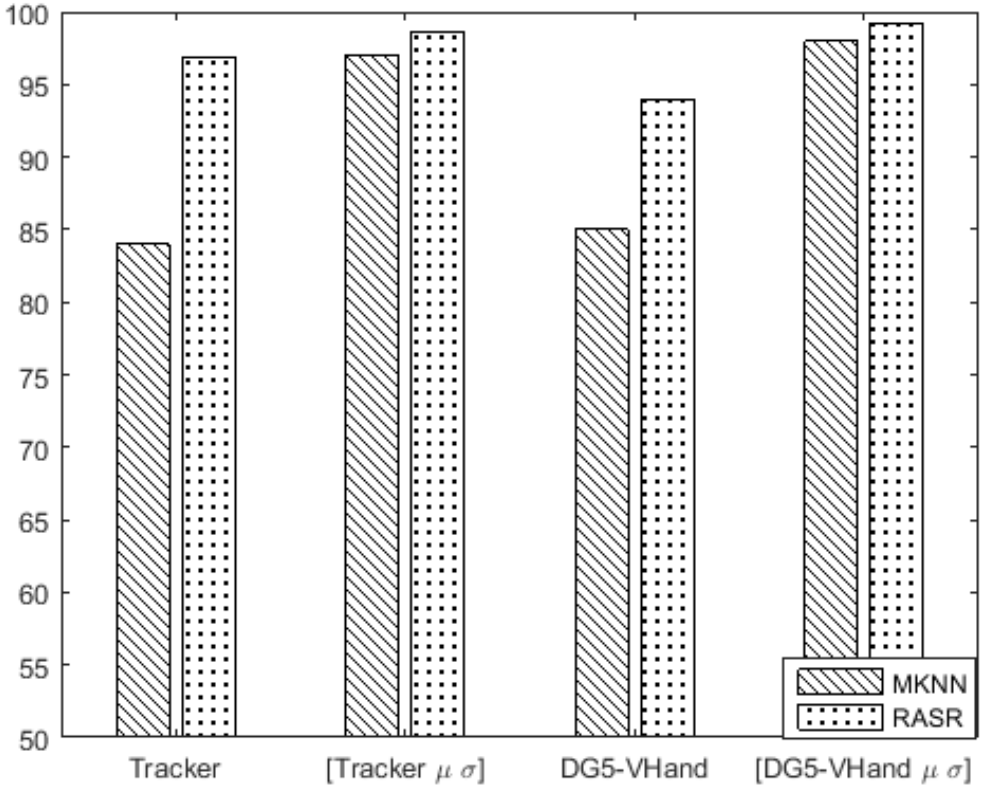

Figure 31: Word recognition rates of MKNN and RASR.

