TOWARDS A CHEREME BASED DYNAMIC SOUTH AFRICAN SIGN LANGUAGE GESTURE RECOGNITION SYSTEM

by

Addmore Machanja

A thesis submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

at

The University of the Western Cape

Supervisor: Prof Vladimir B. Bajic

UNIVERSITY of the
WESTERN CAPE

November 2007

https://etd.uwc.ac.za/

Declaration

I declare that the work contained in this thesis is my own original version. It has never been submitted to any other University for degree purposes. All the images that are published in this work were exclusive published with the full consent of the people involved.

Addmore Machanja November 2007

Signed…………………………………………..

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Acknowledgement

My heart-felt gratitude goes to Dr Lorna Holtman and her team, the Dean of Science Prof Jan van Beverdonker, and to all the staff members in the Computer Science Department for the moral, logistical and the financial support they gave me during my studies. I would also like to thank my former supervisor; Prof Christian W. Omlin for giving me the opportunity to enroll for a PhD at UWC. During my early days of stay in SA, Prof Omlin used his own personal resources to help me to settle down. I would like to thank my supervisor, Prof Vladimir B. Bajic, for his timely guidance which helped me to finish this thesis. Prof Vladimir B. Bajic started working with me at a time when I was beginning to lose hope in my work. His foresight helped me to focus on those aspects of my research which I could possibly complete within a prescribed time-frame. Above all, Prof Vlad was very patient with me and he was also very much concerned about my well-being. I owe him a lot of thank you(s). I would also like to thank my colleagues; Paul Kogeda, Camilius Sanga and many other people for the encouragement they gave me during my studies.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

DEDICATIONS

To my wife and children: This is the culmination of the events that kept us apart for several years. These were trying times but I salute your resilience and patience.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Abstract

Hand gestures are a natural and intuitive way of human to human communication. Motivated by the achievements made towards automatic speech recognition, and by the ease with which people sign, many researchers started working on sign language recognition systems. Besides, technologies used to build gesture recognition systems pose as an alternative to the cumbersome and the failure prone mechanical devices that are currently used as human-machine interface devices. Most of the available gesture recognition systems represent each sign language gesture with an individual gesture model. Such systems can only recognize a limited number of dynamic sign language gestures. It is cumbersome to build and maintain a gesture recognition system that uses thousands and thousands of individual gesture models. Sign language linguists argue that all sign language gestures are derived from small sets of reusable components, the cheremes. However, computer vision is such an ill-posed problem to the extent that it very difficult to sufficiently detect the basic gesture components from image data during image processing. In most cases important gesture information is lost as a result of occlusion, image noise or during the process of transforming 3D world views into 2D projections. Gesture recognition systems that recognize a large vocabulary of sign language gestures can only be built if we devise image processing algorithms that achieve high quality hand segmentation and tracking. This research presents a multi-cue based segmentation method that helps to improve the extraction of the hand-shape chereme. A Support Vector Machine (SVM) is then used for verifying the hand-shapes that are associated with each input gesture. Hand segmentation results directly affect the extraction of the hand position and hand movement cheremes. The hand movement patterns are learnt and recognized through the Hidden Markov Model (HMM). A sequence of cheremes that represent each gesture is used to build an online gesture dictionary that helps the gesture recognition module to classify the input gestures. In this research, video footages of signing people are used as input gestures. Since the meaning of a gesture differs from society to society, in this project we only focuses on dynamic gestures from the South African Sign Language (SASL). The technologies

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

used in this project will find many applications in various fields of Human Computer Interaction (HCI).

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Acronyms

HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
SASL	South Africa Sign Language
PC	Personal Computer
YCrCb	A colour format where Y is the luminance, Cr and Cb are the red and the blue chroma components
SVM	Support Vector Machine
3D	Three dimensional
2D	Two dimensional
CPU	Central Processing Unit
KLT	Kanade-Lucas-Thomas
RGB	A colour format where R is the red component, G is the green component and B is the blue component
HSV	A colour format where H is the hue, S is the saturation and V is the value/luminance component
MHI	Motion History Image
ASL	American Sign Language
ROI	Region of Interest
VPL DataGlove	A glove based instrument for motion analysis that was developed by the VPL Company

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

TABLE OF CONTENTS

Declaration ii

Acknowledgement iii

DEDICATIONS iv

Abstract v

Acronyms vii

Chapter 1 Introduction to a chereme based dynamic South African sign language gesture recognition system 1

1.1 Introduction 1

1.2 Overview of the existing gesture recognition systems 3

1.3 Research problem 8

1.4 Objectives 9

1.5 Research hypotheses 10

1.6 Premises and assumptions 11

1.7 System design 13

1.8 Accrued benefits 17

1.9 Novel characteristics 18

1.10 Research presentation 21

Chapter 2 Gesture and sign language recognition processes 24

2.1 Background 24

2.2 Linguistic and social views about gestures 28

2.2.1 Gesture definitions 28

2.2.2 Popular gesture taxonomies 29

2.3 Languages, gestures and sign language gestures 33

2.4 Gesture recognition systems 35

2.5 SASL cheremes 37

Chapter 3 Image segmentation and tracking 39

3.1 Brief Overview 39

3.2 Introduction 39

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

3.3 Image segmentation 41
3.3.1 Overview of concept 41
3.3.2 Segmentation methods 42
3.3.2.1 Threshold-based segmentation methods 43
3.3.2.2 Edge-based segmentation methods 46
3.3.2.2.1 Edge detection operators 49
3.3.2.3 Rosenfield algorithm for identifying connected components 52
3.3.3 A review of hand gesture recognition methods 53
3.4 Tracking 59
3.5 Conclusion 61
Chapter 4 Segmenting and tracking gesturing hand regions 62
4.1 Introduction 62
4.2 Integrating motion, skin-colour and edge information 63
4.3 Morphological filter 67
4.4 The border tracing algorithms 68
4.4.1 Head tracing algorithm 68
4.4.1.1 Significance of the head position 70
4.4.2 Tracing the boundaries of irregularly shaped objects 71
4.5 Extracting the statistical feature set 77
4.5.1 Extracting fingertip motion 82
4.6 Tracking candidate hand regions 84
4.7 Classifying sets of candidate hand regions 86
4.8 Summary 88
Chapter 5 An evaluation of system performance 89
5.1 Introduction 89
5.2 KLT based hand tracking 89
5.3 Choice of threshold values 91
5.4 Segmentation results 101
5.5 An assessment of the segmentation results 105
5.5.1 Evaluating the low-level hand segmentation process 105

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

5.5.2 Analyzing the hand tracking and recognition processes 110
5.5.3 Verifying the hand-shape recognition processes 113
5.6 An assessment of the segmentation results 115
Chapter 6 Discussion and Conclusion.. 116
6.1 Background 116
6.2 Characteristics of the new gesture recognition systems 117
6.3 Accomplished tasks 117
6.4 Limitations of the study 119
6.5 Future activities... 120
6.6 Concluding remarks... 121
References 122
Appendix A: Letter of Consent 137
Appendix B: Some of the features from person 4 gesture 1.................................. 138

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

LIST OF FIGURES

Figure 1.1: Fundamental stages of a chereme based gesture recognition system...... 15
Figure 2.1: Classification of dynamic gestures 31
Figure 3.1: A 3D Representation of the difference image: Adapted from Gerhard [28] 45
Figure 3.2: A sudden rise in f(x) value Adapted from Srikanth [106] 51
Figure 3.3: Glove-based Japanese Sign language recognition system: Adapted from Sagawa [5] 54
Figure 4.1: Extracting cheremes from visual images 63
Figure 4.2: Neighbour pixels and tracing direction 73
Figure 4.3: Probabilistic matching of image blobs 84
Figure 5.1: Tracking the moving objects using the KLT tracker 90
Figure 5.2: Performance of skin colour threshold values: Adapted from Hualu and Shih-Fu [9] 92
Figure 5.3 (a): Effects of increasing (decreasing) Cr values on the segmentation results 93
Figure 5.3 (b): Effects of increasing (decreasing) edge threshold values on the segmentation results 93
Figure 5.3 (c): Effects of increasing (decreasing) motion threshold values on the segmentation results 94
Figure 5.4: Determining the area covered by false skin regions for Cr≥128 95
Figure 5.5: Relationship between Cr threshold values and rates of false detections and false dismissals 96
Figure 5.6: Relationship between motion threshold values and rates of false detections and false dismissals 98
Figure 5.7: Relationship between edge threshold values and rates of false detections and false dismissals 99
Figure 5.8: Segmentation results for frames 1 and 2 103
Figure 5.9: Segmentation results for frames 3 and 4 104

UNIVERSITY of the WESTERN CAPE

https://etd.uwc.ac.za/

Figure 5.10: Segmentation results for frames 1 and 2 104
Figure 5.11: Segmentation results for a blurred input image 108

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

LIST OF TABLES

Table 2.1: Examples of dynamic SASL gesture cheremes 38

Table 4.1: Determination of the Hand Position chereme 71

Table 5.1: Relationship between Cr values and the sizes of falsely detected or falsely rejected skin-coloured regions 96

Table 5.2: Relationship between motion values and the sizes of falsely detected or falsely rejected moving regions... 98

Table 5.3: Relationship between edge threshold value and the quantity of falsely detected or falsely rejected edges 99

Table 5.4: Describing categories used to classify segmentation results 106

Table 5.5: Observed characteristics of a multi-cue segmentation result shown in Figure 5.12 108

Table 5.6: Observed attributes of a samples of 20 output images......................... 109

Table 5.7: Analysis of blob matching results.. 111

Table 5.8: Automatic blob matching process for data shown in appendix B 112

Table 5.9: Person dependant and person independent SVM based hand-shape recognition rates... 114

Table 6.1: Comparing the new system with other vision-based gesture recognition systems 117

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Chapter 1

Introduction to a chereme based dynamic South African sign language gesture recognition system

1.1 Introduction

Gestures are a natural and an intuitive way of communication [32][15]. Most hearing-impaired people use sign language as their only mode of communication. However, some non-signers do not understand sign language; and hence often there is communication break down between the hearing-impaired people and the non-signers. If automatic gesture recognition systems could be developed, then such systems would lay a firm foundation upon which comprehensive sign language recognition systems would be built. Sign language recognition systems would help to eradicate the current communication barriers that are frequently encountered by the hearing-impaired people [10]. Besides benefiting the deaf communities, technologies used in automatic gesture recognition system can also be adapted for use in a wide range of applications in the field of Human Computer Interaction (HCI). The possible areas of application range from building surveillance systems for secured establishments; processing of medical images; remote controlling of home electronics and remote controlling of mechanical devices that are used at construction sites. Gesture recognition technologies can also be adopted for enhancing human-machine interface devices especially those that are used in the field of robotics and in other gesture driven applications [36].

UNIVERSITY of the
WESTERN CAPE

Most of the available gesture recognition systems can only recognize a limited number of gestures under controlled operational environments [33][19][8][5]. Each sign language is made up of thousands and thousands of individual gestures. This makes it practically impossible to build a gesture recognition system that stores several individual gesture models. In any case, such a system would consume a significantly large portion of the computer memory. Besides, the gesture modeling and searching

https://etd.uwc.ac.za/

processes would drastically affect the application's run-time. Previous efforts to build gesture recognition systems that could recognize a large number of dynamic gestures have not yielded much success [5][8][23]. However, achievements made by researchers in the field of automatic speech recognition have fueled more research activities aimed at producing automatic gesture recognition systems [15][25][32].

A vision-based gesture recognition system that interprets sign language gestures in a similar way to how human beings interpret those gestures would help to enhance the manner in which dynamic gestures are interpreted by machines. However, computer vision is riddled with many problems that relate to the segmentation and tracking of the gesturing hands. Under natural conditions, perfect segmentation and efficient tracking of a gesturing hand are difficult to achieve. This research explores ways of enhancing the hand segmentation and tracking processes. Improving the segmentation results allows us to preserve important gesture information. Identifying the hand shapes, the hand positions and/or the type of motion exhibited by gesturing hands are viewed as the basic steps towards recognizing sign language gestures [7]. However, some sign language linguists' argue that the gesture recognition process must incorporate information about the direction of motion assumed by a gesturing hand [11]. According to sign language linguists, the basic components of a gesture are often called cheremes [16]. All gestures that constitute each sign language are drawn from a limited number of reusable cheremes [23]. Sign language cheremes are historical-cultural in nature; and hence different sign languages often use different sets of cheremes.

UNIVERSITY of the
WESTERN CAPE

In an attempt to recognize large numbers of dynamic South African Sign Language (SASL) gestures, it is envisaged that machines should first be trained to recognize a small set of cheremes that constitute the SASL gestures. It is also envisaged that a machine which is trained to recognize all sign language cheremes, can easily be reconfigured to recognize any arbitrarily chosen dynamic SASL gestures. In attempt to teach machines to identify SASL cheremes, we designed a system that processes video footages of signing people and outputs a statistical description of the hand shapes, hand

https://etd.uwc.ac.za/

positions and the hand trajectories that are associated with each gesture. In this study we analyze both the simultaneous and sequential combinations of cheremes [103] that constitute each gesture. The gesture recognition technologies used in this research can also be extended for recognizing gestures that are used in other HCI applications. Since the meaning of each gesture differ from society to society [53][11], this project is restricted to recognizing dynamic gestures from SASL. SASL cheremes are extracted from a collection of video footages of people who repeatedly performed the same set of gestures. Video images are first broken down into a sequence of image frames. In practice a consecutive sequence of image frames which are extracted from a particular gesture often contain slightly varying descriptions of hand shapes, signing positions and motion trajectories [23]. During the training stage, specific combinations of cheremes that describe each gesture, and any other information that describe the dynamics of the gesturing hand, are fed into the system's gesture dictionary where they are used to represent the gesture in question. An input gesture is deemed recognized only if the probability that a combination of cheremes obtained from the input gesture matches one of the gestures in the system's gesture dictionary is higher than a predefined threshold value.

1.2 Overview of the existing gesture recognition systems

UNIVERSITY of the
WESTERN CAPE

Early works on automatic gesture recognition systems mainly focused on developing gesture-driven interfaces for operating and controlling electronic devices [4][10]. Most of the available gesture-driven interfaces are vision-based devices which are mainly used in the remote controlling of electronic and mechanical devices [20][12][48]. Kunii's [6] computer-based three-dimensional (3D) models of gesturing hands, though not totally vision-based, represents some of the early attempts by computer vision researchers aimed at producing vision-based gesture recognition systems. Though 3D models provide comprehensive gesture information, they are however, computationally expensive to implement hence they are rarely used in applications which require real-time gesture processing [49][50]. The computationally heavy parameter estimations

https://etd.uwc.ac.za/

processes that are associated with 3D modeling often slow down the gesture recognition process. Nowadays most vision-based gesture recognition systems implement appearance based models. Appearance based models exploit the two-dimensional (2D) representations of objects. Two-dimensional representations are relatively easy to compute but they often provide insufficient information about the object of interest. In order to identify specific objects, some algorithms that use appearance based models fuses the object's geometrical information with some low level image cues [23]. In practice 2D models are often negatively affected by the unreliable parameter estimations that are associated with the mapping of 3D real-world views into 2D projections. Yilmaz, Javed and Shah argue that it is extremely difficult to accurately model 3D objects using 2D models since some important gesture information is often lost while transforming 3D real-world views into 2D projections [51].

Thad's research [19] produced one of the most successful vision-based gesture recognition systems. In his first experimental system, Thad tracked a signer's hands using coloured gloves. He later demonstrated that skin-colour information can also be used to identify and track a signing person's hands [60]. Thad's works, just like most of the available vision-based gesture recognition systems, neither address the creation of a person independent full lexicon system nor the spatial aspect of sign languages. In particular, Thad overlooked the importance of fingertip information and the need to extract specific hand-shape information. Nonetheless, Thad's work demonstrated that the Hidden Markov Model (HMM) [40], a mathematical tool that was originally designed for the speech recognition problem, can also be adapted for recognizing continuous sets of sign language gestures.

UNIVERSITY of the
WESTERN CAPE

Although vision-based input devices are an attractive alterative for most HCI applications, computer-vision itself is an ill-posed problem [33]. Rigorous image processing algorithms that mitigate the effects of occlusion, and/or jagged boundaries must be designed [23]. Jagged boundaries are either caused by abrupt camera motion,

https://etd.uwc.ac.za/

fluctuating lighting conditions, blurred image regions or by the processing algorithms. These problems negatively impact on the quality of the hand segmentation process. The segmentation process is even worsened by the fact that in practical situations, some background objects may possess similar colours and/or other visual attributes as the object of interest.

Apart from vision-based gesture recognition systems, Hirohiko and Takeuchi [8][5][43] implemented mechanical glove based gesture recognition systems. These systems mainly exploit dynamic matching techniques that compare the output of a wired glove with some predefined models of some particular gesture components. However mechanical gloves often take away the naturalness with which people sign [15][4]. Besides, if mechanical gloves are deployed in electro-magnetically charged environments, their highly sensitivity nature will drastically affect the quality of the output data. Some people cite health reasons for not using data gloves [4].

Many research activities that attempted to address the problem of locating gesturing hands from a sequence of video images either used the coloured glove approach or they restricted the background environment. However, other systems fuse multiple image cues in order to reduce the impact of background noises on the gesture recognition process [26][52][47]. Zieren *et al.* [33] incorporated an algorithm for tracking and predicting the features of an image in an attempt to increase the chances of correctly localizing the gesturing hand. Besides, Zieren *et al.* also applied probabilistic inference methods in order to determine the trajectory assumed by a gesturing hand. Zieren's approach, to some extend, addressed the hand overlap problem.

UNIVERSITY of the
WESTERN CAPE

Various approaches to the hand segmentation problem have been proposed. For instance, Akyol and Pablo [26] segmented the gesturing hand by combining pixel level skin colour and coarse motion information. It is argued that this approach helped them to achieve fast detection of the gesturing hand [26]. On the other hand, Gerhard [28][31] assumed a stationary background before computing the difference in intensity

https://etd.uwc.ac.za/

between corresponding pairs of image pixels that were arbitrarily taken from any two successive image frames. In Gerhard's work, the difference image is assumed to represent the gesturing hand. Although Gerhard's system achieved 92.7% recognition rate for individual gestures, his work does not address the fluctuating lighting conditions and/or any other unexpected changes in the background conditions. One of the major limitations of Gerhard's work is that it recognizes sequences of continuous gestures as if they were single indivisible entities. Such a system would require too many individual gesture models if it is used for building a comprehensive gesture recognition system. Other interesting approaches to building vision-based gesture recognition systems were separately presented at MIT Media laboratory [34], University of Udine [54] and at the Carnegie Mellon University [10].

The setbacks evidenced in the available gesture recognition systems demonstrate the complexities involved in building gesture recognition systems. In most of the available systems, each gesture is represented by an individual model. However, each sign language is composed of large numbers of gestures and hence it is tedious, time consuming and memory wasting to build and store large numbers of autonomous gesture models that should be used by a gesture recognition system. From a sign language linguist's perspective [7][11][53], each sign language gesture is derived from a small set of atomic gesture components, the cheremes. Stokoe [7], in his work which led to the publication of the first American Sign Language (ASL) sign language dictionary, identified the tab, the dez and the sig cheremes. The tab chereme describes the start and end positions of a gesture. The dez chereme describes the shape assumed by the gesturing hand while the sig chereme describes the type of motion exhibited by a gesturing hand. According to Adam [39] the tab, the dez and sig cheremes later become known as the *location, the handshape and movement* cheremes respectively. Battison, cited in [11], argued that besides the above mentioned cheremes, knowledge of the orientation of a gesturing hand is required in order to successfully recognize a hand gesture. He proposed that palm orientation should constitute the fourth chereme which he named the ori chereme.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

If vision-based gesture recognition systems could use cheremes to recognize dynamic gestures then a small number of models would sufficiently represent all sign language gestures. A successful chereme based gesture recognition system is capable of recognizing large numbers of dynamic gestures without need for building numerous autonomous gesture models. However, for such a system to succeed, firstly there is need to improve the hand segmentation and tracking processes. Perfect hand segmentation enhances hand tracking, and in some cases tracking improves the segmentation process by reducing the probable regions of interest [54][52].

In this research we present a framework for building a vision-based gesture recognition system that recognizes dynamic gestures by first recognizing the basic components of each gesture, the cheremes. The proposed system, hereafter referred to as the Chereme based Recognition of Dynamic Sign LAnguage Gestures (CRD SLAG), first estimates the hand configuration, the movements that are executed by a gesturing hand and the relative hand positions associated with each gesture. Although it is trivial for the human eye to depict the above mentioned factors, under real life situations it is almost impossible for machines to correctly identify different hand shapes from 2D projections. Human hands are non-rigid objects which can spontaneously change their visible configurations in accordance to the task at hand. The segmentation and tracking problems are further compounded by the existence of large variations within the same or different gesture classes. Different people, and in some cases the same individual, perform the same gesture in a slightly different manner. The speed with which the hand moves, the trajectory assumed, and sometimes the position at which a gesture is executed often varies from person to person. According to Sturman and Zeltzer [36], machines often disregard the exact hand configurations; hence they consider all similar looking gestures as if they are different representations of the same gesture.

In this project the CRD SLAG's gesture recognition modules will only be trained to recognize dynamic gestures from the SASL. Knowledge of the structure of some SASL

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

gestures will be used to achieve gesture recognition. Here gesture recognition is only deemed successful if the system first identifies the components of each input gesture and then finds a gesture, from some of the gestures represented in the system's gesture dictionary, that more or less has the same number and the same order of gesture components as those identified from the input gesture.

CRD SLAG is composed of four main modules; the hand localization and segmentation module, the feature extraction module, the hand-shape identification and tracking module, and the gesture recognition module. In this research, localization of the hand is achieved by fusing skin color, motion and the crude object boundary information. Once a gesturing hand has been localized and segmented from the rest of the image data, a minimum set of features that best describes the configurations, the motion, and the position of the gesturing hand is extracted. The displacement vectors that characterize object motion, together with other features that statistically describe an object of interest, are then used to train the Support Vector Machine (SVM) and the Hidden Markov Models (HMMs) which, later on, are respectively used for classifying the hand-shape cheremes and the motion that are associated with each dynamic gesture.

1.3 Research problem

UNIVERSITY of the
WESTERN CAPE

Most of the available gesture recognition systems can only recognize small sets of specific sign language gestures. These gesture recognition systems mainly represent and recognize gestures in their entirety. The available systems are not designed to recognize gestures from their sub-components. Since each sign language is composed of large numbers of individual gestures, it is very difficult to build gesture recognition systems that are capable of modeling and recognizing large numbers of gestures. From a linguist's point of view, each sign language gesture can be broken into its basic components. If we can build a gesture recognition system that recognizes gestures from their basic components, then it would be possible to build gesture recognition systems that recognize large numbers of dynamic gestures. It is important to note that the hand-

https://etd.uwc.ac.za/

shape chereme represents a broad range of possible hand shapes that are of significance in each sign language.

In Section 1.1 of this chapter we outlined some of the factors that make it difficult to achieve perfect hand segmentation. Such problems inherently affect the hand-shape detection process. What then should be done in order to improve the hand-shape detection process and/or the hand segmentation process in general? A successful hand detection and modeling process is a fundamental step towards attaining a chereme based gesture recognition system. However, just like in the automatic speech recognition problem, it is not easy to decompose a gesture into its basic components [35]. The detected position of a gesturing hand is largely dependant on the hand detection process. The position chereme is herein expressed as the relative position of the detected hand blob in relation to the nearest major section of the human body. The centre of mass of each probable hand region is mainly used when determining the position chereme. The movement chereme, which describe the trajectories assumed by gesturing hands, can be recognized by implementing a Hidden Markov Model or any other time series data analysis model. During the training stage, all gesture components that describe each gesture are entered into a system dictionary, and this information will in turn be used for identifying each gesture during the gesture recognition phase.

UNIVERSITY of the
WESTERN CAPE

1.4 Objectives

The main goal of this research is to develop a framework for building a gesture recognition system that is capable of recognizing a large vocabulary of dynamic gestures from the SASL. Gesture recognition systems that recognize individual gestures from their basic components can presumably recognize large numbers of arbitrarily chosen dynamic gestures. A component-based gesture recognition system proposed in Yanghee *et al.* [32] identifies hand movement patterns as the only primary attribute of a gesture. Yanghee *et al.* also pointed out that automatic recognition of

https://etd.uwc.ac.za/

hand movement patterns is not a trivial task. In our endeavour to build a comprehensive gesture recognition system, we proposed the following sub-objectives:

- Isolate SASL cheremes.
- Design a segmentation algorithm that reduces the side effects of image blurring.
- Localize and segment hand and head regions from input image data.
- Extract features that describe each hand-shape and the respective hand positions.
- Develop a machine learning system to assess the possibilities of recognizing hand configuration.
- Validate the hand segmentation and the hand shapes recognition processes.
- Identify hand movement patterns.
- Implement a chereme-based gesture dictionary where different gestures are represented by sequences of their constituent gesture components.
- Review the feasibility of building a comprehensive chereme-based gesture recognition system.

The above mentioned sub-objectives in some way explain how a framework for recognizing a large vocabulary of dynamic SASL gestures would be designed. In the following section we discuss the hypotheses presented for this thesis.

UNIVERSITY of the
WESTERN CAPE

1.5 Research hypotheses

Sign language linguists argue that each gesture is made up of specific combinations of particular gesture components. From a biological point of view, hand movements patterns are determined by the ability and the extent to which the hand joints and muscles can flex [36][6]. Sign language linguists also observed that different hand gestures are sometimes made up of similar hand configurations, related hand movements or are sometimes executed from the same body positions. As such, all gestures that constitute each sign language are derived from a small pool of a cheremes which is peculiar to each sign language. We assume that it is possible to use computer vision algorithms to break down SASL gestures into their constituent gesture

https://etd.uwc.ac.za/

components. Once all the components of a particular gesture are identified, then a definition for each respective gesture, in terms of the identified gesture components, can be entered into the system dictionary. It is also assumed that each gesture would be uniquely identified by a particular combination of gesture components, and hence the chereme-based gesture recognition system would recognize large numbers of SASL gestures. This hypothesis is derived from a linguistic's perception of a sign language [7][39]. In short, our two hypotheses are summarized as follows:

- We can automatically recognize hand-shape, position and the movement patterns (the cheremes) that are associated with each SASL gesture.
- We can recognize a large number of SASL gestures by first recognizing their gesture components.

1.6 Premises and assumptions

The dynamic gesture and the automatic speech recognition problems share many common attributes. For instance, both speech and dynamic gestures are continuous in nature [7] and they are interpreted according to their spacio-temporal contexts [15]. As a consequence of these similarities, researchers working on the dynamic gesture recognition problem found that some tools that were originally designed for automatic speech recognition can also be applied to the dynamic gesture recognition problem. For instance, the HTK, a toolkit that implements the Hidden Markov Model [3], which was originally designed for automatic speech recognition researches, can also be applied for solving the automatic gesture recognition problem [32][19][28]. It has been observed that the HMMs can successfully model the spacio-temporal aspects of dynamic gestures [19][45]. To a greater extent, the presented chereme-based gesture recognition system is designed in a similar manner to some of the presented automatic speech recognition systems [7][35]. In a chereme-based gesture recognition system, it is important to break down gestures into cheremes just like Oscar *et al.* [35] indicated that spoken words are also first broken down into morphemes in the automatic speech recognition problem.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Apart from benefiting from automatic speech recognition research, the continually improving computing and imaging technologies [25] significantly contribute to the successes recorded by some automatic gesture recognition researchers. Nowadays, large quantities of cheap and powerful video cameras, sophisticated computers and other accessories that are required for building successful vision-based gesture recognition systems are increasingly available on the market [25]. Most personal computers (PCs) and workstations have very high processing speeds, multiprocessing capabilities and very large data storage capacities; which are some of the prerequisites for implementing vision-based gesture recognition systems. On the other hand, some computers have inbuilt frame grabbers, high speed firewire ports and/or other digital signal processing chips that are useful for streaming or processing image data. This development has significantly reduced the costs of acquiring extra hardware that are required by vision based gesture recognition systems [4]. High speed and high resolution digital cameras are increasingly becoming more affordable.

The design of the proposed dynamic gesture recognition system takes into consideration the achievements made by the previous gesture recognition systems, and attempts to overcome the limitations exposed by some of the available systems [15]. This research draws many parallels with the gesture recognition systems in which gestures were first subdivided into gesture components [5][8][23][24][25][43], and those systems in which vision-based tracking of hands was implemented [19][4][12][14][28][32]. According to Kjeldsen [4], researchers working on vision-based recognition systems have designed fast and computationally efficient vision algorithms.

UNIVERSITY of the
WESTERN CAPE

In this research, single-handed gestures are captured by a single view digital video camera. The captured gestures are used as input data. It is assumed that all significantly large blobs of moving skin coloured pixels that are detected in the image data, either represent the gesturing hand or the signing person's face. Hand tracking is significantly

https://etd.uwc.ac.za/

enhanced if a signer wears non-skin coloured clothing. A maximum likelihood approach is used for matching the candidate hand regions during the hand detection and tracking process. However, just like in many other probability based processes, a misclassification of a hand region may trigger tracking and/or recognition failures. In this research it is also assumed that the head region can easily be detected by analyzing the contours that circumscribe the skin coloured blobs. We assume that the head region is represented by the largest elliptical shaped blob, which is mostly found at the top section of each image frame.

1.7 System design

In a computer system, image data is represented by sets of binary numbers that correspond to specific lighting intensities. Given this background, how then does a computer identify moving hands and/or how does a computer detect hand motions? In some vision-based gesture recognition systems, the problem of tracking gesturing hands is simplified by using coloured gloves under controlled backgrounds [19][37]. Although Thad argues that skin colour information can be used for tracking the moving hands [60], other researchers say that skin colour alone cannot unambiguously isolate an object of interest [26][79]. In particular, Akyol [26] explains that skin-colour information on its own does not guarantee perfect hand segmentation and tracking. The assumption that skin-colour would only appear on the hand and face regions, and not on other background objects, is not always true. Those researchers who only used motion or colour information to segment an object of interest, in most cases produced partially reliable hand segmentation results [26][52].

UNIVERSITY of the
WESTERN CAPE

In this research, dynamic gesture information is extracted from sequences of video image frames. The hand segmentation algorithm first isolates the moving skin-coloured pixels basing on particular combinations of motion, skin colour and edge information. We search for the occurrence of certain YCrCb colour [81][29] values in each input image. Chai [47] discovered useful ranges of the Cr and Cb values that can be used for

https://etd.uwc.ac.za/

locating the probable skin regions. All non-skin coloured regions are discarded immediately after they are detected since no further processing is required for such regions. We obtain coarse hand motion information by subtracting the grayscale values of the previous image frame from those of the current image frame. Grayscale values are represented by the Y component of the YCrCb colour space. Coarse object boundary information is extracted by implementing an improved boundary tracing algorithm. Unlike in Liu and Lovell's work [24] where object boundaries are merely estimated by implementing the Canny edge detector, the boundary tracing algorithm presented in this research traces all the possible outer boundaries of a moving skin-coloured blob. Depending on whether a dead end is met before all boundary pixels are traced, our boundary tracing algorithm can reorient itself for tracing object boundaries in either clockwise or anticlockwise direction. Our boundary tracing algorithm is also designed to overcome the difficulties that are often encountered when tracing jagged or blurred boundaries which are either induced by image noise, image processing modules or blurring effects that are caused by fast moving objects.

While trying to isolate the probable skin-coloured regions, firstly we fuse the image edge data with both the skin-colour and motion information. The Sobel edge detector is used for computing the image edges. Large skin-coloured blobs are either classified as face or hand regions depending on the total surface area, the shape and/or the magnitude of motion exhibited by an image blob. Figure 1.1 shows some the important stages of the proposed gesture recognition system.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

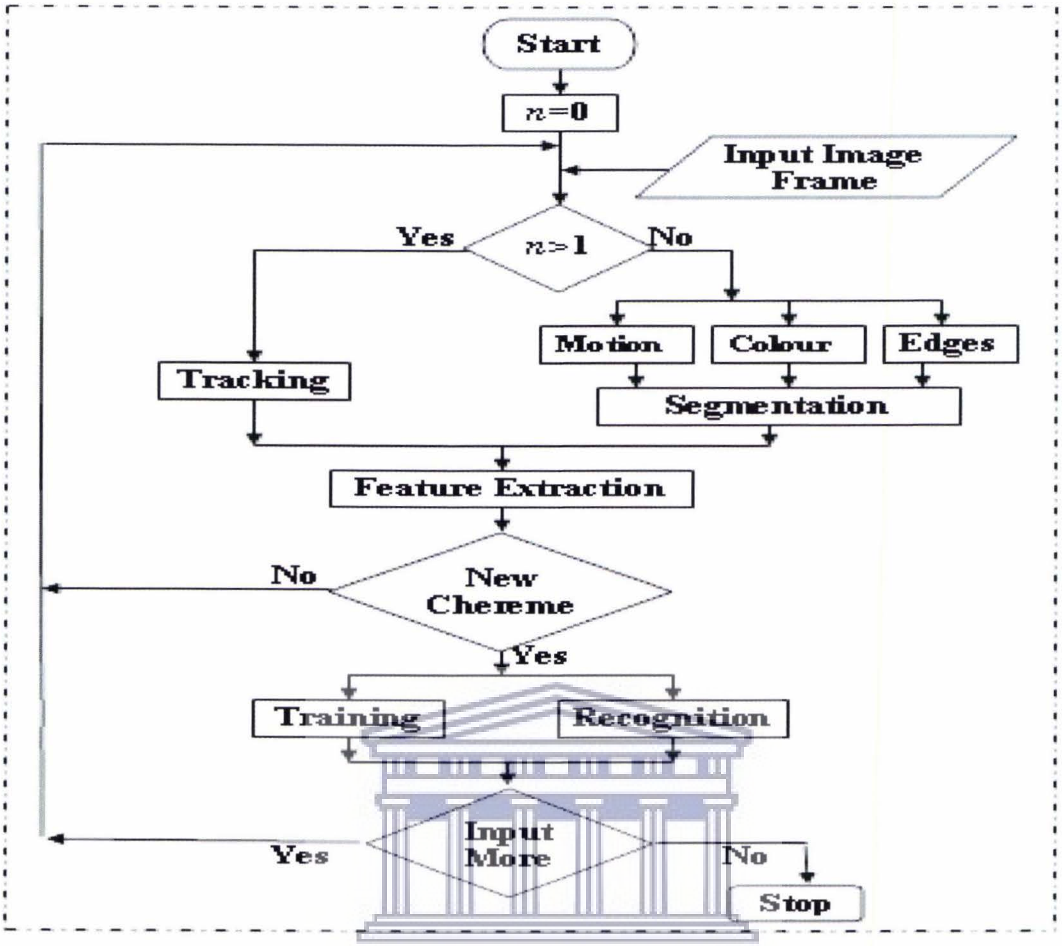


Figure 1.1: Fundamental stages of a chereme based gesture recognition system

UNIVERSITY of the
WESTERN CAPE

Our system represents object motion using displacement vectors along the x and the y directions respectively. The magnitude of motion is obtained by computing the distance between any two centres of mass that are extracted from two consecutive contextually similar blobs. The motion vector is one of the features used in the system for determining the type of movement exhibited by a gesturing hand. We also estimate the shape of the gesturing hand by calculating the approximated ratio of the area covered by the hand region to the area of the minimum bounding rectangle. The system computes the standard deviations and the co-variance matrix which describes the distribution of the skin-coloured pixels along the x, y, and xy directions. Information about the statistical distribution of the skin-coloured pixels is then used to determine

https://etd.uwc.ac.za/

the sizes of the major and minor axes of the best-fit ellipse that bounds the hand region, and/or the angle of inclination of the bounding ellipse. Instead of using the major and minor axes as individual features, we chose to use the ratio of the magnitude of the major axis to the minor axis as this ratio helps to avoid the possible information distortion that would be brought about by the different hand sizes.

We use symbolic notation to represent different cheremes in the system dictionary. For instance, in the system dictionary the three basic cheremes, namely hand shapes, position, and movement types, are represented by different letters of the alphabet. Cheremes that fall within the same class can be distinguished by appending a numerical value to the alphabetical character that represents the chereme class name. These symbols are then used to provide a gesture spelling in a chereme-based gesture dictionary. The new symbols are automatically generated by the system whenever a new chereme is identified. During the gesture recognition stage, the input gestures are first broken down into cheremes. The system then determines the matching set of cheremes that describe one of the gestures which are represented in the system's gesture dictionary. If the level of similarity between the best matching gesture models is above a specific threshold value, then gesture recognition is deemed successful. On the other hand, if the level of similarity is below a specific threshold value, one of the following possible explanations is proffered: either the gesture is not yet represented in the system's gesture dictionary or the gesture recognition module has failed to correctly identify the components of the input gesture.

UNIVERSITY of the
WESTERN CAPE

Incorrect identification of gesture components is normally occasioned by noisy data and/or because the training data samples do not adequately represent all the possible gesture variations.

https://etd.uwc.ac.za/

1.8 Accrued benefits

Besides ensuring effective communication between the hearing-impaired people and those who use spoken languages; technologies used in chereme-based gesture recognition systems can also be adapted for improving the interface devices and the processing modules of some HCI based applications. Technologies used in gesture recognition systems can also be adapted for enhancing tele-robots' perception of their environment, building surveillance systems for securing households against criminal activities and for use in other high security establishments. Many vision-based image processing algorithms, especially those used in the field of medical imaging will greatly benefit from the proposed gesture technologies. Elimination of irrelevant data helps to reduce the amount of information processed by the gesture recognition modules. So far it has proved very difficult to build telephone systems for the deaf because of the large volumes of visual data that should be transmitted from one point to another [27]. The real-time transmission of comprehensive image data from one point to another requires dedicated, high bandwidth channels. Such channels are too expensive to maintain. Only a few rich people can afford the costs of dedicated communication channels. However, deaf people are generally poor and often marginalized; and hence they cannot afford expensive communication devices. If only a small portion of image data, which is just sufficient for recognizing an executed gesture, is transmitted from one point to another through the ordinary telecommunication network, then the amount of bandwidth required for building telephone systems for the deaf would be minimized. Virtual reality based training platforms will also benefit from impressive gesture recognition technologies. Hand based remote controlling of household electronic devices and the remote controlling of heavy machinery used at construction sites may also be enhanced by incorporating gesture recognition technologies. We also propose that gesture recognition technologies should be deployed as human substitutes of air traffic controllers at airports. In short, automatic gesture recognition technologies can be adapted for improving almost all HCI related applications.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

1.9 Novel characteristics

In this research, a new approach to the hand segmentation and tracking problem is proposed. The approach involves fusing motion, skin-colour and edge information in such a way that almost all real boundaries of skin coloured regions are preserved. In most cases, when a segmentation algorithm combines motion with any other visual cues, object boundary information is often lost especially in situations where object motion is too insignificant. Our image segmentation module implements a boundary tracing algorithm which has been designed in such a way that it can trace boundaries of blurred image regions and those of overlapping skin-coloured regions. By combining the skin-colour cues with course motion information [26][28], our segmentation algorithm is able to detect all moving skin-coloured regions without need for human intervention. Some vision-based gesture recognition systems implement histogram based colour models which are obtained by implementing offline skin-colour detection modules. The offline skin-colour detection modules are often trained using manually segmented skin coloured and non-skin coloured training samples. The major limitation of the histogram approach is that in real life situations most users do not have time to train the system on how to isolate skin colours. The following items summarize some of the new features that were incorporated in CRD SLAG.

UNIVERSITY of the
WESTERN CAPE

- *Improved boundary tracing algorithm*

  The new boundary tracing algorithm is designed to trace object boundaries in both clockwise and anticlockwise directions. The new algorithm can easily reorient itself in the event that a dead end is encountered before all boundary pixels are traced. Dead ends are often encountered when the image projection go beyond the image frame boundary. Other forms of dead ends are encountered whenever the algorithm traces thin non-continuous lines that are either caused by image noise or that are induced by some of the image processing modules. Our boundary tracing algorithm

https://etd.uwc.ac.za/

also implements a quasi-deformed model approach for identifying and tracing the outlines of the head region in a cluttered environment.

- *A rigorous multi-cue fusion techniques*

  A two-tier threshold method for detecting object boundaries using motion, edges and skin colour cues is implemented. The above mentioned two-tier threshold method is a condition-based method of fusing image cues that was devised in order to minimize loss of boundary information especially when object motion is very insignificant.

- *A diversified feature set used for isolating and tracking hand regions*

  Whereas most researchers are tempted to use the principal and minor axes of the best fit ellipse that bounds the image blob as two separate feature values, we find it more convenient to use the ratio of these two parameters as it helps to avoid unnecessary distortions that arise whenever the system processes different hand sizes. Besides, we found it necessary to describe the object outlines using the number of points of intersection between the object's boundary line and each one of the horizontal and vertical lines that sub-divide the current image blob into equally spaced sub-regions. The central co-ordinate system of all these sub-regions must coincide with that of the centre of mass of the current blob. This approach helps us to learn the object shape without necessarily studying the correlation between every adjacent pair of object boundary pixels. Many statistical features that are commonly used for describing the region of interest in various computer vision publications [28][42] are also used in this research.

- *Rigorous candidate hand matching algorithm*

  The candidate hand matching process is designed to eliminate all image blobs that fail to match with any other blob within a specific interval of frame sequences. The acceptable inter-frame distance is conditional based. Besides, the candidate hand matching process is designed to ensure that the best matching candidates are

UNIVERSITY of the WESTERN CAPE

https://etd.uwc.ac.za/

identified basing on the level of similarity between the neighbourhood patterns exhibited by the corresponding image blobs [24]. The blob matching process is robust to changing focal views, finger occlusion, or other inter-frame variations.

- *Vision-based extraction of SASL handshape cheremes*

  Most gesture recognition systems overlook the importance of hand-shape information in the whole gesture recognition process. Besides, most image segmentation algorithms are not designed to cope with a myriad of factors that negatively affect the quality of the segmentation results. In a chereme-based gesture recognition system, the quality of the segmentation results largely affects the adequacy of the extracted features, and the gesture recognition process per se.

  Automatic Chereme detection

  The hand-shape and position cheremes are automatically isolated by our system. In order to verify the correctness of each detected hand-shape, we implemented a support vector machine (SVM) based system module that accurately distinguishes between several characteristics of 2D appearance models. Historically SVMs have been used for an identifying object from groups of similarly looking objects.

  Chereme-based gesture spelling

  Particular combinations of letters of the alphabet and some numeric values are used to represent individual cheremes. A chronologically defined sequence of symbolic cheremes serves as a spelling for each gesture in the system's gesture dictionary.

- *Chereme-based sign language database (dictionary)*

  The sign language dictionary will be used for translating sign language gestures into English texts. However, these translations can be extended to include translations from English words into sign language gestures and/or translations from one sign language gesture into a corresponding gesture in a different sign language. The automatic translation of gestures from one sign language into

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

another sign language is important for facilitating communication between deaf people who use different sign languages.

- *Vision and chereme-based sign language recognition system*

  If machines acquire the ability to recognize gestures, then it is possible that if these machines acquire the rules of grammar of each sign language they would also be able to interpret sign languages. We propose that rules of grammar of each sign language should be encoded into the gesture recognition modules.

1.10 Research presentation

This thesis is presented in six chapters. The first chapter gives a general overview of the research. It highlights all the aspects that are covered in the thesis. Chapter 1 also briefly discusses some of the methodologies that are used for solving the gesture recognition problem. Chapter 2 explores the reasons why many researchers are interested in the gesture recognition problem. We also explore the available gesture recognition systems. Besides, chapter 2 also presents the limitations of the available gesture recognition systems. We also define the key term, "gesture", from a sign language linguist's perspective, social perspective and in the context of a specific gesture driven application system. Chapter 2 also reviews the different gesture taxonomies. Knowledge of the different gesture taxonomies helps us to enhance our understanding of the usage and/or interpretation of different gestures.

UNIVERSITY of the
WESTERN CAPE

Chapter 3 reviews the various approaches to the image segmentation problem. This process helps us to borrow ideas from other researchers. Such an exposure helps us to suggest viable solutions to some of the problems that were encountered by other researchers. Chapter 3 also reviews the various segmentation methods, highlighting the strengths and weaknesses of each segmentation method. We also discuss the various ways of identifying connected image blobs. A detailed review of the different approaches to the dynamic gesture recognition problem is also given. The review

https://etd.uwc.ac.za/

covers the commonly used object tracking methods and the relationship between the object tracking and object segmentation processes.

Chapter 4 describes the design of the new chereme-based dynamic SASL gesture recognition system. A prototype system was implemented in order to verify the proposed system design. As part of the first phase of the system design process, we explored how robust hand segmentation can be achieved. Here it was found out that robust hand segmentation can be achieved by devising efficient ways of integrating specific low-level image cues. The chapter also discusses how high-level knowledge of an object is encoded into each resultant bitmap image. Irrelevant bitmap regions are discarded as soon as they are discovered. Features that describe the remaining blobs are extracted; and the extracted features are in turn fed into the tracking and the recognition modules. A probabilistic feature tracking module, which produces one or more lists of similarly textured blobs, is used for aligning related image blob. The validity of each list of aligned blobs is further tested by passing their constituent features into the SVM. Alternatively the produced feature lists are also fed into the HMMs, which are in turn trained to recognize the gesture sequences. However, the HMM based learning module is not yet implemented.

UNIVERSITY of the
WESTERN CAPE

Chapter 5 presents an evaluation of the results produced by our system prototype. The chapter discusses how specific parameter values affect the quality of the segmentation results, and how parameter values that optimize the system performance were chosen. Although we could not compare all our results with the results produced by other systems, we still confirmed the commonly held notion regarding the effects that increasing (decreasing) specific threshold values have on the segmentation results. Vision researchers argue that it is difficult to compare the output of different gesture recognition systems [93][34][104] since different systems are trained on different data samples. Any slight variations in the imaging environment normally affect the output of the used data samples and the overall performance of the image processing algorithm. It can be argued that different systems are always bound to produce different

https://etd.uwc.ac.za/

results since it is almost impossible to create a homogeneous imaging environment for all systems. Hence there is no firm premise upon which an effective comparison of the output of our image segmentation and tracking algorithm against the outputs of other vision-based segmentation and tracking algorithms can be done. Chapter 5 also analyzes the performance and consistency of our object tracking algorithm. The SVM test results for each sequence of aligned blobs gives a relative measure of the consistency of the hand shape recognition process.