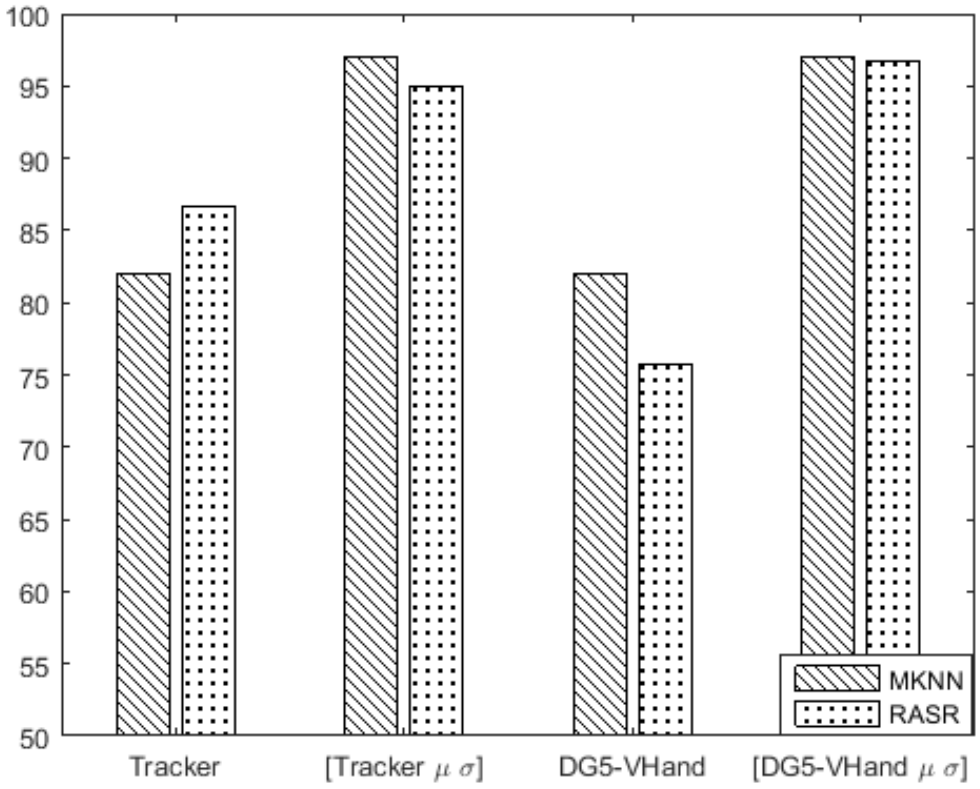

Figure 32: Sentence recognition rates of MKNN and RASR.

Table 10: Computational Time Comparison

Approach	Train Time (sec)	Classification Time (sec)
RASR	55.72	2.03
GT^2K	60.4	2.42
MKNN	-	18.87

4.2. Vision-based Datasets

This section is devoted for a discussion of recognition results of the third dataset which was collected using a camera only.

The feature extraction phase, as explained in Section 3.1, depends on two empirical parameters which must be determined prior to classification. The first one is the DCT cutoff, which is the number of DCT coefficients to retain in a feature vector. Figure 33 shows sentence recognition rates achieved using RASR for various DCT cutoffs. As expected, the recognition rate increases as the number of coefficients increases. This is due to the fact that DCT coefficients are not correlated. Thus, increasing the number

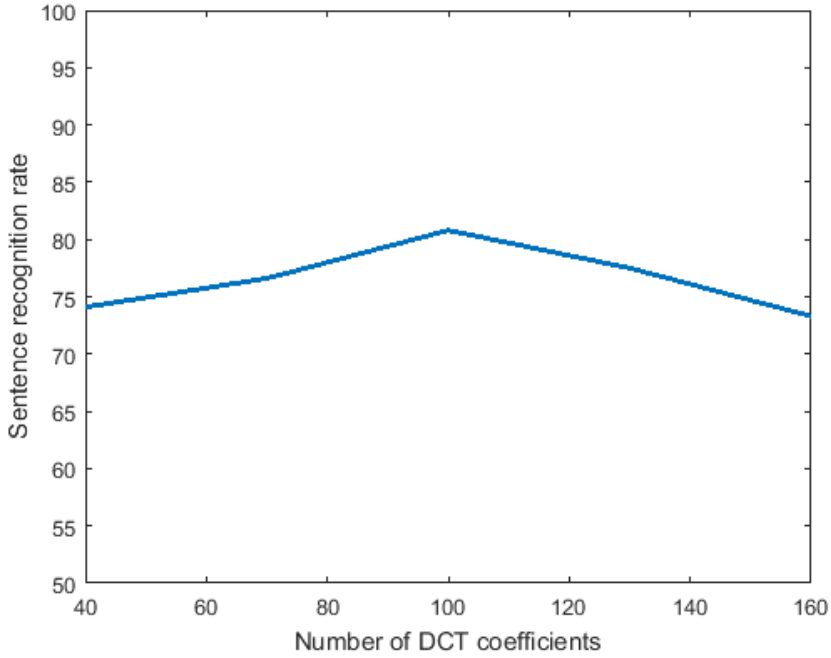

Figure 33: DCT cutoff vs sentence recognition rate.

of DCT coefficients increases the entropy. Nevertheless, recognition rates in general decrease as the dimensionality of the feature vector increases, thus there is normally a point after which any increment in the DCT cutoff will cause the recognition rates to decrease. In our case, the best classification rate is achieved with 100 DCT coefficients as shown in Figure 33.

The second parameter to be determined empirically is the weighting parameter x of (11). Figure 34 shows its effect on word recognition rate using MKNN. The highest rate achieved at $x = 1$.

Firstly, we form feature vectors using DCT coefficients of raw images instead of image differences. We apply 2D DCT transformation to raw images and retain the top left DCT coefficients using zig-zag scanning. The feature vectors are then fed to the three classification approaches; MKNN, RASR and GT^{2}K. The word and sentence classification results are shown in Table 11. It is shown that the highest classification results are achieved by RASR.

Next, the effect of computing 2D DCT of thresholded image differences is examined. When comparing Tables 11 and 12, it is noticed that the improvement as a result of using the thresholded image differences. Recognition rates of all approaches used had increased with no exception. For instance RASR sentence recognition rate

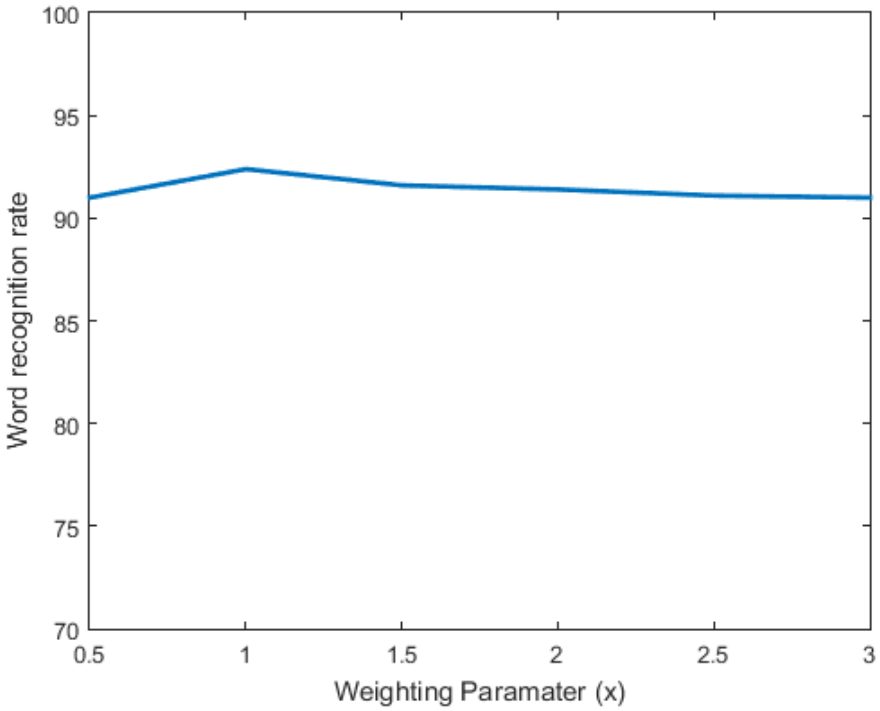

Figure 34: Values of weighting parameter vs recognition rate.

Table 11: Recognition rates of raw vision data.

MKNN		RASR		GT^{2}K	
Word	Sentence	Word	Sentence	Word	Sentence
82.50	77.20	**94.18**	**80.80**	93.10	73.00

increased from 80.8% to 85.0%. This is due to the motion between successive frames being emphasized by the thresholded image difference approach. The results in the tables also show that the proposed classification solutions using RASR and MKNN are superior to existing work [50].

To summarize, the best recognition rates of sensor and vision based datasets are listed in Tables 13 and 14. It is shown that MKNN always achieves the best sentence recognition rate. On the other hand, in terms of word recognition rates, RASR generates the best rates. Additionally, the summarized results reveal that data acquisition through

Table 12: Recognition rates of thresholded image difference.

MKNN		RASR		GT^{2}K	
Word	Sentence	Word	Sentence	Word	Sentence
91.60	**89.17**	**95.60**	85.00	94.00	80.00

Table 13: Best Word recognition rates.

Dataset	Approach	Rate
[Tracker μ σ]	RASR	98.64
[DG5-VHand μ σ]	RASR	99.20
Vision	RASR	95.60

Table 14: Best Sentence recognition rates.

Dataset	Approach	Rate
[Tracker μ σ]	MKNN	97.00
[DG5-VHand μ σ]	MKNN	97.78
Vision	MKNN	89.17

motion trackers on their own suffice for sign language recognition. This is an interesting finding taking into account that no data gloves are needed. Lastly, the results in both tables confirm that sensor-based data acquisition results in higher recognition rates in comparison to the camera-based approach.

4.3. Multiple Users

An open problem of SLR is applying the system to a user (signer) on whom the system has not been trained. The way in which each person performs signs might be different. These interpersonal variations in the signs make the recognition even harder. In this section, the performance of Polhemus G^4 and RASR on multiple users is examined. Another user was asked to perform the dataset of the 40 Arabic sign language sentences using Polhemus G^4. Same as the previous user, each sentence was repeated 10 times. RASR results of both users are depicted in Figure 35. It is apparent that the performance of the Polhemus G^4 and RASR on both users is fairly close. For the first user, word and sentence recognition rates are 96.88% and 86.67% respectively, compared to 95.00% 84.00% word and sentence recognition rates for the second user.