Chapter 6 presents the achievements and the shortfalls of this research. In this chapter we also highlight pending issues and recommend future research activities that would enhance the chances of building a successful chereme-based gesture recognition system.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Chapter 2

Gesture and sign language recognition processes

2.1 Background

This chapter explores the gesture notion from different perspectives. We also explain what constitutes a sign language. The chapter also explores how communicative symbols are derived and used in different application and/or social domains. A thorough analysis of the different gesture components and a review of the sign language structure help us to determine the most viable approach to the gesture recognition problem.

Before the nineteenth century, non-signers regarded sign language as an incomplete language. Klima and Bellugi, cited in [101], say that people regarded sign language as a mere collection of vague and loosely defined pictorial gestures that were not bound by any rules of grammar. Klima and Bellugi also say that non-signers believed that the depictive nature of sign language gestures inhibited people's thinking capacity. These misconceptions about sign language eventually saw deaf people being discouraged from using signs. Instead, they were forced to learn lip-reading. Depriving the deaf people of the only language they knew of, negatively affected their ability to communicate and their aptitude to learn from other people [16].

Although linguists and sociolinguists now acknowledge that sign is a full-blown language, which has its own lexicon and rules of grammar [16][7], many people do not understand sign language. Deaf communities are still marginalized since the lack of a common language between the hearing impaired and the speech using peoples is acting as a social barrier that hinders the deaf from fully integrating with the non-signers. A comprehensive sign language recognition system would enable the hearing impaired people to easily communicate with non-signers even in the absence of human

UNIVERSITY of the WESTERN CAPE

https://etd.uwc.ac.za/

interpreters. A comprehensive sign language recognition system should be able to translate grammatically correct sequences of sign language gestures into text or speech, and/or vice-versa [13]. Since sign languages differ from community to community, it is imperative to build gesture recognition systems that translate between different sign languages as such systems will also enable different communities of deaf people to communicate with each other.

Gesture recognition technologies that reduce the amount of visual data that are processed by the gesture recognition modules at each moment in time can also be adapted for building a cheap telephone system for the deaf [27]. Generally, a large bandwidth is required for transmitting image data from one point to another. Using the available data communication equipment, the process of transmitting large chunks of image data from one point to another through a digital communication network slows down or even jams the communication channel altogether. On the other hand, gesture recognition technologies should extract only a minimum set of image data that makes a gesture apparent. Such information normally represents a very small fraction of the overall image data. If adequately designed gesture recognition technologies are adapted for building the telephone system for the deaf, then a significant saving in terms of the bandwidth that is required to transmit gesture information would be achieved. In a deaf telephone system, animation models would be used at each receiving terminal in order to enable the deaf people to visualize the transmitted message.

UNIVERSITY of the
WESTERN CAPE

Inspired by the ease with which people sign, many research activities aimed at producing gesture controlled mechanical and/or electronic devices were undertaken [4][12][15][20]. Xiaojin *at el.* [94] designed a “finger-menu” selection paradigm for a wearable computer. In Xiaojin’s system, a command menu is activated by moving an opened hand in front of a head-mounted camera. Each of the hand’s five fingers represents a menu option, and each menu option is activated by flexing the finger that corresponds to it. The command menu is deactivated by moving the hand out of view of the head-mounted camera.

https://etd.uwc.ac.za/

Automatic gesture recognition technologies can also be adapted for a wide range of HCI based applications. Since hand gestures are executed using the human body as the central coordinate system, no extra cognitive effort is required for ensuring that the body-space, control-space and task-space mappings are effectively handled [36]. Secondly, hand gesture driven control systems allow users to quickly switch between different control functions. The ability to quickly switch between control functions is of paramount importance in situations where emergency responses are a requirement. However, system developers must always ensure that quick responses do not adversely affect the reliability of the system.

Most of the available human-machine interface devices consist of simple mechanical devices such as the mouse, keyboards and joysticks. These devices often limit the speed and the naturalness with which humans interact with computers [10][25]. Nowadays CPUs and memory chips are becoming increasingly smaller and more powerful. In the near future, it is envisaged that all machines would be equipped with several highly specialized CPUs. The improving computing technologies would further expose the limitations of the currently available human-machine interface devices. Most of the available 2D user interfaces do not scale well with the ever increasing processing powers of machines [104]. Inevitably, new human-machine interactions methods should be designed. Unfortunately, conservative users who normally detest being forced to change the way they interact with other people, would find it difficult to adapt to new ways of interacting with machines. However, since gestures are an easy and natural way of interacting, gestures present themselves as the most viable option for enhancing human-machine interaction which system users would not dare to resist. No special skills are required in order to use gestures. Besides, the highly flexible nature of human hands enables the hand to easily assume different spacio-temporal appearances. The different spacio-temporal appearances can be treated as a powerful domain from which complex sets of control instructions may be derived. Gesture driven interface devices can be used as 3D input devices. Although properly designed

UNIVERSITY of the
WESTERN CAPE

gesture driven interfaces are posed to bridge the communication bottlenecks that are inherently associated with the current mechanical interface devices [15], the complexity and efficiency of the information transfer process will largely dependant on the prevailing operational circumstances. In the case of virtual environments, the interface device is sometimes overly obtrusive, awkward or constraining; and this severely degrades the system performance [2].

Before building an automatic gesture recognition system, the researchers must first study how gestures are interpreted or used. The available gesture recognition systems must be reviewed in order to exploit their strengths. Some researchers may also devise ways of overcoming the shortfalls of the current systems. A review of each available system must particularly focus on how information is captured and processed by different gesture recognition systems. It is important to choose the gesture recognition methodologies that best meet the scope of the proposed application. Coarse information about the hand dynamics, which largely contain some of the salient features that best describes each gesture, is often exploited for augmenting the hand segmentation and/or the gesture recognition processes [97]. The dynamics of a gesturing hand are often explained from an analysis of the hand movement patterns.

UNIVERSITY of the
WESTERN CAPE

In this research we devise a framework for building a dynamic gesture recognition system that recognizes a large number of South African Sign Language gestures. Knowledge of the structure of the specific sign language gestures would be encoded into the system in order to improve the gesture recognition process. Subsection 2.2 gives different definitions for the term gesture; and it also presents some of the most popular gesture taxonomies. In particular, subsection 2.2.1 defines the term gesture, while subsection 2.2.2 highlights some the commonly used gesture taxonomies.

https://etd.uwc.ac.za/

2.2 Linguistic and social views about gestures

Gestures are interpreted in the context of their application domain. From a social perspective, groups of people who share a common history and a common culture often develop their own form of sign language. Thus it is obvious that we have different types of gestures, some of which might have the same meaning across different societies. The next two subsections present different gesture definitions and the popular gesture taxonomies.

2.2.1 Gesture definitions

Communication is a way of passing information about, or from the environment around us. Most people communicate through speech, while a small percentage of people communicate through consciously executed body movements, the gestures [16]. According to the Merriam-Webster online dictionary [17], a gesture is a movement of the body parts or limbs that expresses or emphasizes an idea, sentiment, or attitude. Matthew Turk defines a gesture as a synchronized body movement that is intended to convey information or to interact with the environment [2]. Generally gestures are a natural and intuitive way of non-verbal interaction among people. Almost all people use gestures in one way or another. For instance, before human infants learn how to speak, they mainly communicate through gestures [7][4]. The hearing persons use gestures to explain situations which words alone cannot sufficiently describe, or when they are operating under noisy conditions or in under water environments where it is not possible use speech [4][16].

UNIVERSITY of the
WESTERN CAPE

The above given gesture definitions show that gestures are mainly designed to facilitate human to human communications. However, computer vision researchers have now devised a wide range of gesture driven applications. In all these applications, gestures are defined according to their application domain. The increasing numbers of gesture-driven HCI applications have resulted in increasing numbers of gesture taxonomies.

https://etd.uwc.ac.za/

Even within the human to human communication domain, different gesture taxonomies have been proffered [104][20]. In practice new gestures are acquired as people interact with the objects in their environment. Stokoe [16] says that people learn more about the world by manipulating the objects around them. In fact people from different spheres of life often develop gestures as a way of modeling their perception of reality and their societal needs; and hence gestures are mainly iconic in nature and societal based. Gestures used by traders in a market place are different from those used by musicians. Gestures used in military establishments are different from gestures used in sign languages.

In cases where knowledge of the semantics of a gesture can easily be learnt from its syntax, it is recommended to study gestures in the context of a specific sign language. Sign language is a highly structured visual language. In the field of HCI, similarly looking gestures may have different interpretations depending on their application domains. Andrew [4] says that it is very difficult to define a gesture without specifying its application domain. In this research we are mainly interested in exploring the automatic interpretation of dynamic gestures arbitrarily chosen from the South African Sign Language.

UNIVERSITY of the
WESTERN CAPE

2.2.2 Popular gesture taxonomies

In Cohen [20], the term gesticulation refers to all gestures that accompany another language, especially those that accompany spoken languages. In Cohen's work, all gestures that occur independently of speech are called autonomous gestures. Cohen further sub-classifies autonomous gestures into four dichotomies: the acts-symbol; opacity-transparency; semiotic-multisemiotic; and centrifugal-centripetal dichotomies. The act-symbol dichotomy represents some pure action gestures that mimics or symbolizes the behaviour of the described object. Whereas the interpretation of actions that characterize an act gesture is obvious even to non-signers, the meaning of a symbolic gesture is often inferred from the associated actions or is learnt through

https://etd.uwc.ac.za/

exposure to the associated actions. For instances, when someone starts smoking while being questioned by some detectives, such an action is usually presumed to imply that that person is trying to hide something.

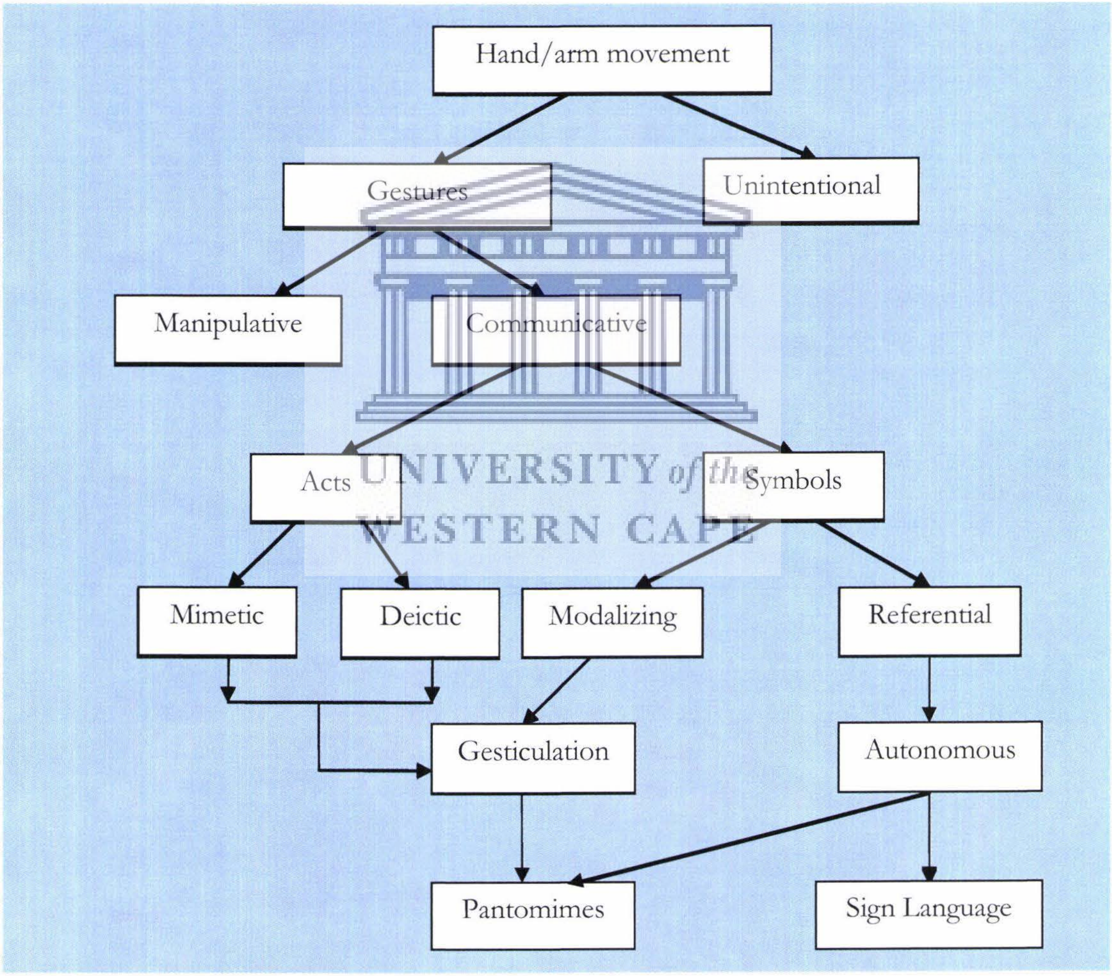
The opacity-transparency dichotomy refers to the easy with which gestures are interpreted. Some gestures are clearly understood across cultures while others are not easy to interpret. In most cases the same gesture may assume different meanings in different cultures. Even within the same culture, non-signers may require additional clues in order for them to interpret certain gestures. Opacity refers to gestures that do not have an obvious meaning. Transparent gestures are easy to interpret, and often they are cross-cultural in nature. However, gestures that have similar meanings across different cultures are not very common since most gestures are opaque in nature. Autonomous semiotic gestures are mainly used in conjunction with sign languages. Multisemiotic gestures are those gestures which accompany other languages, especially oral languages. Centrifugal gestures are designed to invoke a reaction or attention of a specific object, while centripetal gestures are observed and interpreted as moods which are usually not directed at any particular subject [20][96].

In spite of the presence of numerous gesture taxonomies, we find Quek's taxonomy [97] useful for our purposes. Quek's taxonomy also gained wide acceptance from other HCI researchers [15][95][98]. Figure 2.1 presents a modified version of Quek's gesture taxonomy. Although this figure just focuses on dynamic gestures, hand gestures are broadly classified into static and dynamic gestures. According to figure 2.1, dynamic hand gestures are described as intentionally executed hand movements which either have a manipulative or a communicative role. Manipulative gestures are predominantly those gestures which are used for controlling mechanical or electronic devices. A tight relationship between hand movements and hand shape is exploited for building gesture controlled mechanical or electronic devices [96][99]. In gesture recognition systems, it is important to observe and track all symbolic movements [41]. The inadequacies of computer vision algorithms affect the effectiveness with which object tracking is done.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

In most cases the salient features that are associated with each gesturing hand are often lost during tracking [41]. As a result, gestures used in gesture driven human-machine interfaces are not always correctly understood by machines. In order to enhance the performance of gesture driven interfaces, feedback/control loops that monitor whether a manipulative gesture is properly executed and interpreted, are always used in gesture controlled devices. In its simplest form, the feedback loop only consists of the human eye. The human eye verifies the machine's response to an executed gesture, and based on the observed response, the operator decides on whether the gesture was properly understood by the machine or whether the gesture must be repeated.



UNIVERSITY of the WESTERN CAPE

Figure 2.1: Classification of dynamic gestures

https://etd.uwc.ac.za/

Communicative gestures are those gestures whose intention to convey information is paramount, manifest and openly acknowledged [20]. Gesture communication is deemed successful only if one object transmits information which in turn is observed, interpreted and responded to by the other object. Communicative gestures are further subdivided into acts and symbols. Quek [97] says that acts are communicative gestures in which the body movements are directly linked to how a gesture is interpreted. In Quek's works, acts and symbols are defined in a similar way to how they are defined by Cohen [20]. Mimetic gestures are classified under acts, and these often transcend across cultural barriers. For example, in all sign languages the flip flop motion of stretched hands often symbolizes flying. Acts also include the pointing gestures, which are often referred to as deictics. However, some gesture taxonomies classify pointing gestures under the manipulative gesture category.

Deictics allow the natural language speakers to easily describe objects which words alone cannot sufficiently explain [16]. Deictics are frequently used to identify the object(s) that are discussed in a conversation [13]. Mimetic gestures are role-playing actions that depict the behaviour or characteristics of the referenced object or activity [13]. Mimetic gestures are mainly used in pantomimes. The difference between pantomimes and symbolic gestures is easily noticed when non-signers and native signers of a particular sign language are asked to perform the same sign [101]. Pantomimes usually last much longer than the corresponding symbolic gesture. For instance, the symbolic gesture for wheel motion is often represented by moving a stretched forefinger in small circular motions within the chest region. However, in pantomime the same sign can be performed by moving the whole arm in bigger and overly continuous circles. In essence, symbolic gestures are characterized by short and precise movements whereas pantomimes are random in nature. Quek [96] defines a symbol as some kind of motion shorthand. According Nespoulous *et al.* [100], acts eventually evolve into symbolic gestures once unique and generally accepted ways of performing particular mimetic gestures are adapted. Even though symbolic gestures are

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

not always easily understood by non-signers, they however, have an important linguistic role [15]. The next subsection discusses how sign language linguists view hand gestures. We are particularly interested in analyzing the gesture composition and the relationship between a gesture and a sign language.

2.3 Languages, gestures and sign language gestures

In everyday life, human infants, animals and insects use innate signals to demonstrate their desires and to convey their perceptions of the world. These rudimentary ways of communicating do not constitute a language [7]. A language is a highly structured mode of communication which is characterized by clearly defined rules of grammar and a well structured lexicon. Although speech is the default mode of communication for most people, sign language is the only mode of communication for the deaf people. According to Pearl [7], non-signers used to regard speech as the major constituent of any language. Any other form of communication which was not based on speech was not considered a language [18]. This misconception about sign language, together with the then prevalent belief that a language invokes thoughts, led to strong stigmatization of the deaf people [101]. Non-signers also believed that deaf people could not think properly since they had ‘no language’ [7]. However, it is now clearly understood that communication is a physical way of expressing a mental concept. This implies that thoughts precede the ability to express. Thus it can be argued that a language does not necessarily invoke thinking. Besides, sociolinguists and sign language linguists have since found out that sign language is a full-blown language [101][18].

UNIVERSITY of the
WESTERN CAPE

Sign language is a highly structured visual language which is mainly used as the basic mode of communication by the hearing impaired people. Signing people mainly communicate through hands, head or other coordinated body movements. We have already pointed out that not all forms of communication constitute a language. In order for communication to be deemed a language, firstly reality must be identified using finite reusable segments. These segments are often meaningless on their own, yet they

https://etd.uwc.ac.za/

form the basic building blocks of any language. A continuum of experience can be conveyed in a symbolic manner by merely combining the various components of a language according to the rules of that language. For instance, in spoken languages, words are mere variations of amplitudes and frequencies of sound waves that are mainly produced by the vocal chords. Even though sound waves are continuous in nature, some artificial pauses are incorporated in speech in order to allow the listener to distinguish between the spoken words. Linguists even argue that each spoken word is made up of several basic units of sound, the phonemes, which are meaningless on their own [35]. However, different phonemes are combined in various ways, in order to produce various spoken words [7][101]. Just like in speech, Stokoe says that individual gestures are composed of simultaneous combinations of cheremes [16][101]. In Section 1.2, we pointed out that Stokoe identified the hand-shape, the hand position and the hand motion as the basic units that constitute every sign language gesture. In fact all gestures that constitute each particular sign language are built from a small finite set of cheremes, hence it can be argued that each sign language has its own set of reusable cheremes. In each sign language, the reusable set of cheremes is very small when compared to the number of gestures that occur in the sign language per se. A gesturing hand can assume many different postures; and thus creating different hand-shape cheremes that can be used to create a variety of gestures [23].

UNIVERSITY of the
WESTERN CAPE

Early studies on the phonological variations of sign language gestures concentrated on identifying the simultaneous combinations of cheremes rather than analyzing the manner in which the cheremes combine. However, in Bayley at el. [103], sign language gestures are regarded as sequential combinations of different units. Bayley argue that phonological variations in sign language gestures are not necessarily explained from the identity of the simultaneously combined units.

In each sign language, a signer consciously uses hands or other body parts to convey as much information as would be achieved by speech [7]. Sign language sentences are sequences of grammatically connected communicative gestures whose impact on a

https://etd.uwc.ac.za/

native signer are much similar to that achieved by a similar English sentence on an English speaking person. Each sign language has its own rules of grammar and its own vocabulary which, in most instances, is a total inverse of the English grammar. Rules of grammar are the conventionally accepted ways of combining the individual units of a language such that the resultant propositions are always meaningful. The dissimilarity between sign language sentences and English language sentences can also be traced back to the dissimilarity between English words and sign language gestures. In most cases there are no one-to-one mappings between English words and sign language gestures. Every sign language is usually composed of thousands and thousands of gestures. In Chapter 1 we explained that gesture recognition systems that maintain thousands and thousands of individual gesture modules are difficult to build. This phenomenon is caused by the inability of sign language recognition systems to clearly distinguish between minimal pairs of signs which are composed of slightly different hand-shapes [23].

2.4 Gesture recognition systems

UNIVERSITY of the WESTERN CAPE

Most of the available gesture recognition systems focus on learning the sequential combinations of the salient features that are extracted from the gesturing hands. However, such an approach is best suited for building small vocabulary and/or person-dependant gesture recognition systems. We envisage that to achieve comprehensive gesture recognition, the gesture recognition process must analyze both the sequential and the simultaneous combinations of the basic gesture units. The major limitation of the existing gesture recognition systems lies in their inability to sufficiently identify the basic gesture units. This research addresses this problem. The HMM has been successfully adapted for solving the sequential gesture recognition problem [40][3][19][32][45].

A gesture recognition system that interprets complete sign language sentences must incorporate high level knowledge of the syntax of the corresponding sign language

https://etd.uwc.ac.za/

since lexical meaning alone is not enough to interpret sign language sentences [13]. Comprehensive interpretation of sign language is very difficult to achieve. Besides the mere fact that it's not easy to incorporate the rules of grammar into a sign language recognition system, we have also seen that the image segmentation, tracking and the feature extraction processes are too complex to achieve. Software alone hardly achieves real-time processing of images, yet the success of a gesture recognition system largely depends on the efficiency of these gesture processing tasks. Most image processing algorithms fail to preserve the salient features that characterize gesturing hands. In fact, some of the features are lost as the image is transformed from a 3D world view into a 2D model. In practice, the gesture recognition process involves some substantial analysis, and understanding of human actions and behaviours. Human actions and behaviours are influenced by a complex set of variables that include one's cultural background, linguistic capabilities, the conditions under which a gesture is executed and one's level of knowledge about nature.

Even though computing and imaging technologies are rapidly improving, computer vision is still one of the biggest challenges for the gesture recognition problem [49][13]. Computer vision algorithms must be designed to model a myriad of variables; each of which needs close monitoring. As a result, it is almost impossible for vision researchers to produce comprehensive sign language gesture recognition systems. Most researchers tend to investigate a few factors at a time. This research explores ways of enhancing the extraction of cheremes that constitute the South African Sign Language gestures. The extracted cheremes are used for recognizing arbitrarily chosen dynamic SASL gestures.