Another test was performed on the datasets of both users combined. The combined dataset consists of 40 Arabic sign language sentences. Each sentence was repeated 20 times; 10 times by user 1 and 10 times by user 2 as shown in Table 15. 70.0% of the combined dataset was used for training and the rest of the 30.0% for testing. 94.5% word recognition rate and 81.2% sentence recognition rate were achieved on the

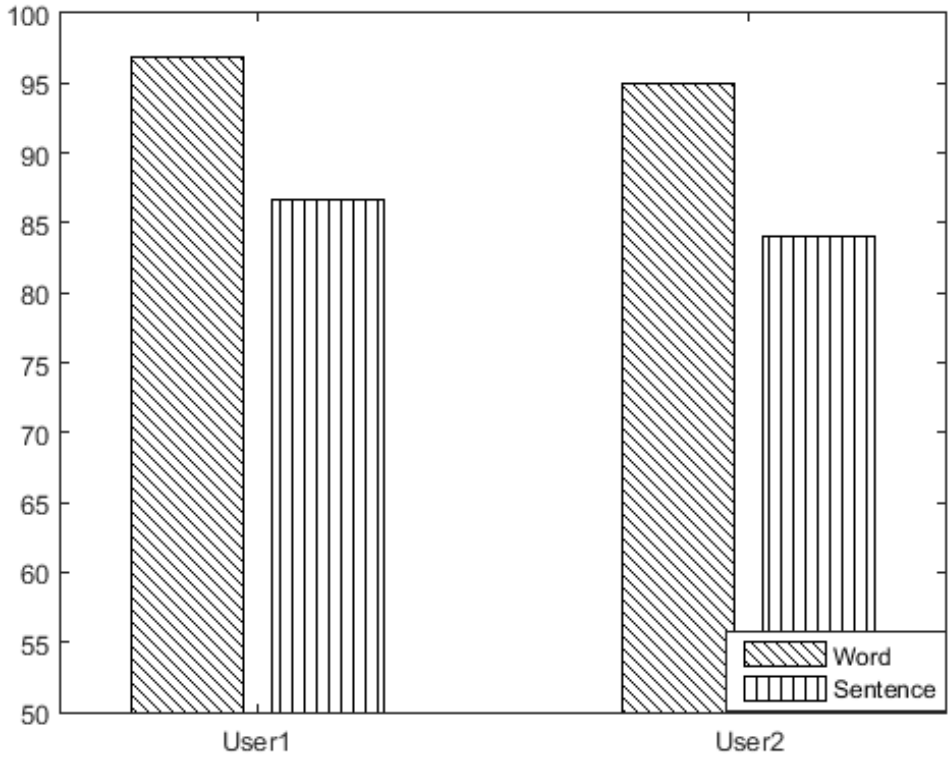

Figure 35: RASR recognition rates of multiple users.

Table 15: Datasets of multiple users collected using Polhemus G^4.

User	Number of Sentences	Number of Repetitions	Total Number of Sentences
User 1	40	10	400
User 2	40	10	400
combined	40	20	800

combined dataset as shown in Table 16. Although the recognition rates have decreased, it is still considered very high.

For the final set of our experiments, RASR was trained on one user and tested on another one. Table 17 shows the confusion matrix of the results of this test. The main diagonal elements are the results of training and testing on the same user, they were reported before in Table 16 and repeated again to aid comparison. The anti-diagonal

Table 16: RASR performance on multiple users' datasets collected using Polhemus G^4.

User 1		User 2		Combined	
Word	Sentence	Word	Sentence	Word	Sentence
96.88	86.67	95.00	84.00	94.5	81.2

Table 17: Confusion matrix of word recognition rates of multiple users' datasets.

		Testing dataset	
		User 1	User 2
Training dataset	User 1	96.88	47.60
	User 2	50.30	95.00

elements show the results of training on one user and testing on the other one. It is quite clear that the performance deteriorates considerably when the algorithm was tested on a different user than the one it has been trained on. For instance, only 50% word recognition rate was achieved when RASR was trained on User 1 and tested on User 2. This could be attributed to many factors; first, many users might be necessary to train a user-independent system. Another factor could be the features themselves; as mentioned previously, the Polhemus G^4 measures only the position of the hand and this might not be descriptive enough to build a user-independent system.

Chapter 5: Conclusion

This thesis examined various data acquisition approaches and various classification techniques for Arabic Sign language recognition. Two datasets were introduced using motion detectors and a camera. A third data set was acquired using data-gloves which was reused from a previous work. Three classification tools were used; MKNN, RASR and GT^{2}K. The thesis also used various feature extraction approaches including window-based statistical features and 2D DCT transformation. The experimental results revealed that the adopted feature extraction techniques used, enhanced the recognition rates for both sensor-based and vision-based datasets. The results also revealed that RASR is superior to GT^{2}K in terms of word and sentence classification rates. RASR also required less computational time for the classification. The modified KNN achieved the best sentence recognition rates for all datasets exceeding both HMM toolkits. Additionally, sensor-based data turned out to be more precise than vision-based data. Although the Polhemus G4 motion tracker only measures hand position and orientation, it achieved higher recognition rates compared to DG5-VHand data gloves, which measure both hand position and configuration. What can be concluded is that motion trackers could be very useful for SLR. The performance of Polhemus G^4 and RASR on multiple users is examined and achieved promising results, but the system still needs to be trained on many users. Moreover, other glove input devices should be used alongside the Polhemus G4 to provide more descriptive features for user-independent Sign language recognition.

In future work, our classification approaches and feature extraction techniques could be tested on large vocabulary. Moreover, our system could be expanded by training it on many users.

References

[1] M. Deuchar, *British Sign Language*. Routledge, 2013.

[2] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, Dec 1998.

[3] "Dgs-corpus," 2015. [Online]. Available: http://www.sign-lang.uni-hamburg.de/dgs-korpus/ [Accessed: 2016-4-27].

[4] "Dictasign project:," 2016. [Online]. Available: http://www.sign-lang.uni-hamburg.de/dicta-sign [Accessed: 2016-4-27].

[5] "Bsl corpus project," 2016. [Online]. Available: http://www.bslcorpusproject.org/ [Accessed: 2016-4-27].

[6] R. Yang and S. Sarkar, "Detecting coarticulation in sign language using conditional random fields," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 2. IEEE, Sep 2006, pp. 108–112.

[7] R. Yang, S. Sarkar, and B. Loeding, "Enhanced level building algorithm for the movement epenthesis problem in sign language recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, June 2007, pp. 1–8.

[8] R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 3, pp. 462–477, Jan 2010.

[9] H. Cooper, B. Holt, and R. Bowden, *Sign Language Recognition*. London: Springer London, 2011, pp. 539–562. [Online]. Available: http://dx.doi.org/10.1007/978-0-85729-997-0_27

[10] S. C. Ong and S. Ranganath, "Automatic sign language analysis: a survey and the future beyond lexical meaning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 873–891, Jun 2005.

[11] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 4, pp. 461–482, July 2008.

[12] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer Graphics and Applications*, vol. 14, no. 1, pp. 30–39, Jan 1994.

[13] "Dg5 vhand," 2016. [Online]. Available: http://www.dg-tech.it/vhand3/index.html [Accessed: 2017-4-21].

[14] "Cyberglove systems," 2015. [Online]. Available: http://www.cyberglovesystems.com/ [Accessed: 2017-4-21].

[15] K. Meyer, H. L. Applewhite, and F. A. Biocca, "A survey of position trackers," *Presence: Teleoperators & Virtual Environments*, vol. 1, no. 2, pp. 173–200, 1992.

[16] S. C. Agrawal, A. S. Jalal, and R. K. Tripathi, "A survey on manual and non-manual sign language recognition for isolated and continuous sign," *International Journal of Applied Pattern Recognition*, vol. 3, no. 2, pp. 99–134, 2016.

[17] M. Al-Rousan and M. Hussain, "Automatic recognition of Arabic sign language finger spelling," *International Journal of Computers and Their Applications*, vol. 8, pp. 80–88, 2001.

[18] K. Assaleh and M. Al-Rousan, "Recognition of Arabic sign language alphabet using polynomial classifiers," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 2136–2145, Jan. 2005. [Online]. Available: http://dx.doi.org/10.1155/ASP.2005.2136

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

[20] C. Oz and M. C. Leu, "American sign language word recognition with a sensory glove using artificial neural networks," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 7, pp. 1204 – 1213, Oct 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197611001230

[21] T. Shanableh, K. Assaleh, and M. Al-Rousan, "Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 3, pp. 641–650, June 2007.

[22] Y. L. Gweth, C. Plahl, and H. Ney, "Enhanced continuous sign language recognition using PCA and neural network features," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2012, pp. 55–60.