UNIVERSITY of the
WESTERN CAPE

In this research 2D appearance-based models are used. Using appearance-based models, coarse information about hand dynamics is deduced from an analysis of the centres of mass of the consecutive hand blobs. Special attention is given to slight changes in hand configurations and hand positions. A review of the available gesture recognition systems shows that motion estimation and object tracking are difficult to

https://etd.uwc.ac.za/

achieve since the composition of the tracked features change from time to time as a result of occlusion [41].

Despite the fact that sign language linguists have shown that gestures can be recognized from their basic components, most of the available gesture recognition systems recognize gestures as whole entities. Such gesture recognition systems are incapable of recognizing a large number of gestures. However Quek [97], believe that machine vision technology promises to make substantial gesture recognition a practical reality. The next subsection presents some of the SASL cheremes that were identified in this research.

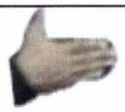
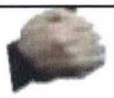
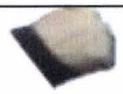
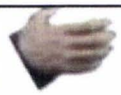
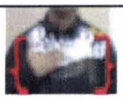
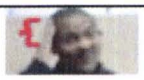
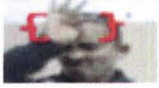
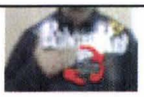
2.5 SASL cheremes

Although gesture cheremes differ from one sign language to another, all cheremes are derived based on the same principals. In each sign language, a set of hand-shape cheremes is a union of all possible hand shapes that has linguistic significance in that particular sign language. In this research, finger configurations help to distinguish between different hand-shape cheremes. Each sign language is associated with specific hand movement patterns. For instance, Table 2.1 presents brief descriptions, some pictorially, of a few SASL cheremes.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Table 2.1: Examples of dynamic SASL gesture cheremes

Chereme Class	Symbolic Name	Description	Pictorial Representation
Hand Shape	H1	Flat palm, all four fingers stretched and extended thumb	
Hand Shape	H2	all fingers in a clenched fist position	
Hand Shape	H3	Fore-finger bend inward at 90^0 angle; and other fingers clenched	
Hand Shape	H4	All fingers loosely bend inwards at about 90^0	
Position	P1	Chest position	
Position	P2	Head region, just above the ear	
Position	P3	Head region; forehead	
Movement	M1	circular motion, e.g. about the chest	
Movement	M2	Side to side (do not know sign)	

UNIVERSITY of the
WESTERN CAPE

This chapter reviewed how gestures are constructed and how different groups of people use gestures. In the next chapter we discuss how visual information is captured and processed by a computer. Different segmentation and image tracking methods are chapter also reviewed in Chapter 3. The chapter also reviews some of the available vision-based gesture recognition systems.

https://etd.uwc.ac.za/

Chapter 3

Image segmentation and tracking

3.1 Brief Overview

This chapter presents a wide collection of the commonly used image segmentation methods. The strengths and weaknesses of these segmentation methods are discussed. These discussions help us to verify the applicability of the presented methods to the hand segmentation problem. It is well known that the quality of the segmentation results largely affect the accuracy of the associated object tracking algorithm. Perfect hand segmentation is difficult to achieve since it is almost practically impossible to devise real-time models that perfectly model all the changes suffered by a gesturing hand. This chapter also presents some of the typical solutions to the hand gesture recognition problem. The KLT feature tracker and other alternative approaches to the object tracking problem are presented in Section 3.4.

3.2 Introduction

UNIVERSITY of the
WESTERN CAPE

In the field of digital image processing, computing devices are expected to classify image data into perceptually or contextually similar regions. In order to successfully recognize an object of interest, vision algorithms must first locate the object and then monitor and track its physical properties and/or its behavior over a specific time-frame. A chereme-based hand gesture recognition system should be designed to monitor the hand-shape, the hand position and the hand movement patterns. The task of locating and tracking gesturing hands in video sequences is effortless for the human eye to perform, but it is far more complex for computer vision machines to achieve. In fact, it is even very difficult for machines to correctly identify the outlines of some of the objects contained in video images. Object outlines helps to delineate an object of interest from the rest of the image data. An adequately delineated object is easier to

https://etd.uwc.ac.za/

track. On the other hand, if the object tracking algorithm is based on predictions of future object positions, then the tracking process indirectly enhances the segmentation algorithm by reducing the search space that is operated on by the segmentation algorithm. A small search space minimizes the number of computations that are realized during image segmentation. As a result, the segmentation process would run faster. According to Cullen [75], image segmentation is one of the most difficult problems that computer vision researchers have to address. The quantity of the segmentation results directly affects the robustness of a gesture recognition system. Perfect segmentation is difficult to achieve because of the following reasons:

- important information is often lost while transforming 3D world views into 2D images,
- some objects are too flexible to the extent that it is difficult for researchers to incorporate information about object shape into the image segmentation algorithm,
- 2D views hardly address the occlusion problem,
- complex object shapes are difficult to model,
- sporadic changes in illumination patterns or camera motions often induce false object information,
- the presence of perceptually or contextually similar objects make it difficult to correctly identify a region of interest,
- gradually changing object illumination patterns make it very difficult to correctly identify suitable segmentation threshold values.

UNIVERSITY of the
WESTERN CAPE

Since perfect image segmentation is hard to achieve, most segmentation algorithms are specifically designed to meet the requirements of particular application systems. In general, the available vision-based hand segmentation algorithms are largely inefficient, imprecise and insufficient for the gesture recognition problem. This chapter reviews some of the underlying issues that affect the quality of the segmentation results. The surveyed literature show that various segmentation and tracking algorithms have been developed [55][1][38]. For each method reviewed in this chapter, we also

https://etd.uwc.ac.za/

highlight the strengths and weaknesses of that particular method. The chapter also reviews how object segmentation and object tracking processes compliment each other.

3.3 Image segmentation

3.3.1 Overview of concept

Segmentation is the process of delineating the object of interest from complex background scenes. In most cases the delineation process is based on perceptual similarity between neighboring pixels [51] and/or knowledge of the structure of the object. Every segmentation algorithm must decide on an effective partitioning criteria and a method for achieving efficient partitioning [64]. Segmentation algorithms that are solely based on perceptual grouping of low-level cues hardly produce acceptable results. It is practically impossible to find a selection criterion that effectively separates all objects of interests from background objects. Segmentation algorithms that depend on exploring the distributions of low-level image cues mainly assume that good partitioning criteria are always easy to find. In practice the foreground is not always sufficiently perceptually different from the background [86], hence perfect segmentation is difficult to achieve since it is difficult to find threshold values that clearly separate the object of interest. If the segmentation process is entirely dependant on factors such as similarity, proximity and good continuation [65], then the probability that the segmentation algorithm will create too many partitions is very high [64]. How then does the image processing algorithm identify the most probable region of interest?

UNIVERSITY of the
WESTERN CAPE

In order to increase the chances of correctly identifying a region of interest, high-level knowledge about the object of interest is often incorporated into the preliminary segmentation results [64]. Shi and Malik [64] argue that the process of partitioning an image should be done from the bigger picture going downwards, just like a painter first marks out the outlines of an object before filling in the details. In the field of computer vision, image boundaries provide useful object information. In fact computer vision researchers argue that objects can be recognized from their crude outlines [55][56]. However, in real life situations, image processing algorithms and/or background noises often distort the object's boundary information. For instance morphological filters, which are mainly designed to mitigate the impact of background noises on the segmented images [24], often produce jagged and/or non-continuous image boundaries [47]. Non-continuous boundaries are difficult if not impossible to trace [86]. Most vision-based applications are designed based on specific assumptions which often oversimplify the segmentation problem. Most of these assumptions are intended to overlook the challenges that are contributed by some of the factors that were mentioned in Section 3.1. The next subsection reviews the different segmentation methods.

3.3.2 Segmentation methods

Segmentation algorithms are mainly classified into region growing techniques, threshold based segmentation and/or edge based segmentation methods [1]. Region growing techniques and threshold based segmentation methods isolate image objects by grouping together regions with similar or slightly varying visual or contextual features [55]. In region growing based segmentation methods, especially the split-and-merge approach, the computational costs of splitting and merging the image into similar regions are often too large [87] such that this approach is hardly used in applications that require real-time processing. In the Seeded Region Growing (SRG) technique, the number of connected components obtainable from each input image is determined by the number of initial seeds used. An initial seed is a predetermined image value to which all image regions with similar characteristics are connected during the image segmentation process. The SRG technique assumes that each connected area meet exactly one seed. Just like the split-and-merge segmentation method, the SRG technique is hardly used in real-time applications since it is not easy to predetermine the quantity and the quality of the initial seeds that guarantees adequate representation and separation of image objects. Despite all this, region growing techniques are very useful for computing the perceptual similarity, the proximity and

UNIVERSITY of the WESTERN CAPE

https://etd.uwc.ac.za/

the continuation of low-level image cues in image data. Other segmentation methods that also classify image objects according to the homogeneity of low-level cues are the threshold based segmentation methods. Threshold based segmentation methods are discussed in the next sub-section.

3.3.2.1 Threshold-based segmentation methods

Threshold-based segmentation methods are mainly used in applications where high processing speeds and automatic processing are a requirement. In the case of dynamic gesture recognition systems, it has been noted that skin colour is a fast and a fairly robust way of segmenting gesturing hands even when the imaging conditions are not quite favourable [79][105]. For instance, skin colour is robust against partial occlusion, changing camera resolutions and/or fluctuating lighting conditions [75][79][47][57]. It was observed that skin colours of people of different ethnic origins are narrowly distributed over certain ranges of the chrominance colour space [41][47][58]. This observation implies that skin colours of people of different ethnic origins have the same chromatic colour properties, and the perceived skin colour differences are mainly attributed to intensity variations [30][41]. In an effort to standardize the image processing problem, computer vision researchers have established various chromatic colour spaces [81][29]. The most commonly used colour spaces are the YCrCb, HSV, RGB and the normalized RGB [47][29]. The HSV colour space achieves better isolation of skin coloured objects from non-skin coloured objects than both the YCrCb and the normalized RGB colour spaces [47][49]. However, it was demonstrated that in the H-S colour plane, skin colour peaks are not always fixed at specific positions [94]. This implies that, in the HSV colour space, it is not possible to build a static skin-colour model for all images. Besides most video data are available in YCrCb format and the process of transforming these videos into other colour spaces is time consuming [47]. The YCrCb colour space is recommended for applications where real-time detection of skin coloured regions is required. However, it has been observed that segmentation results obtained exclusively from skin colour maps cannot adequately

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

identify an object of interest [79]. Skin colour alone fails to locate hand regions when the imaging background contains other skin-coloured objects.

Some vision-based gesture recognition systems use motion for segmenting gesturing hands [28]. In [28] motion maps are obtained by thresholding the differences in intensity values between corresponding pixels that are arbitrarily taken from any two consecutive image frames as shown in the equation below.

$$D(x, y, t) = \begin{cases} 0: \left| D^{1}(x, y, t) \right| < S \\ 1: \left| D^{1}(x, y, t) \right| \geq S \end{cases}$$

The above equation says that the motion, $D_{(x,y,t)}$, occurring around a pixel P(x,y) at a time t, is 0 if the absolute difference in the intensity between any two corresponding pixels, $P(x,y)$ and $P^{1}(x,y)$, that are arbitrarily taken from any two consecutive frames, is less than $(<)$ a given threshold, S. Otherwise the motion is 1 if the difference in intensity is greater than or equal to $(\geq)$ S. Here $D^{1}(x,y,t)$ represents the difference in intensity between any two corresponding pixels. The resultant motion map is sometimes called the difference image or the Motion History Images (MHI) [28]. MHI represent motion as a scalar quantity. Figure 3.1 shows a typical 3D representation of motion maps.

UNIVERSITY of the
WESTERN CAPE

Optical flow methods compute both the magnitude and the direction of motion at each pixel. In general optical flow methods are more computationally expensive to implement than the MHI method. Although it is much easier to compute pixel motion using MHI approach, the resultant motion maps do not necessarily represent object motion. Fluctuating lighting conditions, moving background objects, and/or small camera jerks also contribute to some of the detected motion values. Segmentation results which solely depend on coarse motion detection methods are largely flawed, and hence they significantly compromise the object recognition process [28].

https://etd.uwc.ac.za/

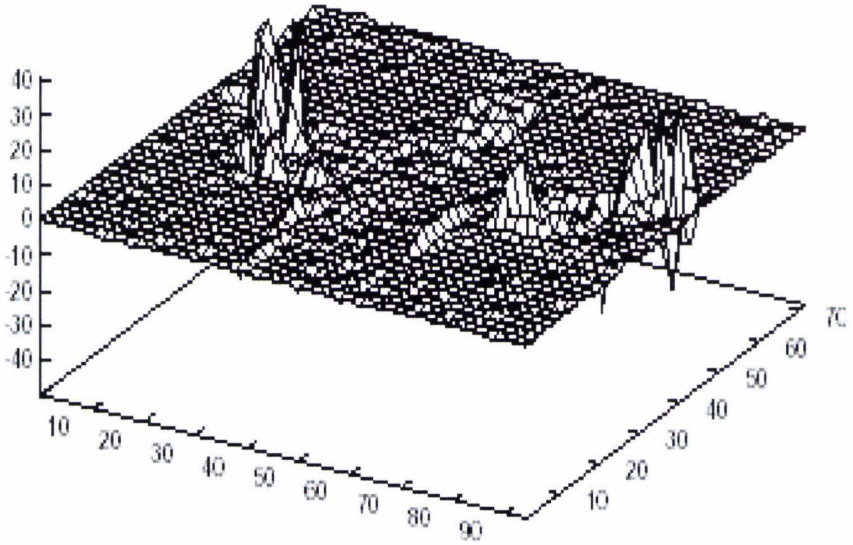


Figure 3.1: A 3D Representation of the difference image: Adapted from Gerhard [28]

The seeded region growing, the split-and-merge and some threshold based segmentation algorithms are examples of similarity based segmentation methods. Similarity based segmentation methods classify objects according to some predefined criteria [76]. These methods use some predetermined partition criteria which are mostly derived from specific low-level image cues. The commonly used image cues include object colour, lighting intensities, motion and texture. In natural images, the concentration of low-level image cues often decrease or increase gradually from region to region. This makes it very difficult to clearly distinguish the boundaries of overlapping and those of similarly looking objects. More robust segmentation results are obtained by fusing different visual cues [22][26][49][52][75][92]. Birchfield [22] argues that in a multi-cue based segmentation, the chosen image cues must complement each other as much as possible as this reduces possible loss of object information. Suat [26] demonstrated that in multi-cue based segmentation methods, the segmentation process is faster if skin colour and motion information are simultaneously processed than when they are processed sequentially. Suat implemented the Bayesian decision theory in order to determine whether an image pixel is skin coloured or not. A better approximation of the pixel colour is obtained by computing the pixel's a-priori colour

UNIVERSITY of the WESTERN CAPE

https://etd.uwc.ac.za/

value as an average value of its 8-nearest neighbourhood pixels. However, the need for computing and averaging several neighbourhood a priori values at each pixel; and the use of the watershed algorithm for determining the object boundaries increases the computational complexity of Suat's algorithm. Besides, the watershed algorithm often produces over-segmented regions that often require further processing.

Suat used the watershed algorithm to label connected components of an image. Once all the connected components are identified, prior knowledge about the object can be applied in order to screen the candidate hand regions. Besides the watershed algorithm, computer vision researchers devised many other algorithms for isolating connected image components. Edge based segmentation methods identify some contour-based edges that most probably represent object outlines, and these are in turn used for isolating the connected blobs in an output image. However, image edges do not always form continuous lines which conform to object boundaries. In order to successfully extract object boundary information, there is a need to devise methods that extract and trace all the weak and the strong edges of an object. Edge-based segmentation methods are discussed in Section 3.3.2.2. Section 3.3.2.3 presents Rosenfield's algorithm for labeling connected components.

UNIVERSITY of the
WESTERN CAPE

3.3.2.2 Edge-based segmentation methods

Contour-based segmentation methods are largely invariant to changes in lighting conditions or object colour [89]. In addition, contours efficiently represent objects with large spatial dimensions. Object contours are closely related to object boundaries, and thus information about the global features of an object, which is often inscribed within the boundaries of each object, can be extracted only after identifying all the contours that are contained in each image [55]. Object contours are mainly extracted by edge-detection operators [76][82]. Image edges provide significant information that assists in peripheral vision and the object recognition process in general [85]. However, as has been explained before, some boundary information is often lost during image

https://etd.uwc.ac.za/

processing and hence the quality of the output image often deteriorates visually [90]. Convectional contour tracing algorithms isolate one image region at a time.

According to Kim *et al.* [88], contour tracing methods are classified into parameterized and non-parameterized methods. The Kalman snake [66] and the adaptive snake models [67] are examples of parameterized contour tracing methods. Parameterized contour tracing methods are mainly used for segmenting non-rigid objects. These methods mainly estimate and represent object features in parametric form. A Kalman snake model is an active contour model in which optical flow methods are used for determining the energy potentials along each contour. Optical flow methods are robust to image clutter. The adaptive snake model automatically adjusts the parameters of the snake model according to the characteristics of the image. The Kalman snake and the adaptive snake models are computer generated curves that are designed to trace the boundaries of non-rigid objects. Their 3D versions are sometimes simply called deformable models or active surfaces. Deformable models are frequently used in medical imaging where it is often very difficult to apply classical segmentation techniques such as edge detection and thresholding methods, which are very sensitive to image noise and the sampling artifacts that are associated with medical images [77]. Deformable models are used when the basic information about the object can perfectly be encoded using a small number of parameters. In essence, parameterized contour tracing methods use compact object data representations and they are ideal for fast real-time applications. However, parametric models cannot adequately segment highly flexible objects which are characterized by frequently changing appearances.

UNIVERSITY of the
WESTERN CAPE

Geometric models are often used for representing object shape. The parameters of a geometrical model can only be computed once the process of deforming an image has been completed. It is, thus very difficult to compute suitable geometrical models for an object which is characterized by continuously changing 2D views. The curves that define a deformable model are obtained by minimizing the total energy required by the internal smoothness forces and the external image forces [78][77]. External forces are

https://etd.uwc.ac.za/

mainly generated from edges, gradient vectors, or texture information derived from image data. Internal forces hold the curves that define the deformable model together, keeping them from bending too much. According to Kass *et al.* [cited in 74], a function that minimizes the sum of internal and external energies is represented as follows:

$$E = E_{\text{int}} + E_{ext} = \int_{\tau \in T} \left(\xi_{\text{int}}(\phi(\tau)) + \xi_{ext}(\phi(\tau))\right)\delta\tau$$

where $\xi_{\text{int}}(\phi(\tau))$ and $\xi_{ext}(\phi(\tau))$ are the internal and the external forces respectively. Although deformable models often produce quality segmentation, the process of computing and matching parameters is time consuming. The computational complexity of a deformable model template increases drastically if the image data contain largely varied texture patterns.

In non-parameterized contour tracing methods, object contours are traced and represented as sets of connected pixel boundaries. Various algorithms for identifying connected components in output images have been reported [68][69][70][71]. Edge based segmentation is one of the earliest segmentation approaches used by computer vision researchers. Edges typically characterize object boundaries but edge information alone cannot sufficiently differentiate between image objects. Image edges do not all always conform to the contours that are apparent to human eye [85]; they merely represent regions of sharp lighting discontinuities [72]. Apparently not all edges represent image boundaries. Some edges are generated by noisy conditions in the image data [71]. On the other hand, if two or more similarly coloured regions overlap, sometimes no edges are detected in the region of overlap even if true edges exist in that region. In order to extract meaningful image boundaries, additional constraints are often required. In some cases, prior knowledge about object shape, colour, and/or texture information are often blended with edge data in order to extract meaningful object boundaries. Object boundary information is useful to algorithms that implement deformable template models [73][74] or statistical pattern recognition methods [42][28]. The major limitation of the edge-based boundary detection methods is that they are not very useful where blurred edges, roof edges or low contrast regions [85]

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

are involved. In addition, Russel and Marina [85] further argue that edge detection algorithms cannot effectively partition an image into regions of interest or clusters of information that fit with a specific semantic. The task of tracing object contours is somewhat a more difficult problem to solve [56][73]. Various edge detection algorithms are discussed in the following sections.

3.3.2.2.1 Edge detection operators

The Sobel, Robert's Crossing, Prewitt and the Laplacian operators are the most popular gradient based edge detectors [82][76][83]. Gradient based edge detection methods assume that every sharp change in image brightness represents an image edge. Accordingly, most edge detectors identify edges as local maxima of first-order derivatives of a pixel-wise brightness function, $f(x,y)$.

In practice the input image is represented as a set of discrete pixels, so the first derivative of the brightness function, $\Delta f(x, y)$, is often approximated by a convolution of the grayscale image with one or more linear masks. In general, the first derivative of the image brightness function, $f(x,y)$, at a pixel P which is surrounded by a group of pixels, p, is approximated as shown in the equation below:

UNIVERSITY of the
WESTERN CAPE

$$\Delta f(x, y) = \partial_x f(x, y) + \partial_y f(x, y) \approx \sum_{k=-1}^{1} \sum_{j=-1}^{1} K(j, k)\, p(x - j, y - k)$$

where $K(j,k)$ is the convolution mask (sometimes referred to as the kernel) and $p(x\text{-}j,y\text{-}k)$ are the neighbourhood pixels of any arbitrary chosen pixel, $P(x,y)$. In simplest terms, the first derivate of the intensity function at any given pixel is approximated by a dot product of a group of grayscale image pixels, p, and a linear matrix. Equations 3.1 and 3.2 presents the two convolution masks that are used in the Robert's Cross operator to approximate the gradient of the image brightness function along the x and y directions respectively.

$$g_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3.1}$$

https://etd.uwc.ac.za/

$$g_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad (3.2)$$

The Robert's Cross operator has the simplest convolution mask but is very sensitive to noise. A Prewitt detector uses a pair of 3x3 convolution masks to approximate the x and y derivatives of the image brightness function. The Prewitt masks for detecting vertical and horizontal edges are shown in equations 3.3 and 3.4.

$$g_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad (3.3)$$

$$g_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad (3.4)$$

The Sobel edge detector is rather more robust than the Prewitt edge detector. The commonly used kernels for a Sobel detector are given in equations 3.5 and 3.6.

$$g_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad (3.5)$$

$$g_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad (3.6)$$

UNIVERSITY of the
WESTERN CAPE

Once the horizontal and the vertical components of the edge are obtained, the magnitude of the edge is computed as shown in equation 3.7.

$$g = \sqrt{g_x^2 + g_y^2} \qquad (3.7)$$

In order to reduce the computational complexity of equation 3.7, the magnitude of an edge is often approximated by summing the absolute values of the horizontal and the vertical components of each edge (see equation 3.8).

$$g = \left|g_x\right| + \left|g_y\right| \qquad (3.8)$$

https://etd.uwc.ac.za/

For instance, using the Sobel operators, the difference in intensity, G, between an arbitrarily chosen image pixel P_5, and one or more of its Moore neighbourhood pixels P_1; P_1; P_2; P_3; P_4; P_6; P_7; P_8 or P_9 is computed as follows:

$$G = |(P_1 + 2*P_2 + P_3) - (P_7 + 2*P_8 + P_9)| + |(P_3 + 2*P_6 + P_9) - (P_1 + 2*P_4 + P_7)|$$

The sensitivity of an edge detector often depends on the size of the convolution mask and the threshold values used [82]. A large sized mask is less sensitive to noise but it also suppresses the weak edges. A small sized mask can detect almost all edges but is very sensitive to noise. Often an edge detector must make a trade-off between the sensitivity and the accuracy. The problem of image noise is sometimes mitigated by smoothening the image data with a Gaussian filter just before the whole image processing is initiated. Edge detectors which use masks that combine the image smoothening and edge detection functions are best suited for detecting weak edges under noisy background conditions. For instance, the Laplacian operator implements a composite edge detection function. This operator uses a single 5x5 mask that approximates the second derivative of the image brightness function. Laplacian operators are useful for extracting edges that are represented by a sudden increase or decrease in intensity (see Figure 3.2). The fact that the Laplacian operator uses only a single 5x5 mask makes it less expensive to implement. However, its major limitation is that it is very sensitive to noise.