[23] J. Forster, C. Oberdörfer, O. Koller, and H. Ney, "Modality combination techniques for continuous sign language recognition," in *Iberian Conference on Pattern Recognition and Image Analysis*. Madeira, Portugal: Springer, Jun. 2013, pp. 89–99.

[24] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Hybrid CNN-HMM for continuous sign language recognition," in *British Machine Vision Conference*, York, UK, Sep. 2016.

[25] J. Pu, W. Zhou, J. Zhang, and H. Li, "Sign language recognition based on trajectory modeling with HMMs," in *MultiMedia Modeling: 22nd International Conference, MMM 2016, Miami, FL, USA, January 4-6, 2016, Proceedings, Part*

I. Springer International Publishing, Jan 2016, pp. 686–697. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-27671-7_58

[26] W. Kong and S. Ranganath, "Towards subject independent continuous sign language recognition: A segment and merge approach," *Pattern Recognition*, vol. 47, no. 3, pp. 1294 – 1308, March 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320313003865

[27] W. W. Kong and S. Ranganath, "Automatic hand trajectory segmentation and phoneme transcription for sign language," in *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, Sept 2008, pp. 1–6.

[28] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, Dec 2015.

[29] W. Gao, G. Fang, D. Zhao, and Y. Chen, "A Chinese sign language recognition system based on SOFM/SRN/HMM," *Pattern Recognition*, vol. 37, no. 12, pp. 2389 – 2402, Dec 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320304001657

[30] G. Fang, W. Gao, and D. Zhao, "Large-vocabulary continuous sign language recognition based on transition-movement models," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 1, pp. 1–9, Jan 2007.

[31] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, X. Chen, and M. Zhou, "Sign language recognition and translation with Kinect," in *IEEE Conf. on AFGR*, April 2013.

[32] X. Chen, H. Li, T. Pan, S. Tansley, and M. Zhou, "Kinect sign language translator expands communication possibilities," *Microsoft Research, outubro/2013, Disponível em:¡ http://research. microsoft. com/enus/collaboration/stories/Kinect-sign-language-translator. aspx*, Oct 2013.

[33] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the Kinect," in *Proceedings of the 13th international conference on multimodal interfaces*. ACM, Nov 2011, pp. 279–286.

[34] S. Lang, M. Block, and R. Rojas, "Sign language recognition using Kinect," in *Artificial Intelligence and Soft Computing: 11th International Conference, ICAISC 2012, Zakopane, Poland, April 29-May 3, 2012, Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 394–402. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-29347-4_46

[35] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, May 2007.

[36] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006. [Online]. Available: http://doi.acm.org/10.1145/1177352.1177355

[37] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231 – 268, March 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S107731420090897X

[38] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, Jan 2002.

[39] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM computing surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, Dec 2003.

[40] A. Tolba, A. El-Baz, and A. El-Harby, "Face recognition: A literature review," *International Journal of Signal Processing*, vol. 2, no. 2, pp. 88–103, Feb 2006.

[41] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259 – 275, Jan 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320302000523

[42] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607–626, April 2009.

[43] X. Wang, K. Liu, and X. Qian, "A survey on gaze estimation," in *Intelligent Systems and Knowledge Engineering (ISKE), 2015 10th International Conference on*, Nov 2015, pp. 260–267.

[44] M. Mohandes, M. Deriche, and J. Liu, "Image-based and sensor-based approaches to Arabic sign language recognition," *Human-Machine Systems, IEEE Transactions on*, vol. 44, no. 4, pp. 551–557, May 2014.

[45] O. Al-Jarrah and A. Halawani, "Recognition of gestures in Arabic sign language using neuro-fuzzy systems," *Artificial Intelligence*, vol. 133, no. 1-2, pp. 117–138, Dec 2001.

[46] I. Elhenawy and A. Khamiss, "The design and implementation of mobile Arabic fingerspelling recognition system," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 14, no. 2, p. 149, Feb 2014.

[47] T. Shanableh and K. Assaleh, "User-independent recognition of Arabic sign language for facilitating communication with the deaf community," *Digital Signal Processing*, vol. 21, no. 4, pp. 535–542, July 2011.

[48] M. Mohandes, S. A-Buraiky, T. Halawani, and S. Al-Baiyat, "Automation of the Arabic sign language recognition," in *Proceedings. 2004 International Conference*

on Information and Communication Technologies: From Theory to Applications. IEEE, July 2004, pp. 479–480.

[49] M. A. Mohandes, "Recognition of two-handed Arabic signs using the cyberglove," *Arabian Journal for Science and Engineering*, vol. 38, no. 3, pp. 669–677, March 2013. [Online]. Available: http://dx.doi.org/10.1007/s13369-012-0378-z

[50] K. Assaleh, T. Shanableh, M. Fanaswala, F. Amin, and H. Bajaj, "Continuous Arabic sign language recognition in user dependent mode," *Journal of Intelligent learning systems and applications*, vol. 2, no. 01, pp. 19–27, Feb 2010.

[51] N. Tubaiz, T. Shanableh, and K. Assaleh, "Glove-based continuous Arabic sign language recognition in user-dependent mode," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 4, pp. 526–533, Aug 2015.

[52] M. Tuffaha, T. Shanableh, and K. Assaleh, "Novel feature extraction and classification technique for sensor-based continuous Arabic sign language recognition," in *Neural Information Processing: 22nd International Conference, ICONIP 2015, Proceedings, Part IV*. Springer International Publishing, Nov 2015, pp. 290–299. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26561-2_35

[53] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, Nov 2011.

[54] Y. Li, X. Chen, J. Tian, X. Zhang, K. Wang, and J. Yang, "Automatic recognition of sign language subwords based on portable accelerometer and EMG sensors," in *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*. New York, NY, USA: ACM, Nov 2010, pp. 17:1–17:7. [Online]. Available: http://doi.acm.org/10.1145/1891903.1891926

[55] X. Chen, X. Zhang, Z. Y. Zhao, J. H. Yang, V. Lantz, and K. Q. Wang, "Hand gesture recognition research based on surface EMG sensors and 2D-accelerometers," in *2007 11th IEEE International Symposium on Wearable Computers*, Oct 2007, pp. 11–14.

[56] Y. Li, X. Chen, X. Zhang, K. Wang, and Z. J. Wang, "A sign-component-based framework for Chinese sign language recognition using accelerometer and sEMG data," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 10, pp. 2695–2704, Oct 2012.

[57] X. Zhang, X. Chen, W.-h. Wang, J.-h. Yang, V. Lantz, and K.-q. Wang, "Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors," in *Proceedings of the 14th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, Feb 2009, pp. 401–406. [Online]. Available: http://doi.acm.org/10.1145/1502650.1502708

[58] V. E. Kosmidou and L. J. Hadjileontiadis*, "Sign language recognition using intrinsic-mode sample entropy on sEMG and accelerometer data," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2879–2890, Dec 2009.

[59] D. Vlasic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popović, "Practical motion capture in everyday surroundings," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007. [Online]. Available: http://doi.acm.org/10.1145/1276377.1276421

[60] L. E. Potter, J. Araullo, and L. Carter, "The leap motion controller: A view on sign language," in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*. New York, NY, USA: ACM, Nov 2013, pp. 175–178. [Online]. Available: http://doi.acm.org/10.1145/2541016.2541072

[61] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and Kinect devices," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 1565–1569.

[62] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, June 2014, pp. 960–965.

[63] 2016. [Online]. Available: http://polhemus.com/motion-tracking/all-trackers/g4 [Accessed: 2016-4-24].

[64] F. Liarokapis, "Dg5 vhand 2.0," 2017. [Online]. Available: http://fotisliarokapis.blogspot.ae/2008/04/dg5-vhand-20.html [Accessed: 2017-3-22].

[65] C. Vogler and D. Metaxas, "Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 1, Oct 1997, pp. 156–161 vol.1.

[66] W. Gao, G. Fang, D. Zhao, and Y. Chen, "Transition movement models for large vocabulary continuous sign language recognition," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, May 2004, pp. 553–558.

[67] H. D. Yang, S. Sclaroff, and S. W. Lee, "Sign language spotting with a threshold model based on conditional random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1264–1277, July 2009.

[68] "zig-zag scanning," 2017. [Online]. Available: http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=1007 [Accessed: 2017-3-22].

[69] "Knn algorithm," 2017. [Online]. Available: http://learning.cis.upenn.edu/cis520_fall2009/index.php?n=Lectures.LocalLearning [Accessed: 2017-3-22].

[70] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, Jan 1986.

[71] A. Andrei, "Markov. an example of statistical investigation in the text of eugene onyeginillustrating coupling of tests in chain," in *Proceedings of the Academy of Sciences*, vol. 7, 1913, pp. 153–162.