UNIVERSITY of the
WESTERN CAPE

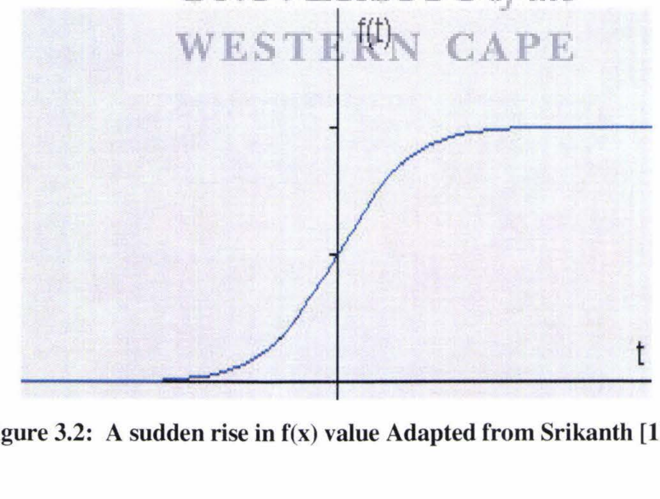


Figure 3.2: A sudden rise in f(x) value Adapted from Srikanth [106]

https://etd.uwc.ac.za/

The Canny operator is based on a very similar kernel to that used in the Sobel detector. However, the Canny kernel incorporates an image smoothening mask. Although the Canny detector is more computationally expensive than the Sobel detector, it however, produces high quality edges. A Canny detector controls the thickness of the detected edge; while its smoothening module helps to discard some of the image noise. The Canny edge detector implements a hysteresis based thresholding approach that suppresses some of the irrelevant edges.

3.3.2.3 Rosenfield algorithm for identifying connected components

In Section 3.2.2.1 we mentioned that object-boundary information and/or connected component labeling techniques help to improve the object segmentation process. In vision based dynamic gesture recognition systems, low-level image cues are traditionally used for segmenting gesturing hands. Subsection 3.2.2.1 also explained that it is not always easy to find a suitable partition criterion that effectively isolates the objects in an image. Often high level knowledge about the object is blended with low level image cues. This section reviews various versions of the Rosenfield algorithm for labeling the connected image blobs.

UNIVERSITY of the
WESTERN CAPE

According to the Rosenfield algorithm [70], an image is scanned twice in order to identify the connected pixels. In the first pass, the algorithm determines whether any of the four neighbourhood pixels; $P_{(x-1,y)}$, $P_{(x-1,y-1)}$, $P_{(x,y-1)}$ and $P_{(x+1,y-1)}$, see Figure 4.2a, of a newly found coloured pixel, $P_{(x,y)}$, is labeled. If some of $P_{(x,y)}$'s neighbours are labeled, $P_{(x,y)}$ is assigned the smallest label, v_k, which is carried by one of $P_{(x,y)}$'s neighbours. However, if none of its neighbouring pixels is labeled, $P_{(x,y)}$ is given a new label, v_{k+1}. In the second pass, the algorithm reassigns a single label value to all neighbouring pixels that were otherwise assigned different values in the first pass. Rosenfield's method requires two arrays, one for storing different labels, $v_1, v_2 \ldots v_k$, where v_k is a positive integer, and the other one for storing labeled pixels, $P_{(i,j)}$, $1 \leq i \leq m$ and $1 \leq j \leq n$; where m and n respectively represent the x and y coordinates of each of the image

https://etd.uwc.ac.za/

pixels that carry the same label values. According to the Floyd-Warshall's (F-W) algorithm [70], if an image region has K labels, most of which are equivalent, the worst-case complexity for the second pass of the Rosenfield algorithm will be at most $O(K^3)$. In this case, large values of K would slow down the run-time of the Rosenfield algorithm. Park *et al.* [68] used a divide and conquer method for resolving the equivalence matrix described in Rosenfield's algorithm [70]. This approach greatly improved the performance of the Rosenfield algorithm. In Park's method, an image is first partitioned into many sub-images. The coloured pixels from each sub-image are then labeled using a similar method to that used for assigning labels in the first pass of the Rosenfield algorithm. The resolution of the equivalence matrix is faster if smaller sub-image sizes are used. After resolving the equivalences in each partition, the connectivity between partitions is established by analyzing the labels of partition boundary pixels. If the label of a partition boundary pixel is greater than zero, then the adjoining partitions are deemed connected and are reassigned a common label value. Experimental results showed that the time required to resolve equivalence labels in image data is inversely proportional to the number of partitions used [68]. However, the process of splitting an image into a number of quadtrees and later on re-merging similar subregions from each quadtree is both complex and time-consuming.

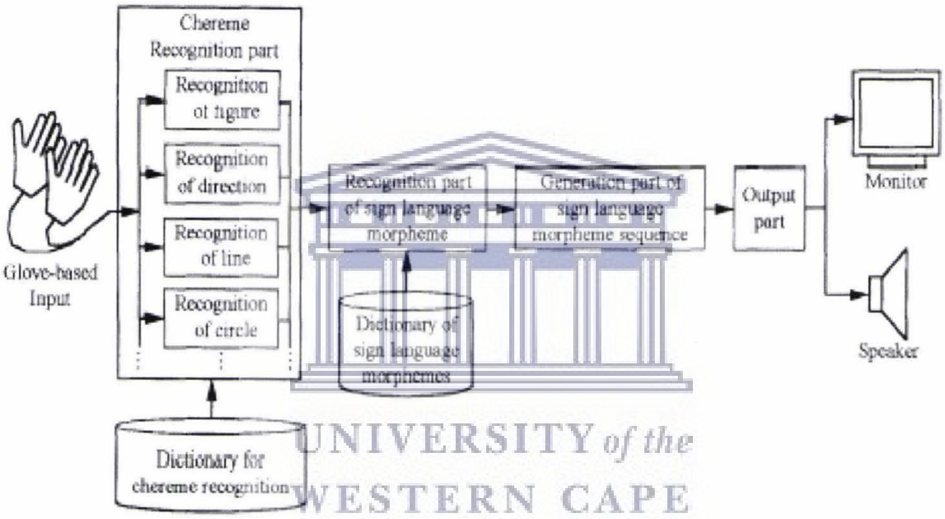
UNIVERSITY of the
WESTERN CAPE

3.3.3 A review of hand gesture recognition methods

Despite, the crucial role that hand gestures may render to HCI applications, it is not easy for vision machines to extract meaningful information about hand gestures. Automatic gesture recognition can only be achieved if machines are taught how to measure and interpret the dynamic and/or static configurations of the hand or other human body parts [15]. The manner in which image data is gathered has a direct bearing on the manner in which that data is processed. Basically gesture recognition systems either exploit vision or mechanical glove-based methods of capturing gestures. Mechanical-gloves are glove-like devices that are fitted with sensors that measure the joints' angles of inflexion, the spatial position and the direction of motion assumed by a

https://etd.uwc.ac.za/

gesturing hand. In glove-based devices, each sensor is connected to a computer via a network of cables or tubes [5][8][25]. For instance, Figure 3.3 is a schematic representation of a cyber-glove-based Japanese Sign Language recognition system [5][43]. In this system, a separate recognition module was created for recognizing each chereme. A list of the probable cheremes that are detected by the chereme recognition modules are then compared with the sign language morphemes that are kept in a system morpheme dictionary. Once the best matching morphemes are found, an optimum sequence of sign language morphemes is generated, and this sequence is in turn used to interpret the executed Japanese sign.


Chereme Recognition part
Recognition of figure
Recognition of direction
Recognition of line
Recognition of circle
Glove-based Input
Dictionary for chereme recognition
Recognition part of sign language morpheme
Dictionary of sign language morphemes
Generation part of sign language morpheme sequence
Output part
Monitor
Speaker

UNIVERSITY of the
WESTERN CAPE

Figure 3.3: Glove-based Japanese Sign language recognition system: Adapted from Sagawa [5]

Different types of electro-magnetic gloves have been invented. A “Sayre” glove [25] is fitted with flexible tubes which are connected to a light source at one end, and a photocell at the other end. Just like the “Sayre” glove, the VPL Data-Glove [25] consists of optical fibre based sensors which are planted along the back of each finger. A measure of the amount of light energy transmitted through the tube at each instant of time is used as a measure of the amount of flexion suffered by a particular finger and/or hand joint. The VPL Data-Glove was one of the most successful gloves [25].

https://etd.uwc.ac.za/

In most instances, glove-based gesture recognition systems are very sensitive to both finger motion and image noise. Sensor gloves can easily detect and quantify even very small fingertip motions. Their highly sensitive nature makes them ideal for use in virtual reality environments. However, glove-based devices are generally cumbersome to wear [25][15]. The discomfort that these devices bring into the human hand negatively affects the naturalness with which people sign [15][4]. Besides, glove based devices are too sensitive to electromagnetic noises to the extent that their performance is severely degraded when deployed in electromagnetically charged environments. Furthermore some users simply detest using wired gloves for health reasons [4].

Vision-based gesture recognition systems theoretically possess several advantages over glove-based systems. First and foremost, vision-based gesture recognition methods do not necessarily impel a signing person to wear the obtrusive gloves which, in most cases, hinder the naturalness with which people sign. Driven by the need to fully adapt the naturalness with which people sign, many computer vision researchers started exploring the possibilities of building vision-based gesture recognition systems and/or gesture driven HCI interface devices. However, as mentioned in Section 2.3.1, computer vision remains the major obstacle to achieving successful vision based gesture recognition systems.

UNIVERSITY of the
WESTERN CAPE

In order to extract gesture information, hand regions must be properly segmented from background objects. In practice, it is extremely difficult to achieve meaningful hand segmentation [54][87]. Several factors contribute to the complexity of the hand segmentation problem. Firstly, the highly flexible nature of the human hand gives rise to a diverse range of hand configurations [91][37]. Each hand configuration may be used to compose different hand gestures. It is very difficult to create an individual model of each sign language gesture. Secondly, while performing a gesture, the gesturing hand may assume different spacio-temporal appearances. Changes in hand appearances make it very difficult to track consecutive hand-blobs. Somehow, hand tracking simplifies the hand segmentation process by drastically reducing the object

https://etd.uwc.ac.za/

search space [52]. Thirdly, occlusion of the gesturing hands and/or the speed with which a gesture is executed makes it difficult to clearly identify the boundaries of the hand region [102]. Fourthly, important information about hand configuration is often lost during the transformation of a 3D hand views into a 2D views [51]. 2D projections of moving hand are often ambiguous [53]. They do not give enough hand-shape information. Other low-level visual cues also fail to unambiguously locate the gesturing hand in the image. However, motion detection methods, especially the background subtraction approach often help produce a relatively short list of candidate hand regions [54].

It should be noted that the solution of most of the above mentioned problems are still beyond the reach of the available computer vision algorithms. Despite the recent advances in computing and imaging technology, computer vision has not yet advanced enough to allow perfect hand segmentation and tracking [49]. In fact no pervasive segmentation or tracking algorithms have been presented. However, a sizable number of domain specific algorithms that work well under constrained environments have been produced.

In order to avoid some of the difficulties that are associated with the colour based hand segmentation algorithms, early vision-based hand detection systems used distinctively coloured gloves. For instance Thad [19] built a Hidden Markov Model based American Sign Language (ASL) recognition system which used coloured gloves for locating and tracking hand regions. In Thad's system, the gesturer wore a yellow glove on the right hand and an orange glove on the left hand and sat on a chair facing the camera. While searching for a hand region, the system scanned the image until it finds a yellow or orange pixel. The newly found yellow or orange pixel is then used as a seed to which similarly coloured neighbourhood pixels are aggregated. Different sets of connected yellow and/or orange pixels are assumed to represent the two gesturing hands. The centroid, and other features that describe each coloured bitmap region are calculated and passed into the HMM. Similarly Hienz *et al.* [91] detected a gesturing hand using a

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

colour coded glove and some coloured markers which were strapped around the elbow and shoulder regions. Each finger had its own colour, and the palm and the back of the hand were also marked with different colours. The hand detection process proceeded in the same way as in Thad's work [19]. Although coloured gloves simplify the hand detection process, it would be unnatural to expect people to always wear coloured gloves before they sign.

Most hand gesture recognition systems use low level image cues to locate and estimate hand poses [49][52][54][87]. In Yuanxin *et al.* [49], skin colour and coarse image motion are used for detecting and segmenting the gesturing hands. The skin coloured pixels that are contained in the input image are sampled out by implementing a histogram based look-up table. Yuanxin *et al.* say that they managed to track continuous hand gestures in real-time using a single camera under uncontrolled background conditions and/or irregular lightening conditions. High level hand-shape information was blended into the image motion maps in order to achieve better segmentation results. Awad *et al.* [52] segmented gesturing hands by logically combining skin colour, motion and position information. In their work, Awad *et al.* identified skin coloured image pixels using a Support Vector Machine based colour distance metric that somehow incorporated some prior knowledge of the previously segmented object. Motion is computed basing on the intensity difference between successive pairs of corresponding skin-coloured image pixels. Awad *et al.* assumed a very small inter-pixel displacement, and as such, the centres of mass of successive hand blobs were estimated using Kalman filters [44].

UNIVERSITY of the WESTERN CAPE

Jonathan *et al.* [54] segmented gesturing hands using a combination of motion and skin colour information extracted from successive image frames. At first a generic skin-colour histogram is used to locate a person's face. Once the face is localized, an adaptive skin-colour model is then used for identifying the most probable skin regions. The motion detector exploits the successive differences in lighting intensities between corresponding image pixels. A large intensity difference is assumed to represent a

https://etd.uwc.ac.za/

moving object or a moving point, while a small intensity difference is assumed to represent the image background. In binary images, motion thresholding techniques often assigns a value of 1 to all motion values which are greater than or equal to a given threshold, while those values which are smaller than the given threshold are assigned a value of 0. A skin coloured pixel is either discarded or kept for further analysis depending on the magnitude of motion it experienced. All moving skin-coloured blobs are segmented in this way. All those skin-coloured blobs that are distributed over a significantly small surface area are eliminated. The remaining image blobs constitute the candidate hand regions. In Jonathan *et al.* [54], blob tracking is achieved by matching each candidate hand region to one of the aligned sequences of image blobs that are drawn from the previous image frames.

In Section 3.3.2.1 we briefly described how Suat [26] segmented gesturing hands using skin colour and motion cues. It is important to note that Suat improved the quality of the hand segmentation results by fusing low level information with the boundary information that was obtained through the watershed algorithm. Aditya *et al.* [92] located and tracked hand regions using a Kalman filter based contour tracing method which works in a similar way to the one developed by Isard and Blake [93]. The hand outlines are represented in the form of B-splines. B-splines ensure continuity and compactness of the represented curves.

UNIVERSITY of the
WESTERN CAPE

Several vision based hand segmentation methods have been reported in literature [28][23][24][94]. However, most of the hand segmentation approaches described by these reports are slight variations of the approaches that were reviewed in this section. Sometimes the choice of a particular segmentation method largely depends on the tracking method used in the application. The next section reviews some of the frequently used hand tracking methods.

https://etd.uwc.ac.za/

3.4 Tracking

When tracking objects, the tracking algorithm must first determine the location of the object in each image frame. A comprehensive object search is either done in each of the consecutive frames or in some cases it is only carried out for the first frame. If the object position in the previous frame is known, the current object position can be estimated from the intensity differences between the two frames [51]. In most cases, the difference image is used for computing the probable changes in object positions after which pairs of perceptually similar regions are aligned. In some way, the aligned sequences of perceptually similar regions represent the object tracking process. The acceptability of a segmentation method depends primarily on the application for which it is designed. According to Awad *et al.* [52], accurate segmentation enhances object tracking, and in cases where the tracking system is based on prediction of movement dynamics, tracking reduces the object search space and hence simplifies the segmentation process. In an attempt to simply the otherwise impossible task of tracking objects from videos, most tracking algorithms impose specific constraints on object motion or on the general imaging environment. The object models and the feature sets used during object tracking largely depend on the objectives that should be achieved by each application [51]. The commonly used object models are point tracking [38][92], geometrical shape fitting [19] and appearance based models [97][26][98]. Appearance based models are a broad classification of models which also include statistical models [42][28]. In practice, researchers often combine different models in order to enhance the performance of a tracking or a segmentation algorithm [57].

UNIVERSITY of the
WESTERN CAPE

The KLT feature tracker [38] was developed on the assumption that the inter-frame displacements are small enough to allow image motion to be modeled using linear transformations. In this case a translation vector would be used to model image motion instead of using the complex affine model. In the KLT feature tracker, it is important to identify image features which are good for tracking. Using optical flow based methods, all image pixels that are characterized by a high standard deviation of the spatial

https://etd.uwc.ac.za/

intensity, Laplacian edges or those pixels that are corner based [96] are often regarded as interesting to track. However, it is difficult to individually track each interesting pixel since pixel values are often abruptly changed by image noise, and hence each image pixel can easily be confused with any one of its neighbourhood pixels. Instead, small sized windows that enclose many interesting pixels are used for tracking object through a series of image frames [95]. The KLT feature tracker also implement a residue monitoring technique that helps to minimize the chances of aligning unrelated tracking windows. A more robust tracking algorithm often utilizes both the affine model and the linear translation vectors as this combination ensures effective representation of all image pixels that probably move at different velocities. During tracking, the feature tracker identifies new features that may be used to substitute lost features in the tracked window. The KLT algorithm is robust against occlusion and is useful for tracking rigid 3D object, but it is not best suited for tracking rapidly changing hand configurations [51].

Many tracking algorithms are designed to track rigid objects [75]. However, these algorithms dismally fail to track highly flexible objects such as the human hand, whose motion patterns and 3D appearances are also quite unpredictable. The gesturing hand often assumes different spacio-temporal appearances [49][57]. Some boundary information of a gesturing hand is often lost due to the blurring effects or background noise, hence it is sometimes difficult to correctly identify an object's contours [87][90][98].

UNIVERSITY of the
WESTERN CAPE

Most object tracking algorithms implement appearance based models [57][54][98]. 3D models are not suitable for the tracking problem since they are computationally expensive to implement, and therefore they are not suitable for applications which require real-time processing. Besides, 3D models often fail to timeously update model parameters, and this greatly compromises the object tracking process [49]. In appearance based models, all 3D real world views are projected onto 2D planes [98]. However, 2D images cannot sufficiently capture non-linear motion hence hand-shape

https://etd.uwc.ac.za/

information is often lost [53]. Loss of important object information negatively impact on the object segmentation process. According to Jonathan *et al*. [54] tracking typically fails in the absence of a perfect segmentation. Inadequate segmentation results give rise to inconsistent image statistics. In algorithms that implement probabilistic blob matching approaches [87][54], inconsistent object information often lead to the establishment of incorrect matches and thus greatly compromising the quality of the tracking results.

3.5 Conclusion

Despite the fact that image segmentation and tracking are widely researched domains, perfect segmentation is difficult to achieve. Available hand gesture recognition systems mainly implement multi-cue based threshold methods. Even the multi-cue approach often fails to adequately segment the hand regions and hence hand-shape information is sometimes approximated using coarse models. On the other hand, the success of a chereme-based gesture recognition system largely depends on the quality of the segmentation results. In Chapter 4 we explain the design and implementation of an improved segmentation algorithm that enables the extraction of cheremes from SASL gestures.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Chapter 4

Segmenting and tracking gesturing hand regions

4.1 Introduction

Perfect hand segmentation helps to preserve important hand-shape information. In Chapter 3 it was pointed out that good segmentation results enhances the hand tracking process. However, the task of locating and segmenting the hand from the rest of the image data is very difficult and sometimes impossible to accomplish. In an attempt to simplify the segmentation problem, some of the available vision based gesture recognition systems were designed to work under controlled background conditions. Some gesture recognition systems assume a stationary background and distinctive foreground colours, while others entirely ignore the significance of detailed hand-shape information to the gesture recognition process [19]. In such instances, hand configurations are often roughly approximated using coarse geometrical models. For instance, Thad [19] describes a hand region by using the best fit ellipse that circumscribes the region.

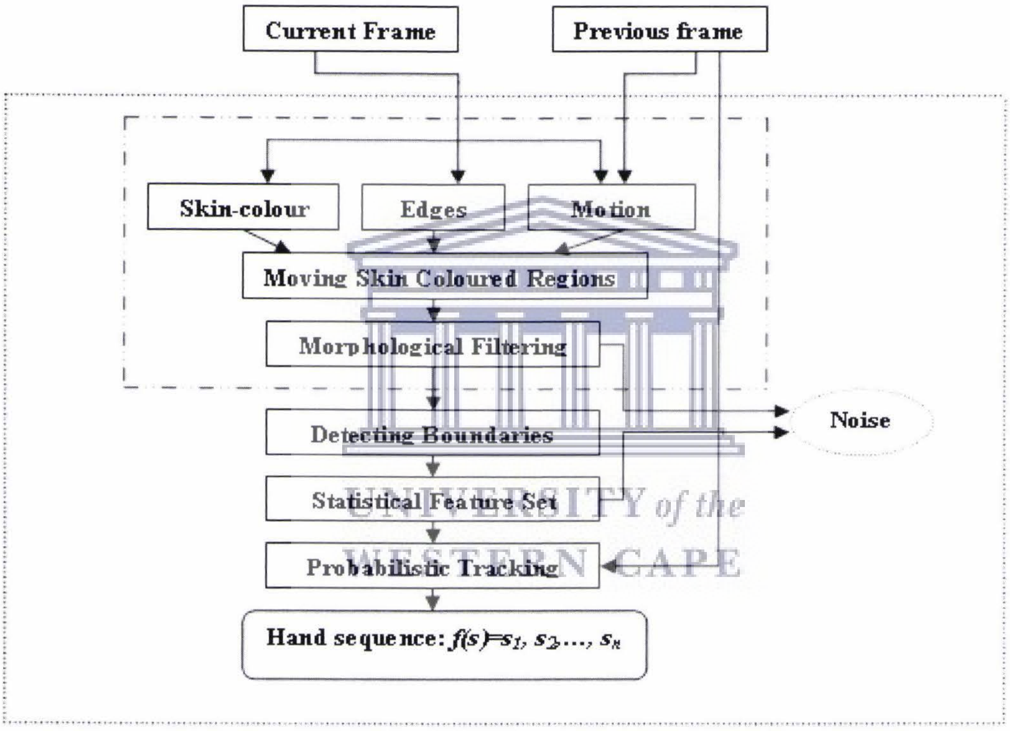
UNIVERSITY of the
WESTERN CAPE

In this chapter, we present our hand segmentation and tracking algorithm. In order to improve the robustness of our segmentation algorithm to rapidly changing speeds of the gesturing hand, we implement a two-tier thresholding method that helps to determine permissible values of moving skin-coloured regions. In our algorithm motion, edges and skin colour cues are fused in such a way that boundaries of slow moving skin coloured objects are always preserved. The chosen threshold values must ensure that the impact of background noises on the overall segmentation results is minimized. If the object boundaries are preserved, it is possible to trace the crude object outlines using a boundary tracing algorithm. Computer vision researchers argue that objects can be recognized from their crude outlines [55][56]. Besides, boundary information helps to confine the space from which a statistical feature set that describes

https://etd.uwc.ac.za/

a segmented region would be extracted. During the hand tracking process, the feature set is used for aligning the candidate hand regions from the current image frame to the candidate hand regions from the previous sets of image frames.