[72] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The annals of mathematical statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[73] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov models: estimation and control*. Springer Science & Business Media, 2008, vol. 29.

[74] P. Blunsom, "Hidden Markov models," *Lecture notes, August*, vol. 15, pp. 18–19, Aug 2004.

[75] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[76] "Hidden Markov models," 2016. [Online]. Available: http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html [Accessed: 2016-4-22].

[77] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, "Sphinx-4: A flexible open source framework for speech recognition," Mountain View, CA, USA, Tech. Rep., Nov 2004.

[78] S. Young *et al.*, "The HTK book," *Cambridge university engineering department*, vol. 3, p. 175, Dec 2002.

[79] A. Lee, T. Kawahara, and K. Shikano, "Julius an open source realtime large vocabulary recognition engine," in *EUROSPEECH*, Sep 2001, pp. 1691–1694.

[80] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, Dec 2011.

[81] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Lööf, R. Schlüter, and H. Ney, "The RWTH Aachen university open source speech recognition system." in *Interspeech*, Sep 2009, pp. 2111–2114.

[82] T. Westeyn, H. Brashear, A. Atrash, and T. Starner, "Georgia tech gesture toolkit: supporting experiments in gesture recognition," in *Proceedings of the 5th international conference on Multimodal interfaces*. ACM, Nov 2003, pp. 85–92.

[83] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi, and H. Ney, "Speech recognition techniques for a sign language recognition system," in *Interspeech*, Antwerp, Belgium, Aug. 2007, pp. 2513–2516.

[84] P. Dreuw, D. Rybach, G. Heigold, and H. Ney, "RWTH OCR: A large vocabulary optical character recognition system for Arabic scripts," in *Guide to OCR for Arabic Scripts*. London, UK: Springer, Jul. 2012, pp. 215–254, iSBN 978-1-4471-4071-9. [Online]. Available: http://www-i6.informatik.rwth-aachen.de/rwth-ocr/

[85] N. Gillian and J. A. Paradiso, "The gesture recognition toolkit," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3483–3487, Jan 2014.

[86] J. Lööf, R. Schlüter, and H. Ney, "Efficient estimation of speaker-specific projecting feature transforms," in *Interspeech*, Antwerp, Belgium, Aug. 2007, pp. 1557–1560.

[87] D. Rybach, S. Hahn, C. Gollan, R. Schlüter, and H. Ney, "Advances in Arabic broadcast news transcription at RWTH," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Kyoto, Japan, Dec. 2007, pp. 449–454.

[88] M. Sundermeyer, M. Nußbaum-Thom, S. Wiesler, C. Plahl, A. El-Desoky Mousa, S. Hahn, D. Nolden, R. Schlüter, and H. Ney, "The RWTH 2010 Quaero ASR evaluation system for English, French, and German," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, May 2011, pp. 2212–2215.

[89] C. Plahl, B. Hoffmeister, M.-Y. Hwang, D. Lu, G. Heigold, J. Lööf, R. Schlüter, and H. Ney, "Recent improvements of the RWTH GALE Mandarin LVCSR system," in *Interspeech*, Brisbane, Australia, Sep. 2008, pp. 2426–2429.

[90] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, May 2002, pp. I–105–I–108.

[91] D. Rybach, "RWTH ASR - the RWTH Aachen university speech recognition system," 2017. [Online]. Available: http://www-i6.informatik.rwth-aachen.de/rwth-asr [Accessed: 2017-3-22].

[92] 2017. [Online]. Available: http://www.hltpr.rwth-aachen.de/rasr/manual [Accessed: 2017-1-20].

Vita

Mohamed Hassan was born in 1992, in Khartoum, Sudan. Mohamed has excellent academic track-record including a MSc CGPA 3.9, BSc First Class CGPA 8.3/10 ranked first in Electronic and Computer Systems in University of Khartoum. He has gained extensive research experience in computer vision for the analysis of human movement leading to first author peer review international conference publication and several IEEE journal paper submissions under review. His outstanding academic performance is acknowledged through several awards including a prestigious Singapore International Pre-Graduate Award in 2016 and the Sudanese Engineering Association Award for Best Graduate Project in 2013. He has gained relevant research experience as a Junior Research Assistant at the University of Illinois in Singapore, as an Image Processing Specialist in the Electro-optics research center in Sudan and as a Graduate Teaching Assistant in UAE and Sudan.