Figure 4.1 explains how cheremes that describe each dynamic gesture are extracted. This diagram also depicts how the hand segmentation and tracking processes are designed in this research. Section 4.2 explains how skin-colour, edges and motion information are used for isolating candidate hand regions.



UNIVERSITY of the
WESTERN CAPE

Figure 4.1: Extracting cheremes from visual images

4.2 Integrating motion, skin-colour and edge information

Image edges are useful for identifying the probable object outlines. A review of edge detection algorithms was given in Section 3.3.2.2.1. Although the Canny algorithm is considered the most optimal edge detector, its computational complexity is much

https://etd.uwc.ac.za/

higher than that of the Sobel edge detector. On the other hand, the Sobel edge detector is too sensitive to the background noises. The Sobel detector frequently highlights background noises as if they were an important component of the image. Despite the above mentioned limitations, the Sobel detector is implemented in this research as a preliminary step towards determining object boundaries. Using the Sobel detector, the magnitude of an image edge at each pixel is computed as shown in equation 3.9. A morphological filter is then used for regularizing image edges. The edge regularization process involves deleting all the weak isolated edges which are mainly caused by image noise. As part of the regularization process, an image is first subdivided into 3x3 pixel sub-regions. The algorithm then determines the total number of edge-based pixels in each sub-region. The strength of the edge that passes through each sub-region is determined by the number of edge-based pixels that are contained in that particular sub-region. A detailed explanation about how our morphological filter is implemented is given Section 4.3.

The Sobel edge detector normally produces a mixture of weak and strong edges. In most cases the detected edges do not conform to continuous curves that characterize object boundaries. Although it is easier for the human eye to tell the object outlines from the extracted image edges; it is far more complex for machines to achieve the same level of perception. In vision-based systems, edge data is often blended with other low-level image cues in order to successfully determine object boundaries. For instance, in this research we combine motion, skin-colour and edge information in order to identify the most probable hand and head regions. The intensity difference between successive image frame pairs is used as a coarse estimate of motion values.

UNIVERSITY of the
WESTERN CAPE

Various methods of detecting and extracting skin-coloured regions have been reported in literature [81][105][29]. However, despite the fact that it has been established that the HSV colour space is more distinctive and less sensitive to colour changes than both the YCrCb and the normalized RGB colour spaces [105][81][59] [29], the YCrCb colour space is chosen for this research for the following reasons: Firstly, most videos

https://etd.uwc.ac.za/

are available in the YCrCb format and the process of converting videos from the YCrCb format into any other format is computationally expensive [47]. Secondly, specific ranges of the chrominance component of the YCrCb colour space are capable of effectively isolating skin-colours of people of different ethnicity [47]. In this research, Cr values ranging from 128 to 173 and Cb values ranging from 77 to 132 are used for segmenting the skin-coloured regions of the input images (see Section 5.3). Gesture recognition systems which implement unsupervised hand detection and segmentation techniques are better suited for the automatic gesture recognition problem than those which depends on manual selection of specific image parameter values.

The difference in intensity between corresponding image pixels that are respectively extracted from the current and the previous image frames is used to estimate object motion. Generally coarse motion values are computed as shown in equation 4.1.

$$D(x, y, t) = \begin{cases} 0 : \left| I_1(x, y, t) - I_2(x, y, t^1) \right| < T \\ 1 : \left| I_1(x, y, t) - I_2(x, y, t^1) \right| \geq T \end{cases} \qquad (4.1)$$

In the above equation, $D(x,y,t)$ represents coarse pixel motion values, which are computed as the absolute value of the intensity difference between any two corresponding pixels $P(x,y,t)$ and $P(x,y,t^l)$ arbitrarily selected from any two consecutive frames I_1 and I_2. The variables t and t^l, respectively represent the different times at which the image frames I_1 and I_2 were observed. Different points on the same image often move at different speeds; hence the intensity differences, $D^l(x,y,t)$, is a set of discrete integer values, Z, that ranges between -254 to +254.

$$D^l(x, y, t) \in Z : Z = [-254, 254]$$

The permissible set of pixel motion values, $D(x,y,t)$, are given as the absolute value of $D^l(x,y,t)$, which ranges from 0 to 254. In order to decide on which motion values are considered significant, this research implements a two-tier decision boundary as shown in equation 4.2. According to this equation, the amount of motion suffered by a skin-coloured pixel is 1 if the conditions specified for threshold T_1 or T_2 are met, otherwise the motion is 0. In equation 4.1, the two conditional thresholds values T_1 and T_2 are represented by a single value, T.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

$$F(x,\ y,\ t) = \begin{cases} 0 : [S_c(x,\ y,\ t) = 0] \cup [|D(x, y, t)| = 0] \\ 1: S_c(x,\ y, t) > 0 \cap [(|D(x,\ y,\ t)| > T_1 \cap Ed(x,\ y, t) > 0) \cup (Ed(x,\ y, t) = 0 \cap |D(x,\ y, t)| > T_2)] \end{cases} T_1 < T_2 \qquad (4.2)$$

The above equation illustrates how the two-tier decision boundaries, T_1 and T_2, are used for determining whether a skin-coloured pixel forms part of an interesting region or not. Firstly, all non-skin-coloured regions, $S_c(x, y, t) = 0$, are discarded. T_1 is then used for thresholding the absolute motion values, $|D(x, y, t)|$, at any of the remaining edge-based pixels, $Ed(x, y, t) > 0$. T_2 is used for verifying whether the amount of motion suffered by any other skin-coloured pixel is of any significance or not. All edge based pixels whose motion values are lower than T_1 are discarded. Low values of T_1 are chosen in order to preserve object outlines. Large values of T_1 will erode the boundaries of all slow moving objects. If the edges that constitute parts of the object boundary lines are lost, then it will be very difficult to confine the computations of the feature set to within a region of interest. If a region of interest is not properly defined, the computed feature set may not at all represent the object of interest. Besides, loss of boundary information often leads to loss of important gesture information. For instance, hand-shape, fingertip motion, and other features that describe the object of interest would be wrongly defined. The fact that the human hand can easily assume different spatio-temporal appearances makes it difficult for machines to recognize gesturing hands by merely analyzing the objects' outlines. In practice, coarse hand information and some prior knowledge of the permissible hand-shapes are often combined and used for identifying candidate hand regions. In spite of the presence or not of image noise; a combination of image edges, motion, skin-colour cues, and other features that characterize the detected image blobs are used to identify the most probable hand regions.

We have just said that low motion threshold values help to preserve image boundaries. However, low motion threshold values also amplify the impact of noise and this consequently distorts the quality of the segmentation results. Poor segmentation results make it difficult for machines to identify the most probable hand and head regions. If some image blobs are not properly modeled, both the hand tracking and the gesture

UNIVERSITY of the
WESTERN CAPE
https://etd.uwc.ac.za/

recognition processes are bound to fail. It has been established that object recognition largely depend on both segmentation and tracking results, however, object segmentation and object tracking also largely depend on each other [54][52]. Despite the use of two threshold values, perfect segmentation is still difficult to achieve.

The following two subsections discuss how the morphological filter was implemented in this research.

4.3 Morphological filter

A morphological filter that eliminates isolated edges and/or moving skin-coloured regions was implemented in our experiments. The subroutine that implements this filter automatically adjusts the sizes of the sampling sub-regions. Initially, an image is subdivided into 3x3 pixel sub-regions. The routine then determines the total number of moving skin-coloured pixels in each sub-region. Whenever, a 3x3 sub-region with less than 5 skin-coloured pixels is encountered, such a sub-region is set to zero, and thus discarded. At the same time, all sub-regions with exactly 5 skin-coloured pixels, which are totally surrounded by sub-regions with less than 5 skin-coloured pixels are also eliminated. The elimination of sub-regions which contain a few skin-coloured pixels helps to remove all isolated skin-coloured pixels. Only significantly large clusters of skin-coloured pixels are maintained. However, the morphological filter presented in this work does not remove the thin, elongated skin-coloured sub-regions. This is a deliberate attempt to avoid deleting possible finger projections at this stage. If all finger-like projections were to be eliminated soon after they are discovered, then all fingertip information would be lost. Some of the finger-like projections are deleted during the boundary tracing process only if it has been established that the magnitude of each of those projections is much larger than the expected size of a finger projection. It must also be noted that all non-skin-coloured pixels are discarded once they are discovered, and hence all edge based pixels that are contained in non-skin regions are also discarded.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

In order to further refine the filtration process, the sizes of the sub-regions are increased in multiples of 3. In this research, 6x6 and 24x24 pixel sub-regions were used. In a 6x6 sub-region, all sub-regions which contained less than 7 skin-coloured pixels were deleted, while in a 24x24 sub-region, all sub-regions with less than 40 skin-coloured pixels are also deleted. As the size of the sub-region is increased, the minimum number of skin-coloured pixels that each sub-region should contain was drastically reduced. This was done in order to minimize the chances of deleting useful object data. We incremented the sub-regions by multiples of 3 in order to ensure that the process of computing the total number of skin-coloured pixels in progressively larger sub-regions is merely reduced to adding the subtotals of each of the smaller sub-regions which are totally contained within the bigger sub-region. This way we reduce the number of run-time operations that are executed by our morphological filter; and hence reduce the complexity of our algorithm.

4.4 The border tracing algorithms

In an effort to identify the hand regions, our algorithm first isolates the head region. The head region is usually represented by a big elliptical shaped region, which is mostly situated within the top section of an image. Computer vision researchers often identify the head region basing on its geometric features. Consequently this research implements two different boundary tracing algorithms, one of which is specifically designed to isolate the head region, while the other is designed to identify the outlines of any other connected region. These two algorithms are explained in subsections 4.4.1 and 4.4.2 respectively.

4.4.1 Head tracing algorithm

Since the human head is a rigid object that assumes a well-known geometric shape, a deformed model based boundary tracing method should be implemented for identifying

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

and tracing the head regions in a noisy environment. This section presents a method of tracing head outlines basing on the density of the connected edge-based pixels and the tracing direction assumed by the last detected pair of boundary pixels. The head tracing algorithm is designed to trace only the edge-based pixels which describe elliptical shaped curvatures.

When searching for the most probable head region, the image is first subdivided into 6x6 sub-regions. The sub-regions are then scanned from left to right and from top to bottom. Whenever a new edge-based sub-region, $P(x, y)$, is encountered, the algorithm identifies another edge-based sub-region, $P(x^1, y)$, such that if x^l is the highest edge-based coordinate value of the current scan-line and $P(x^1, y)$ is directly or indirectly connected to $P(x, y)$. Besides, each of the sub-regions $P(x^1, y)$ and $P(x, y)$ must also be directly connected to one or more edge based sub-regions from the next scan-line. If more than one sub-region from the proceeding scan-line is directly connected to $P(x, y)$ or to $P(x^1, y)$, then the algorithm searches for a set of edge-based sub-regions that produces the best elliptical shaped curvature. Usually a sub-region that contains the highest number of edge-based pixels, which also approximately falls directly below the current maxima (minima), is often designated as the subsequent maxima (minima). The cross-sectional distance between each subsequent pair of minimum and maximum boundary based sub-regions is used as an additional constraint for screening the possible boundary pixels. We implement a sub-function that verifies whether the magnitudes of the first five consecutive cross-sectional distances are comparatively greater than their predecessors. An edge-based sub-region $P(x, y)$ is part of the head region only if we can find a set of edge-based sub-regions, ζ, whose elements are directly or indirectly connected to $P(x, y)$ such that all elements of ζ define an elliptical-shaped curvature that spreads over a significantly large surface area.

UNIVERSITY of the
WESTERN CAPE

Once it is established that each of the first five or so cross-sectional distances is increasingly greater than its predecessor, the algorithm then searches for the largest

https://etd.uwc.ac.za/

cross-sectional distance. Under normal circumstances, whereby the head outlines are not distorted by the head dressings, the largest cross-sectional distance is normally associated with the transversal distance between the ear regions. Moving downwards from the point where the largest cross-sectional distance is encountered, the distance between the extreme boundary-pixels gradually becomes smaller and smaller. Finally, the smallest cross-sectional distance is found around the neck-chin region. In order to verify whether the smallest cross-sectional distance was not incidentally generated by distorted image data, we check for any dramatic and a non-systematic changes in the gradients of the boundary lines. Images boundaries which are not heavily distorted by noise often give rise to steadily increasing or decreasing cross-sectional distances. Starting from the neck-chin region, the cross-sectional distance remains almost constant for sometime before it starts to steadily increase again. A steadily increasing cross-sectional distance, which eventually surpasses the highest recorded cross-sectional distance, represents the shoulder regions of the human body. The search for the head region is deemed successful once the shoulder regions have been identified. However, the neck-chin region is considered the cut-off point for the head region.

The next subsection explains why it is important for our gesture recognition system to locate the head region.

UNIVERSITY of the
WESTERN CAPE

4.4.1.1 Significance of the head position

The rigid nature of human head makes it easier to locate the human head than the human hand. Once the location of the head region is identified, it is then used by our system as a reference point from which the location of other body regions can be inferred. Since all the pictures used in this research stretch from the waist region to the head region, then it is easy to determine a 2D coordinate system that approximates the location of the chest, the stomach, and/or the shoulder regions in a human image. Such a co-ordinate system normally varies from person to person depending on how the image was shot. However, it is very important for a chereme-based gesture recognition

https://etd.uwc.ac.za/

system to determine the position of a gesturing hand in terms of the body regions. In our system, the above mentioned positions are approximated as shown in Table 4.1.

Table 4.1: Determination of the Hand Position chereme

Head region
Width= $C_m \pm 3x24x24\ pixels$
Height= $C_m \pm (C_m - C_{NR})$

Chin Region (C_{NR}) Centre of end of head region (H_{ER}) (H_{ER} is located using software)
Neck Region Longitudinal distance between head and shoulder regions (shoulder is located using software)
Chest Regions (C_R) $C_R \geq \left(\frac{C_m - B_c}{2} \right)$
Stomach Region (S_R) $S_R < \left(\frac{C_m - B_c}{2} \right)$
Waist Region Centre of the Most Bottom Section of an Image, B_c

Used Symbols:

C_m is the centre of mass of the head region

C_{NR} is the centre of the chin region which also approximated as the centre of the line that demarcates the head region from the neck region (H_{ER})

C_R the chest region

S_R the stomach region

B_c centre of the most bottom line of the image frame

UNIVERSITY of the
WESTERN CAPE

The next subsection explains how the boundaries of candidate hand regions are traced.

4.4.2 Tracing the boundaries of irregularly shaped objects

Zhu [49] defined hand segmentation as a process of delineating the moving hand from the complex background. In Section 3.3.2 we explained the different approaches to the segmentation problem. In most cases the outlines of the segmented regions consist of intertwined curves which are difficult to classify using geometrical model fitting

https://etd.uwc.ac.za/

techniques. In this subsection we explain the design of a boundary tracing algorithm that identifies the connected boundary pixels of any cluster of moving skin-coloured pixels. The curvatures that surround of each bitmap region help to identify a set of connected sub-regions that define a particular region of interest. Connected image components literally represent the objects contained in the input image. Once the boundaries of an image blob are identified, it is easier to confine the feature detection process to a particular region of interest.

The problem of finding connected components of image objects has been extensively researched. However, this problem is still far from being solved [71][21][90]. In particular, the choice of the threshold values that are used during segmentation directly impact on the level of interlace between the otherwise different output regions. It should be noted that it is very difficult to find a single threshold value that effectively isolates all regions of interest (ROIs) [4]. Besides the impact of threshold values, image noise and image filtering algorithms often give rise to jagged object boundaries which are very difficult to trace. In most cases, object contours are not always defined by a continuous string of clearly identifiable curves, and hence contours cannot accurately identify image boundaries. Traditional boundary tracing algorithms loop through the object's outer boundary pixels in either clockwise or anticlockwise direction until the first and the second pixels visited at the start of the tracing process are revisited in the same order in which they were first encountered [21][1][71]. However, such approaches may either result in premature exits or endless loops, especially in regions where the object outlines are corrupted by noise. Our boundary tracing algorithm scans each input image from left to right, starting from the top left corner going downwards. In each scan-line, the algorithm searches for consecutive skin-coloured sub-regions. Whenever such a pattern is found, the algorithm assigns a value of 4 as the relative direction, *dir*, of the last visited non-skin-coloured pixel. The algorithm then determines if the current sub-region, $P(x,y)$, lies on the boundaries of a connected region. In order to achieve this, the algorithm verifies whether $P(x,y)$ is connected to any one of its 8-neighbourhood pixels (see Figure 4.2). In this section the words sub-

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

region and pixel are often used interchangeably since all analysis done at sub-region actually mimic the analysis that should be done at pixel-level. However, pixel-level analysis would present too many variations since there are often too many loose image pixels.

Definition 1: A skin-colored pixel, *P(x,y),* is said to be connected to one or more of its neighbours, *P(i,j),* if *P(i,j)* is also skin-coloured and $P(i,j) \in \{P(x+1,y+1), P(x,y+1), P(x-1,y+1), P(x-1,y), P(x-1,y-1), P(x,y-1), P(x+1,y-1), P(x+1,y)\}$.

Definition 2: A connected skin-coloured pixel, *P(x,y),* forms part of the outer boundary of object *X* if one of the last visited neighbourhood pixels, *P(i,j)*; where $P(i,j) \in \{P(x+1,y+1), P(x,y+1), P(x-1,y+1), P(x-1,y), P(x-1,y-1), P(x,y-1), P(x+1,y-1), P(x+1,y)\}$; is non-skin-coloured.

The order in which the neighbouring pixels are visited largely depends on whether a clockwise or anticlockwise trace is implemented.

$P_{(x-1,y-1)}$	$P_{(x,y-1)}$	$P_{(x+1,y-1)}$
$P_{(x-1,y)}$	$\mathbf{P}_{(x,y)}$	$P_{(x+1,y)}$
$P_{(x-1,y+1)}$	$P_{(x,y+1)}$	$P_{(x+1,y+1)}$

a) 8-Neighboring Pixels of *P(x,y)*

3	2	1
4	$\mathbf{P}_{(x,y)}$	0
5	6	7

b) Tracing Directions

Figure 4.2: Neighbour pixels and tracing direction

Some of the fundamental steps of our boundary tracing algorithm are outlined in Algorithm 4.1.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Algorithm 4.1: Improved Boundary Tracing Algorithm:

```
1. Set dir ←4.
2. Scan image pixels from top to bottom, left to right until a skin-coloured pixel P(x,y) is found
3. Set P(x,y) to white and assign a negative value to it; check whether P(x+1,y) is also skin-coloured
   a. If P(x+1,y) is skin-coloured, set M←5 else M←0;
   b. Push into the clockwise stack, the point P(x,y) and dir=1;
4. REPEAT UNTIL all anticlockwise boundary pixels are visited
   a. dir= [(dir+1) MOD 8]
   b. if current pixel, P(i,j), is black
      i. set P(i,j) to white and assign a negative value to it;
      ii. Analyze image pattern and determine dir2
          • If pattern ∈ {12 pattern set}then
            o Push P(i,j) and dir2 into either clockwise or anticlockwise stack
      iii. If dir is odd then dir=[(dir+5) MOD 8] else dir=[(dir+6) MOD 8]
   c. If P(i,j) is negative, POP UP anticlockwise stack; else if anticlockwise stack is empty
      CALL clockwise tracing module; else exit trace boundary if all stacks are empty
5. Clockwise Module
   • REPEAT UNTIL clockwise stack is empty
     a. dir =[(dir-1)MOD 8]
        • If dir == -1 then dir ==7
     b. Repeat STEP 4. b. i. and 4. b. ii. above
        a. if dir is odd then dir=[(dir+3) MOD 8] else dir=[(dir+2) MOD 8]
     c. If P(i,j) is negative, Pop up the clockwise stack; else if anticlockwise stack is
        empty Return to anticlockwise tracing module
```

UNIVERSITY of the
WESTERN CAPE

The boundary tracing algorithm described above scans and marks each boundary sub-region once, and stops scanning only if all the connected boundary sub-regions are visited. In order to ensure that all boundary pixels are visited during the boundary tracing process, this algorithm tracks boundaries in both clockwise and anticlockwise directions. In each tracing step, the algorithm keeps a record of the following parameters:

- the last visited boundary pixel;
- the direction from which the current pixel was entered;
- a flag that indicates whether the previous outer boundary pixel width is greater than 1;

https://etd.uwc.ac.za/

- two stacks, one for stacking all active branching points that are encountered during clockwise trace and the other one for stacking all the active branching points that are encountered during the anticlockwise trace.

In the event that a dead end is reached before all boundary pixels are traced, boundary tracing is continued from the last encountered branching point of each respective tracing direction.

In an attempt to determine which neighbouring sub-region is searched next during the anticlockwise trace, the algorithm adds 1 to the current tracing direction, *dir*. In actual fact, the new value for *dir* is computed as shown in the equation below.

$$dir=[(dir+1) \text{ MOD } 8]$$

The algorithm then checks whether the image pixel pointed to by the new *dir* value is skin-coloured or not. At each stage of the search process, the newly found outer boundary pixel, $P(x,y)$, is always used as the new reference point. Whenever a new border pixel is found, that pixel is first marked after which it is designated as the new reference point. Besides designating the new reference point, the algorithm also checks for any branching points before it backtracks onto the last visited non-skin-coloured pixels. The new tracing direction, *dir*, of the last visited non-skin-coloured pixel is now given in terms of the newly assigned reference point. Its direction is computed using one of the two equations shown below:

$$dir=[(dir+6) \text{ MOD } 8] \tag{4.3}$$

$$dir=[(dir+5) \text{ MOD } 8] \tag{4.4}$$

Equation 4.3 is used only when the current direction is represented by an even number, i.e. 0, 2, 4 or 6; while equation 4.4 is used when the current tracing direction is odd. The anticlockwise tracing module repeatedly executes the above mentioned steps until all linked object boundaries that can be visited from an anticlockwise direction are traced. When all the connected anticlockwise outer boundary pixels are traced, the algorithm then calls the clockwise tracing module.

UNIVERSITY of the WESTERN CAPE

https://etd.uwc.ac.za/

The clockwise tracing module is implemented in a similar way to the anticlockwise module except that the next pixel to be visited, *dir*, is computed through a modulus division of *dir-1* by 8. The expression *dir-1* represents the previous tracing direction minus 1. However, in cases where -1 is obtained as the new *dir*, *dir* is reassigned a value of 7. Again after designating a new reference point, the algorithm checks for any branching point before it backtracks to one of the previously visited positions. In the clockwise tracing module, the backtracking algorithm utilizes one of the following pair of equations.

$$dir=[(dir+2) \text{ MOD } 8] \tag{4.5}$$

$$dir=[(dir+3) \text{ MOD } 8] \tag{4.6}$$

If the current tracing direction is represented by an even number and if a new boundary pixel has been found, equation 4.5 describes how the backtracking is achieved in a clockwise tracing module. On the other hand equation 4.6 shows how backtracking is accomplished if the current tracing direction is represented by an odd number. Just like in anticlockwise tracing, all newly discovered clockwise and anticlockwise branch-off points are passed into the respective stacks. Meanwhile, all clockwise branching points which were entered into the stack are popped out whenever the current tracing route is exhausted. Eventually the function call is returned to the anticlockwise module. The anticlockwise module will terminate the boundary tracing process once both the clockwise and the anticlockwise stacks are empty.

UNIVERSITY of the
WESTERN CAPE

Both the anticlockwise and the clockwise modules use a similar methodology for identifying the branching points. In fact in both modules, branching points mostly occur at the point of intersection of a one pixel thick segment and a multi-pixel thick segment. Other branching points are found where an external boundary borders with an image hole. Although the process of identifying the branching points sounds trivial, we found that the branching points are best described using at least 12 unique blob patterns. The co-ordinates of each newly found branching point, together with the tracing direction that should be assumed whenever a branching point is revisited, are both pushed into a stack. The choice of a future tracing direction for each branching

https://etd.uwc.ac.za/

point mainly depends on whether the current tracing direction is clockwise or anticlockwise based; the structure of the branching point; and whether the new tracing direction would be clockwise or anticlockwise based.

The boundary tracing algorithm presented in this section possesses several advantages over the traditional tracing algorithms [21][1][71]. Firstly, the above given algorithm terminates the boundary tracing process only if all the connected outer boundary pixels are visited. The new boundary tracing algorithm does not use the first and second pixels discovered at the start of the tracing process as its terminal conditions. This situation ensures that neither premature exits nor endless tracing loops are encountered during the boundary tracing process. Secondly, our algorithm ensures that the outer boundary pixels are traced within the least possible number of steps. In fact most of the boundary pixels are visited only once. Somehow, this reduces the computational complexity of our boundary tracing algorithm. Thirdly, the improved boundary tracing algorithm helps to detect fingertips and fingertip motion information. Fingertip information is vital for the hand gesture recognition process. The following subsection explains the statistical feature extraction process.

4.5 Extracting the statistical feature set

UNIVERSITY of the
WESTERN CAPE

After extracting all connected outer boundary pixels of a region, a statistical distribution of the image pixels that lie within the outlined region is computed. In order to ensure that pixels fall within a region of interest, we first determine the start and the end boundary pixels for each scan-line of a current image blob. The difference between the start and the end boundary pixels of each scan-line is used as an estimate of the number of skin-coloured pixels that are contained in each row of the image blob. The total number of pixels enclosed in the image blob gives a rough estimate of the relative size of the image blob. A blob with a relatively small surface area is assumed to have been generated by image noise and is thus discarded. If a skin-coloured blob is relatively large, its shape is approximated using the statistical distribution of the

https://etd.uwc.ac.za/

enclosed skin-coloured pixels. The correlation matrix of the skin-coloured pixels given by δ_{xx}, δ_{yy} and δ_{xy} (see equations 4.10, 4.11and 4.12) respectively, gives a measure of the dispensation of blob pixels along the *x*, the *y* and *xy* directions. According to equations 4.7, 4.8 and 4.9 the centre of mass is the average position of a blob's pixels, from which the dispersion of skin-coloured pixels along both the *x* direction and the *y* direction is measured. The number of skin-coloured pixels $|D(x_i,y)|$ from the current image blob, which fall along the line $x=x_i$; where $x_i \in \{-\infty, \ldots, -2, -1, 0, 1, 2, \ldots, +\infty\}$, must be computed. The sum of the products of $|D(x_i,y)|$ and the corresponding x_i value must also be computed. The sum of the products is then divided by the total number of skin-coloured pixels that are projected at each position. The result of the above process gives an estimated average position along the *x* direction, m_x, of all skin-coloured pixels that are contained within an object of interest. The average position of all skin-coloured pixels along the *y* direction, m_y, is also calculated in a similar manner.

$$m_x = \frac{\sum_{x,y} x_i |D(x_i,y)|}{\sum_{x,y} |D(x_i,y)|} \tag{4.7}$$

$$m_y = \frac{\sum_{x,y} y_i |D(x,y_i)|}{\sum_{x,y} |D(x,y_i)|} \tag{4.8}$$

$$m_c = (m_x, m_y) \tag{4.9}$$

UNIVERSITY of the
WESTERN CAPE

The dispersion of skin-coloured pixels from the centre of mass of an image blob is described by the covariance matrix, Σ,

$$\Sigma = \begin{bmatrix} \delta_{xx} & \delta_{xy} \\ \delta_{xy} & \delta_{yy} \end{bmatrix}$$

Equations 4.10, 4.11 and 4.12 show how the elements of Σ are computed.

$$\delta^2 x = \delta_{xx} = \frac{\sum_{x_i,y} |D(x_i,y)|(x_i - m_x)^2}{\sum_{x_i,y} |D(x_i,y)|} \tag{4.10}$$

$$\delta^2 y = \delta_{yy} = \frac{\sum_{x,y_i} |D(x,y_i)|(y_i - m_y)^2}{\sum_{x,y_i} |D(x,y_i)|} \tag{4.11}$$

https://etd.uwc.ac.za/

$$\delta_{xy} = \sum_{x_i, y} \frac{|D(x_i, y)|(x_i - m_x)(y - m_y)}{|D(x_i, y)|} \tag{4.12}$$

In statistics, variance gives a measure of the dispersion of some sampled observations about the mean. If a very large sample size is used in a survey, the distribution of sample data would assume a normal distribution curve. Variance is calculated as the average of the square of deviations of all possible observations from the population mean [61][62]. On the other hand, standard deviation is a statistic that uses the same unit of measurement as the sampled data, and it is used describe the spread of data about the mean. Standard deviation is computed as the square root of the variance. Where two or more variables were sampled, covariance is used as a measure of the joint variability between those variables. Joint variability describes the degree of association between variables which is induced by an operation or transformation that has been performed on those variables. For instance, joint variability (correlation) between variables x and y is expressed in terms of δ_{xx}, δ_{yy} and δ_{xy}. Statisticians often use the term variance to describe how a variable correlates with itself. On the other hand, δ_{xy} describes the degree of association between variables x and y.

Besides describing shapes in terms of their statistical mean, variance and standard deviations, eccentricity is also often used to describe how image pixels are distributed about the centre of mass. Eccentricity is the ratio of the principal axis to the minor axis of an ellipse that bounds a given image blob. According to [61][57], elements of the covariance matrix, Σ, can be used to calculate the parameters of an ellipse that circumscribes a candidate hand region. These parameters are the principal axis, α, the minor axis, β, and the angle of orientation, θ.

$$\Lambda = \sqrt{(\delta_{xx} - \delta_{yy})^2 + 4\delta_{xy}{}^2} \tag{4.13}$$

$$\alpha = \sqrt{\frac{\delta_{xx} + \delta_{yy} + \Lambda}{2}} \tag{4.14}$$

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

$$\beta = \sqrt{\frac{\delta_{xx} + \delta_{yy} - \Lambda}{2}} \tag{4.15}$$

$$\theta = \tan^{-1}\left(\frac{\delta_{xy}}{\alpha^2 - \delta_{yy}}\right) \tag{4.16}$$

Equations 4.13 to 4.16 show how the elements of Σ are used to calculate α, β, and θ. In this research, eccentricity is used as one of the object features.

The x and y standard deviations of the skin-coloured pixels that constitute each image blob are calculated as shown in equations 4.17 and 4.18.

$$\delta_x = \frac{\sum_{x_i, y} \left| D(x_i, y)(x - m_x) \right|}{\sum_{x_i, y} \left| D(x_i, y) \right|} \tag{4.17}$$

$$\delta_y = \frac{\sum_{x_i, y} \left| D(x_i, y)(y - m_y) \right|}{\sum_{x_i, y} \left| D(x_i, y) \right|} \tag{4.18}$$

In this research, the ratio of the total surface area of a skin-coloured blob to the area of the best fit bounding rectangle is used as one of the important object features. This ratio is often referred to as the rectangularity of an object. Rectangularity measures the amount of space of a minimal bounding box that is covered by a silhouette of an object. It is a coarse estimate of how object pixels are distributed in an image.

UNIVERSITY of the
WESTERN CAPE

Besides the features explained above, two small lists, b_x and b_y, are extracted from the x and the y coordinate values of the image boundary curves in such a way that the distance between any two elements of b_x is γ_1, and the distance between any two elements of b_y is γ_2. The two lists, b_x and b_y, are incorporated into the feature vector in order to augment its capacity to distinguish between closely similar shapes. A list of flags, $\{f_1, \ldots, f_5\}$, that indicate the presence of one or more fingerlike projections in object data is also included in the feature list. Unfortunately, no information that distinguishes between the different fingers is provided since this is quite a cumbersome task to achieve. Firstly, we are not really sure whether all fingerlike projections

https://etd.uwc.ac.za/

represent true fingers. Secondly, it is difficult to figure out the position and the identity of an extended finger as finger configurations are in most cases always affected by occlusion.

All of the above mentioned features are either used for tracking candidate hand regions or for classifying object shapes. Tracking moving hands can be accomplished through a small list of features that enable a rough classification of image blobs. In this research we track candidate hand regions using the features outlined in equation 4.19.

$$F^{T}_{(x,y)} = \left\{ b_{s\_area}, \delta_x, \delta_y, \delta_{xy}, \left| m_x^p - m_x^c \right| + \left| m_y^p - m_y^c \right| \right\} \qquad (4.19)$$

where $b_{s\text{-}area}$ is the total surface area of the image blob, δ_x and δ_y are the standard deviations in the x and in the y directions respectively; δ_{xy} indicates how variable x and y co-vary against each other, and $\left| m_x^p - m_x^c \right| + \left| m_y^p - m_y^c \right|$ is the distance between the centre of mass of the current image blob, (m_x^c, m_y^c), and the previous image blob, (m_x^p, m_y^p).

We used the feature set shown in equation 4.20 for classifying hand-shapes.

$$F^{S}_{(x,y)} = \left\{ \delta_x, \delta_y, \delta_{xy}, \theta, \Re_1, \Re_2, f_1, f_2, f_3, f_4, f_5, b_{x_1}, b_{x_2}, b_{x_3}, b_{x_4}, b_{x_5}, b_{x_6}, b_{x_7}, b_{y_1}, b_{y_2}, b_{y_3}, b_{y_4}, b_{y_5}, b_{y_6}, b_{y_7} \right\} \qquad (4.20)$$

UNIVERSITY of the
WESTERN CAPE

where $\Re_1$ and $\Re_2$ represent eccentricity and rectangularity respectively. The Support Vector Machine [14][63] was implemented in order to classify hand-shapes. Section 4.7 briefly discusses how the SVM is implemented in this research and Section 5.5.3 presents the SVM based classification results for aligned sequences of candidate hand blobs.

In the following subsection, we explain how the improved boundary tracing algorithm isolates fingertip motion.

https://etd.uwc.ac.za/

4.5.1 Extracting fingertip motion

Besides clearly demarcating different bitmap regions, the boundary tracing algorithm is also used for marking out the probable fingerlike projections that are associated with each bitmap region. Fingerlike projections help to isolate fingertips and to quantify fingertip motion. Except in experiments where artificial colouring techniques are applied, the available gesture recognition systems mainly ignore fingertip information. In fact, fingertip motion is often ignored because of the difficulties that surround the process of isolating and tracking finger motion under natural conditions. Fingertip motion only constitutes an insignificant proportion of the overall hand motion. In the field of computer vision, a slight shift in the position of the centre of mass or the standard deviations of an object are often attributed to jaggy camera and/or hand motions, or other forms of noise that are associated with image capture.

Depending on the size of the thin elongated projection mapped out by the boundary tracing algorithm, these projections are either classified as a probable finger region or they are discarded. In Section 4.3.1 we mentioned that a fingerlike projection is discarded only if its magnitude is much greater than a prescribed length, x, which corresponds to an average sized finger projection. But how does a boundary tracing algorithm determine the length of a fingerlike projection? In the first paragraph of Section 4.5 we explained how the surface area of each skin-coloured blob is extracted. It was explained that for each scan line, a physical count of the total number of skin-coloured pixels that are confined within the boundaries of an image blob is carried out. If the number of skin-coloured pixels that are enclosed at a particular section of a scan-line that falls within the ROI is smaller than a specific threshold value, then the cross-sectional width of the ROI is considered too small to constitute the wrist region of a hand. If an image blob with a small cross-sectional width is directly connected to an image blob with a larger cross-sectional width, then the thin cross-sectional area is presumed to represent a finger projection of a hand region. However, a further analysis of the whole image blob should be done. The approximate magnitude of a probable

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

finger projection is determined by simply counting the number of connected outer boundary pixels that outline the thin cross-sectional area of the image blob. In this research, if a thin connected region is more than 360 pixels long, then such a region is discarded since our manual verifications have shown that most finger projections of the images that we worked with were less than or equal 360 pixels long. All fingerlike projections which are more than 360 pixels long are assumed to represent the boundary regions of moving non-skin coloured objects, which were filmed against a skin-coloured background. In order to determine whether the fingerlike projection is moving or not, the centre of mass of each fingerlike projection is determined. If the fingerlike projection persists from frame to frame, the centres of mass and the standard deviations of consecutive image frames are compared with each other.

A fingerlike projection is regarded as representing a moving finger if its centre of mass or one of the two standard deviations changes between successive image frames. The type of finger motion can be learnt by implementing the HMM. Firstly, the statistical data that describes the configurations of each fingerlike projection are extracted and passed into the HMM. In this research, the various transitional states that characterize each HMM are represented by the changing parameter values for the *x* and/or *y* standard deviations, centres of mass, orientation, rectangularity, and eccentricity of the fingerlike projection. Although recurrent neural networks achieve better recognition rates than HMMs, HMM can be sufficiently trained from a few training samples within a short time interval. Each finger motion is learnt and recognized as a separate entity, and hence, each training dataset that is passed into the HMM represents particular finger motions.

UNIVERSITY of the
WESTERN CAPE

After the HMM is sufficiently trained on how to recognize particular finger motions, the recognition module of the HMM is called in order to classify finger motion. In this research finger motions constitute one broad class of the sign language cheremes which we call the finger-motion chereme. The finger-motion chereme is herein represented by the capital letter F. Within the finger-motion chereme, each individual finger motion is

https://etd.uwc.ac.za/

represented by a positive whole number that precedes the capital letter *F*. For instance, individual cheremes may be uniquely represented by one of the following symbols; *F1, F2, F3…, Fn-1, Fn*. Once finger motion is recognized, a symbol that represents that finger motion is passed to the gesture recognition module as one of the components of the gesture being investigated. Other cheremes such as the hand-shape, the hand motion, the relative position of the gesturing hand are also fed into the gesture recognition module. Our system can identify major changes in hand orientation but it is not easy to determine the minor changes in the orientation of a gesturing hand. Consequently the ori chereme [11] is not used in this research.

4.6 Tracking candidate hand regions

After successfully locating the head region, our algorithm then searches for the probable hand regions. In an image, the most probable hand regions are mainly represented by elongated elliptical shaped blobs which either have a steadily increasing or a steadily decreasing cross-sectional distance. In some cases, finger-like projections are found attached to the probable hand regions. The ratio of the major axis t_0 to the minor axis t_1 of the best fit ellipse that circumscribe the hand region is usually much smaller than that of an ellipse that bounds the head region. However, in the case of the hand region, this ratio largely depends on whether the fingers are stretched or clinched.

UNIVERSITY of the
WESTERN CAPE

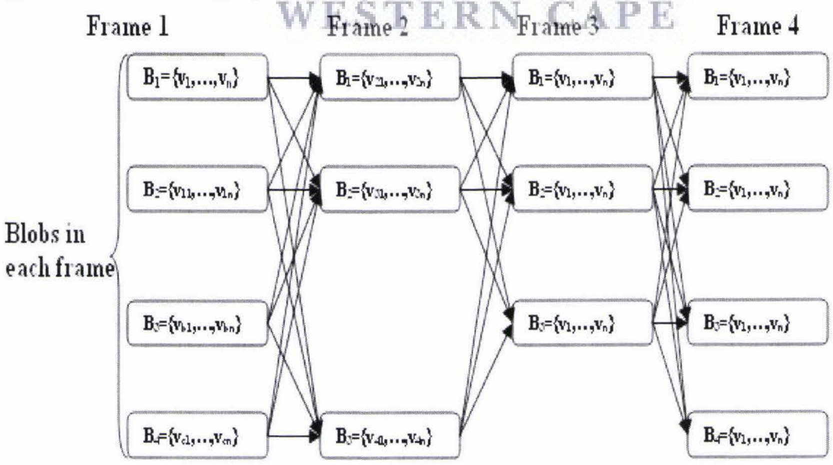


Figure 4.3: Probabilistic matching of image blobs

https://etd.uwc.ac.za/

In order to identify the blobs that most probably represent gesturing hand, similarly textured image blobs from different image frames are aligned to each other as shown in Figure 4.3. The following pseudo-code shows how different image blobs are matched.

Algorithm 4.2: Determining the Best Matching Blobs

```
if BestMatchList is empty;
    BestMatchList = present list of image blobs;
if BestMatchList is not NULL
  {
    For BlobList=1 to n
      {
        LX:
        For BestMatchList=1 to m
          {
            if (FeatureListMatch)
              Store in temporary BestMatch array
          }
        Determine the best of the best matches
        If any of the remaining BlobList is a better match than current best match THEN GOTO LX:
          Else store best in BestMatchList queue
      }
  }
```

The level of similarity between two image blobs that belong to different image frames is determined based on the following set of parameters: namely the total surface area, distance between the two blobs' centres of mass, and the standard deviations of skin coloured pixels along the *x* and the *y* directions respectively. The algorithm first compares the surface areas and the two standard deviations of any unmatched pair of candidate hand regions. If the two blobs' standard deviations and surface areas are within the range of some prescribed threshold values, and if no other blobs have comparatively similar differences, then the two blobs are considered matching each other. On the other hand, if two or more candidate hand regions from either the current frame or the previous frame are comparatively similar, then the best match is decided based on the probable motion vectors. However, if the two blobs' surfaces areas differ by a small magnitude, irrespective of which blob has the bigger surface area, a blob with the smallest difference for both the *x* and *y* standard deviations is considered the best match. If all the above mentioned conditions fail to clearly decide the best match, then one of the blobs from the previous frames whose centre of mass is the closest to

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

the current blob is considered the best match. Such a match is considered to represent slow moving or stationary objects.

Each candidate region can only be aligned to one or none of the candidate regions from the other frames. A blob from one of the previous frames that does not match any other blob from any of the proceeding 6 frames is discarded. Such a blob could have been generated by noisy data.

A candidate hand is considered successfully tracked only if the total number of aligned blobs is greater than half the total number of input frames. The set of parameters which are used for matching candidate hands is considered adequate for hand tracking purposes only if we can always extract at least one complete set of aligned blobs which represents every input image. A successfully aligned sequence of image blobs is one in which a single blob from a current image frame that contains an object of interest is always correctly matched to a sequence of previously matched image blobs that also contain the object of interest.

When processing a single handed gesture, sometimes more than one candidate hand regions are successfully tracked through a sequence of image frames. Out of all the successfully tracked candidate hand regions, only one set of aligned blobs actually represents the gesturing hand. How then do we determine the correct sequence of aligned blobs? This problem is solved by implementing the SVM or any other object classifier. In this research, we use the SVM for classifying the matched candidate hand blobs. The following subsection explains how hand classification is achieved.

UNIVERSITY of the
WESTERN CAPE

4.7 Classifying sets of candidate hand regions

In order to classify candidate hand configurations we implement a Support Vector Machine with a linear kernel [63]. Firstly, the SVM is trained using positive and negative training samples. A positive training sample consists of a set of original data

https://etd.uwc.ac.za/

samples which are somehow intentionally polluted by aggregating a few noise signals. On the other hand, the negative training samples were mainly composed of data samples that represent other hand shapes. Some of the negative training samples were derived from the original data samples which were intentionally subjected to noise signals. In our experiment, the amount of noise used for polluting data samples ranged between 1% and 5% of each the original feature values. Since the testing samples always contain non-polluted data samples, it implicitly means that the training and the testing samples are always are different.

The training samples are used to train the SVM system. Various parameters and different types of kernels are used when training the SVM system (see Table 5.9 for a list of some of these parameters). A trained SVM system is used for evaluating the performance of each test sample against each of the trained samples. The performance results for each test set against each train set are given in terms of specificity, sensitivity and positive predictive values, which are normally expressed as decimal fractions or percentages.

Our train and test data samples were extracted from seven dynamic SASL gestures. Five people were asked to repeatedly perform seven SASL gestures. On average each dynamic gesture exhibited 35 person-dependant variations of a single hand-shape. Thus for each gesture, an average of 175 person-independent hand-shape variations were studied. The negative train data samples for each gesture consists of a consortium of hand-shapes collected from different gesture classes and those hand-shapes which were distorted by image noise. On the other hand, manually verified sequences of correctly aligned hand blobs are used as positive train samples. Negative test samples are selected in a similar manner to how negative train samples are extracted. We used the SVM-light source code [63] to test how sequences of candidate hand regions relate to known hand-shape sequences. We observed the hand-shape recognition rates for both the person independent and the person-dependant situations (see details in Section 5.5.2). However, this work is based on the assumption that the hand configuration does

https://etd.uwc.ac.za/

not change much during the execution of a gesture, and hence a set of successfully matched blobs is considered to represent the same hand configuration.

4.8 Summary

This chapter explains how a ROI is identified and processed by our gesture recognition system. Specific image attributes, such as skin-colour, edges and motion cues were integrated in such a way that all possible boundary information of all moving objects was preserved. Preservation of boundary information was achieved by implementing a two-tier thresholding technique. The two-tier threshold method was also designed to mitigate the impact of background noise on the output image. A morphological filter which also further minimizes the impact of noise was also implemented. Our algorithm also traces the boundaries of each isolated bitmap region. The presented boundary tracing algorithm also helps to distinguish between the different components of an image, and hence confine the space from which features that describe each image blob are extracted. Once the best minimal features that describe each image blob are extracted, a dynamic matching algorithm is then used to align the best matching blobs from different image frames. The blob matching process is also viewed as part of the image tracking process. The aligned image blobs are immediately classified using the SVM. The classification results are presented and discussed in Section 5.5.2.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Chapter 5

An evaluation of system performance

5.1 Introduction

Chapter 4 explains how different image segmentation algorithms isolate the region of interest. Once the objects that constitute an image are isolated from each other, computer vision algorithms extract features that best describe each image object. This chapter presents an evaluation of the different image processing modules that were implemented in this research. Section 5.3 discusses factors that determine the choice of a threshold value. A KLT based hand tracking method, which theoretically sounds a good alternative to the approach used in this thesis, is discussed in Section 5.2. In that section we assessed the applicability of the KLT feature tracker to the gesture recognition problem. Section 5.4 compares the results of single-cue and multiple-cue based segmentation methods.

5.2 KLT based hand tracking

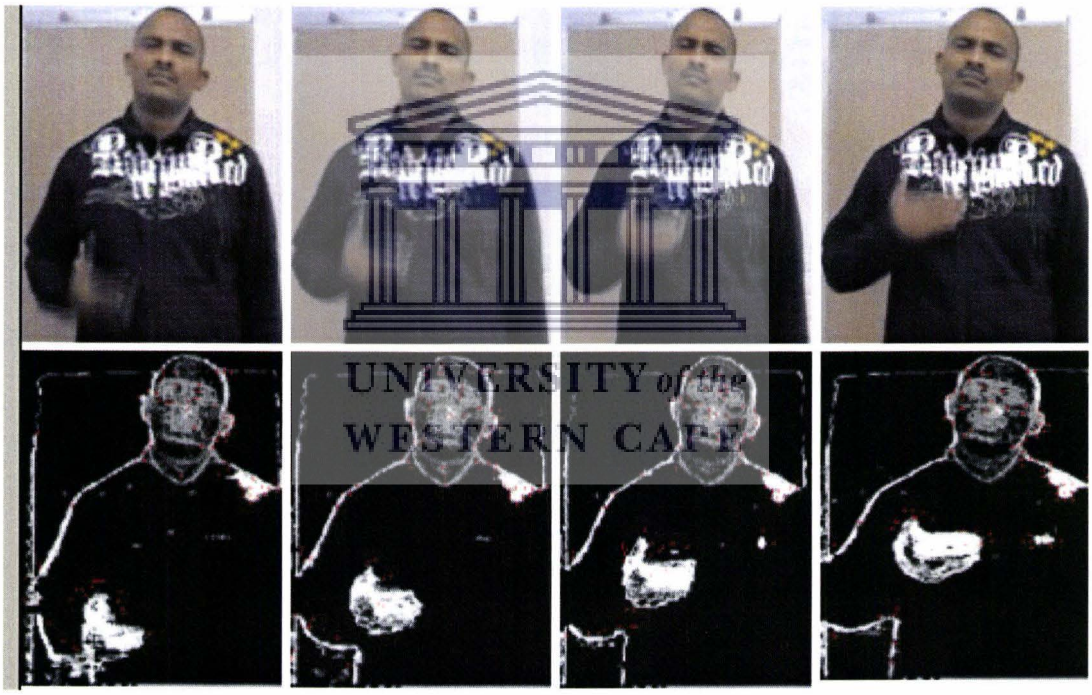
UNIVERSITY of the
WESTERN CAPE

In [38][92] optical flow based methods of tracking object through a sequence of video images are presented. The KLT feature tracker, implements the sliding window principle. Features which are interesting to track are often identified by small sized windows whose contents are carefully monitored during tracking [46]. Monitoring of the feature windows helps to optimize the feature tracking process. While bigger sized windows would be associated with more features, smaller sized windows make it possible to treat all points contained in a tracking window as if they constitute a single point. As such, those points are often viewed as if they move at the same speed.

From the KLT based tracking results shown in Figure 5.1, the red dots indicate the successfully tracked regions. It can be seen that in human images, most features which

https://etd.uwc.ac.za/

are considered good for tracking are found around the head region. Only a limited number of good features are found around the hand regions. This is mainly because the hand often moves too fast, sometimes it frequently changes its 2D orientation as it moves. Fast moving hands are often associated with large inter-frame distances, while rapidly changing hand appearances are associated with the occlusion problem. Because of the above mentioned constraints, the KLT feature tracker often fails to track fast moving hand regions. In order to minimize the loss of tracking information, either the signing hand must move very slowly; or very fast image capturing and processing devices must be used. However, in the spirit of retaining the naturalness with which people sign, no one should be compelled to perform gestures in slow motion. Besides, very high processing speeds are difficult to achieve, and consequently both conditions cannot always be guaranteed.



UNIVERSITY of the
WESTERN CAPE

Figure 5.1: Tracking the moving objects using the KLT tracker

https://etd.uwc.ac.za/

Most of the features that are considered good for tracking cannot sufficiently describe an object of interest. These features hardly provide information about object shape or its 2D projection, yet such information is vital when building automated object detection and tracking systems. The KLT feature tracker achieves commendable tracking results for slow moving, highly textured objects such as the human head, but it is not designed to recognize any object. Computer vision researchers say that objects can be recognized from their crude outlines [55][56], but using the KLT algorithm, the good features to track are not always associated with object outlines. Some researchers still claim that the KLT feature tracker can sufficiently track moving hands [92], but our sample tracking results demonstrate otherwise. In fact, our sample results demonstrate that the KLT feature tracker often fails to track gesturing hands in a highly textured background environment. In this research we track an object of interest based on a probabilistic blob matching technique. The probability of correctly aligning the image blobs is increased if the segmentation results adequately identify the object of interest. Using skin-colour or motion based segmentation, some segmentation results are often misclassified [94]. However, the classification error can be minimized but using optimum threshold values. The next subsection discusses how the choice of threshold values affects the segmentation process.

UNIVERSITY of the
WESTERN CAPE

5.3 Choice of threshold values

The effectiveness of a threshold value, *TH*, can be inferred from the magnitude of false detections and false dismissals that it produces. Hualu and Shih-Fu [9], demonstrated that extremely low skin-colour threshold values stimulate high rates of false detections and low rates of false rejections. This observation is also supported by Zhu, Yang and Waibel [94]. As the threshold values increase, the number of false detections decreases. Conversely, the rate of false dismissals increases as the threshold value is increased. Figure 5.2 is a graphical illustration of the effects of increasing skin-colour thresholds, *TH*, on the rates of false dismissals and false rejections. Generally it is very difficult to find a threshold value that simultaneously eliminates false dismissals and false

https://etd.uwc.ac.za/

detections. As a result, most segmentation algorithms never achieve perfect segmentation.

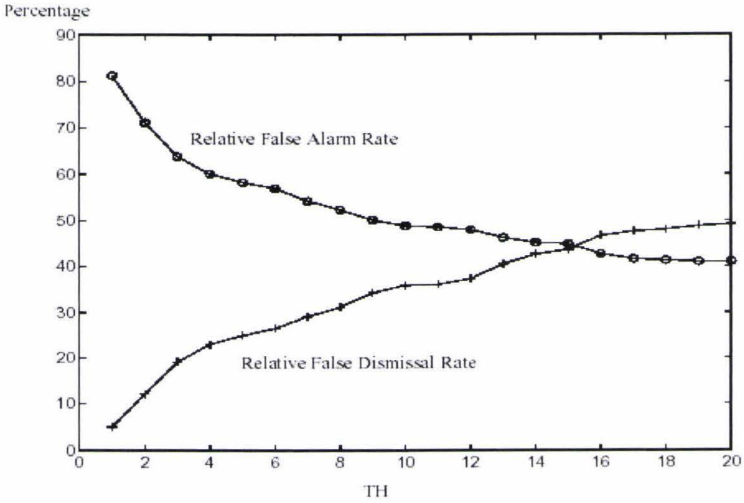
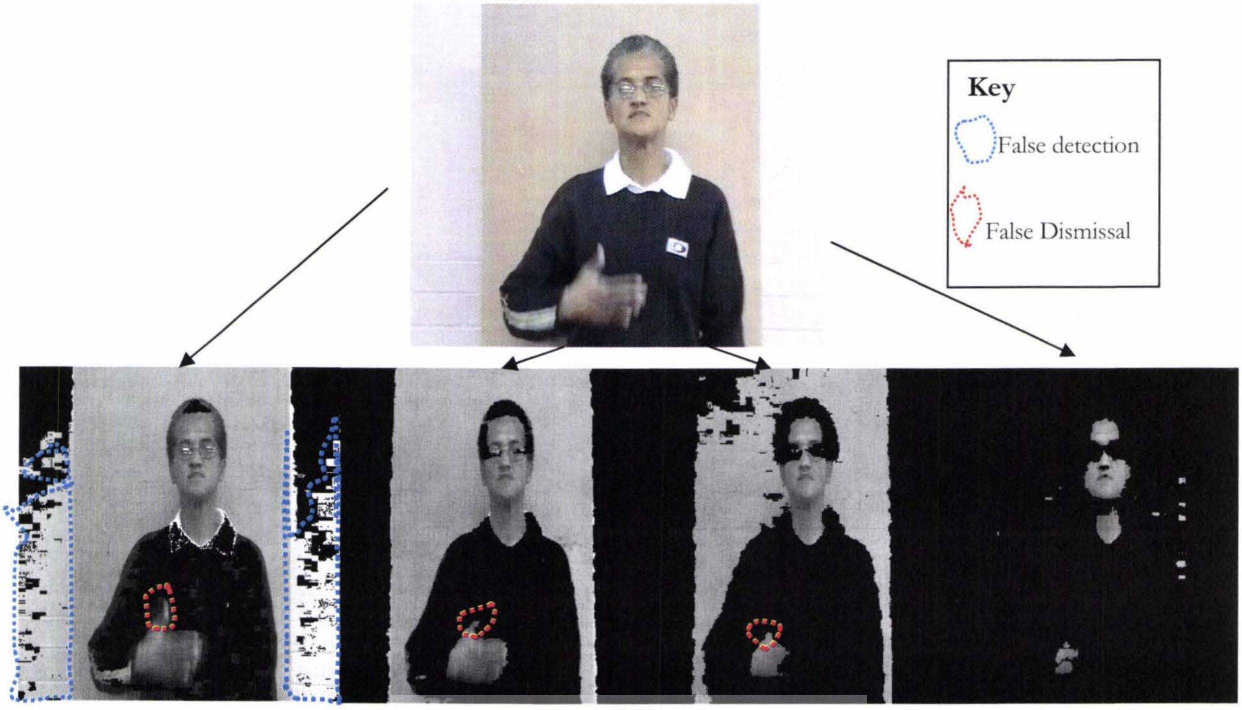


Figure 5.2: Performance of skin colour threshold values: Adapted from Hualu and Shih-Fu [9]

Chai and Ngan [47] say that in the YCrCb colour space, Cr values ranging from 133 to 173 and Cb values ranging from 77 to 127 are suitable for isolating skin-colours of people of different ethnicity. However, in this research we found that Cr values ranging from 128.5 to 173 and Cb values ranging from 77 to 133 are more appropriate for isolating skin-colours. A pictograph of the results we obtained using different threshold values for Cr is given in Figure 5.3 (a). In this figure, all the non-skin coloured regions which were falsely classified as skin-coloured regions are enclosed by a blue dotted lines and the red line outlines the falsely dismissed regions. A closer look at the given output image sequences shows how some sections of an extended thumb were gradually discarded as the threshold values of Cr was increased.

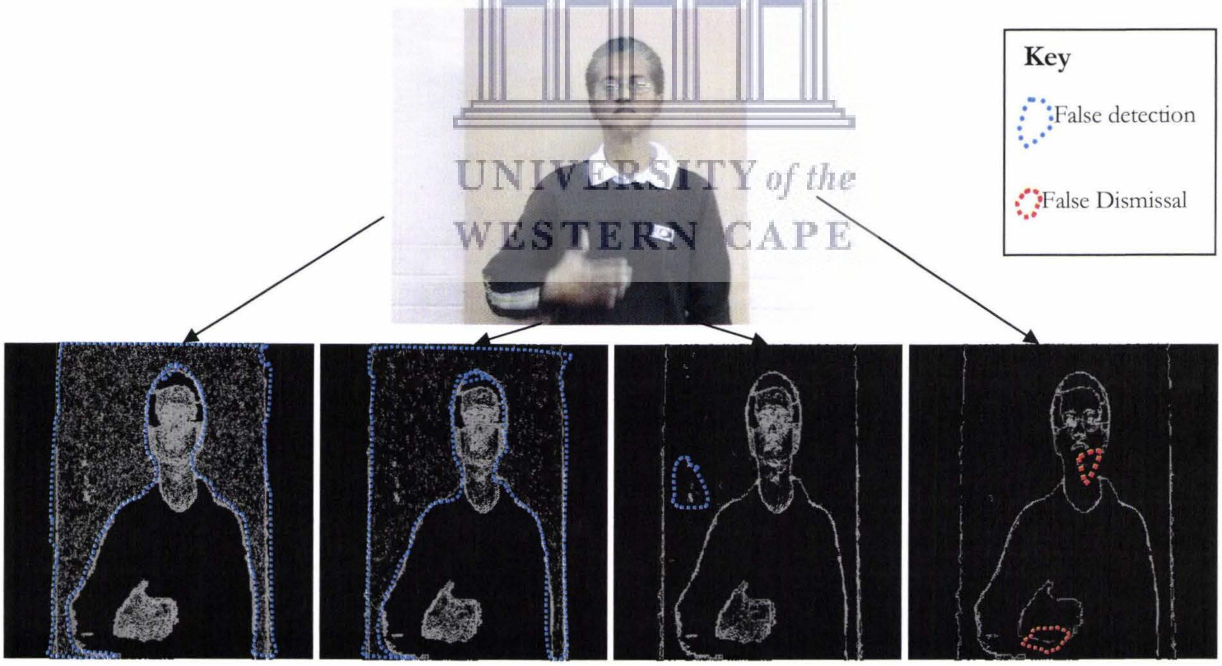
UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/



128≤Cr   128.5≤Cr   129≤Cr   129.5≤Cr

Figure 5.3 (a): Effects of increasing (decreasing) Cr values on the segmentation results

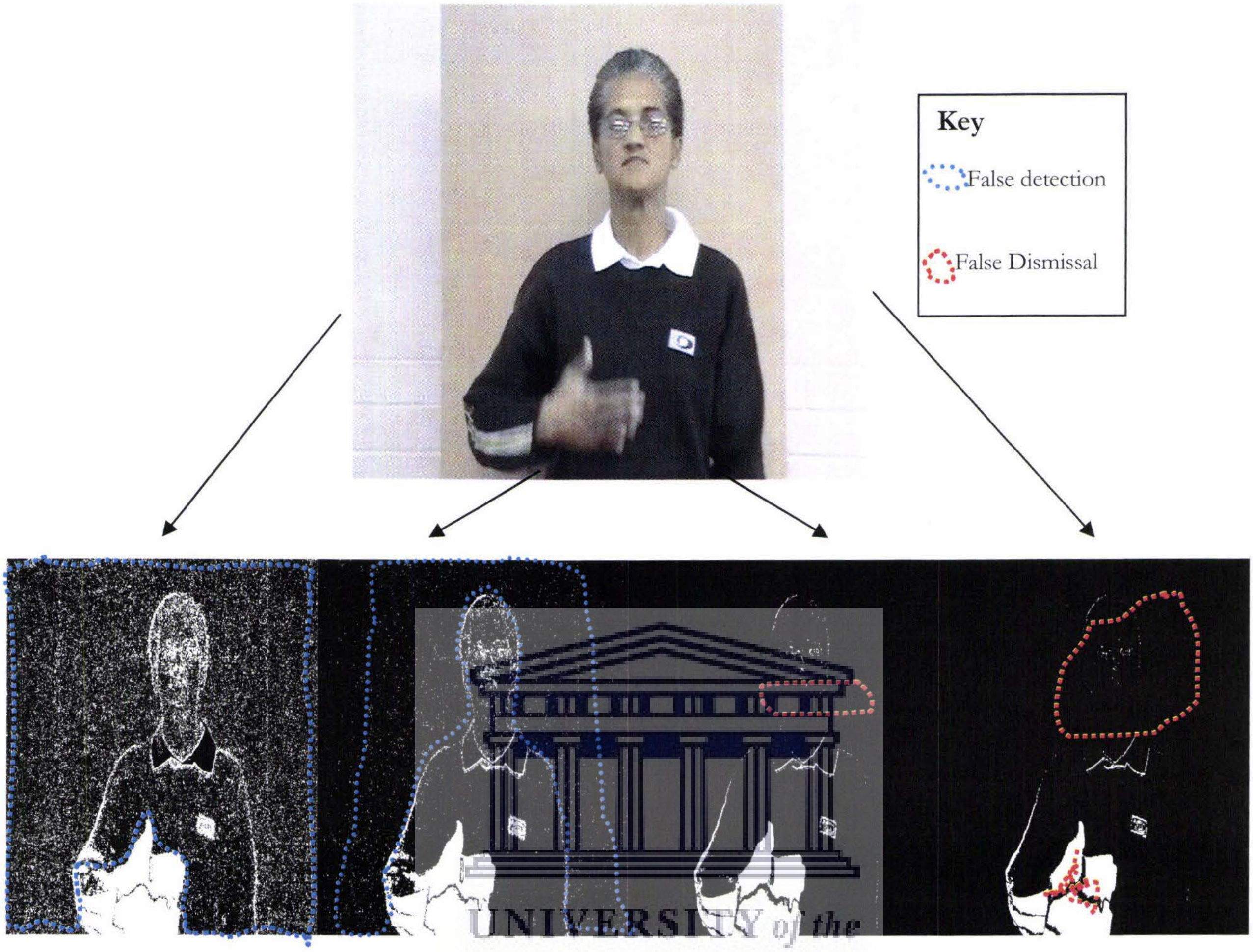


TH≥15   TH≥20   TH≥30   TH≥80

Figure 5.3 (b): Effects of increasing (decreasing) edge threshold values on the segmentation results

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/



Motion≥3 Motion≥6 Motion≥15 Motion≥20

UNIVERSITY of the
WESTERN CAPE

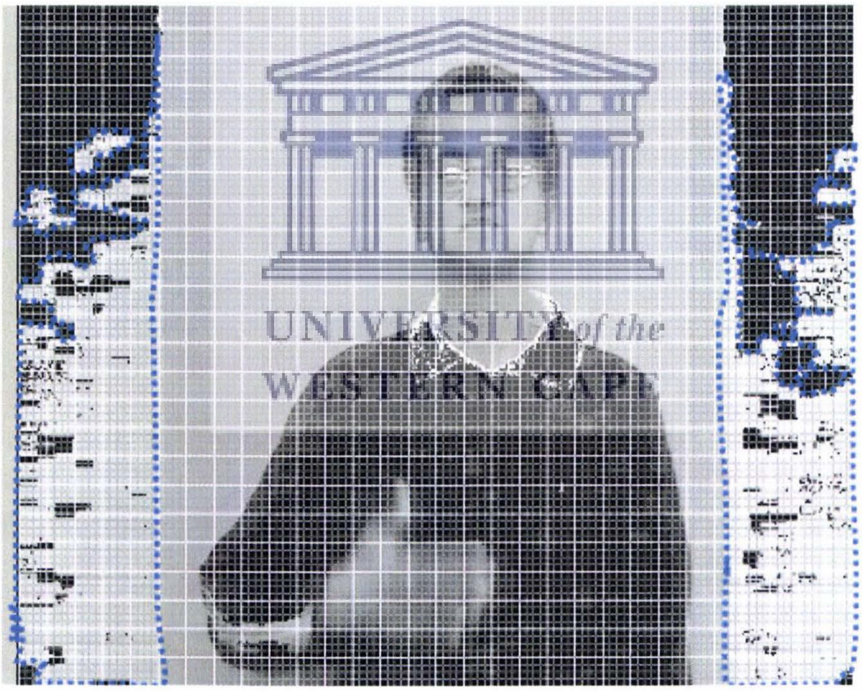
Figure 5.3 (c): Effects of increasing (decreasing) motion threshold values on the segmentation results

It is not easy to accurately quantify the falsely detected and falsely dismissed regions as some blobs are composed of many small chains of interconnected sub-regions. However, the total surface area, *A*, covered by each misclassified region, can be approximated as shown in equation 5.1.

$$A = \int_{a}^{b} f(x)dx \approx \sum_{x_i = a}^{n} h_{x_i} \tag{5.1}$$

https://etd.uwc.ac.za/

where n represents the number of equally spaced sub-intervals that falls within the interval $x_i \in x: a \leq x_i \leq b$. The number of 6x6 or 24x24 misclassified sub-regions, h_{x_i}, is obtained by manually counting the number of misclassified sub-regions that are contained in each sub-interval. The sum of n subtotals of h_{x_i} is used as the estimate of the area covered by falsely detected skin-colour regions. We use 24x24 sub-regions when the area covered by the misclassified regions is too big and when the density of misclassified skin-coloured pixels in each sub-region is very high. Figure 5.4 illustrates how the image is broken down into sub-regions. The area covered by each misclassified region is used for calculating the rate of false detections and the rate of false rejections that are associated with each threshold value. Data that describes the effects of different Cr threshold values is shown in Tables 5.1 (a) and (b).



UNIVERSITY of the
WESTERN CAPE

Figure 5.4: Determining the area covered by false skin regions for Cr≥128

https://etd.uwc.ac.za/

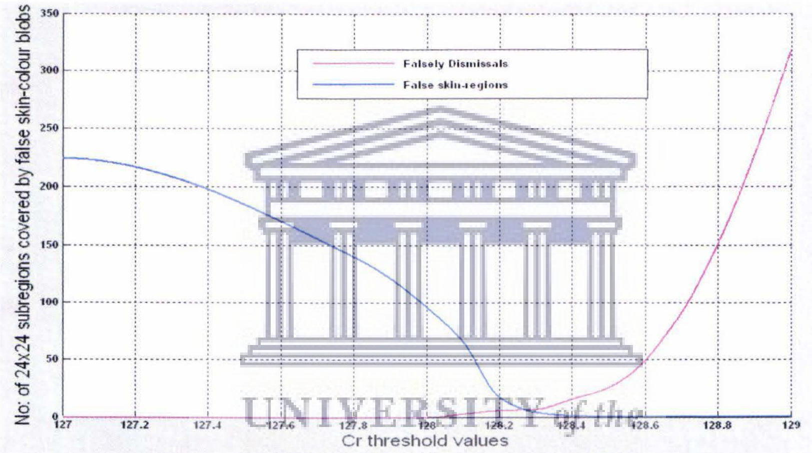
Table 5.1: Relationship between Cr values and the sizes of falsely detected or falsely rejected skin-coloured regions

(a) The Effects of Cr values on the magnitude of false skin-coloured detections

Cr Threshold Values	Estimated quantity of false skin colour regions (No: of 24x24 sub-regions)
Cr≥127	224
Cr≥127.8	140
Cr≥128.1	65
Cr≥128.2	15
Cr≥128.3	4
Cr≥128.4	1
Cr≥128.5	0.4
Cr≥129	0

(b) The effects Cr values on the magnitude of false rejections of skin-coloured regions

Cr Threshold Values	Estimated size of falsely rejected skin colour regions (No: of 6x6 sub-regions)
Cr≥127	0
Cr≥128	0
Cr≥128.1	3
Cr≥128.2	6
Cr≥128.3	7
Cr≥128.4	11
Cr≥128.5	26
Cr≥129	320



UNIVERSITY of the WESTERN CAPE

Figure 5.5: Relationship between Cr threshold values and rates of false detections and false dismissals

The effects of increasing (decreasing) Cr value on the rate of false detection and/or false dismissal is implicitly explained by Figure 5.5. In this figure, the blue line shows how the area covered by false skin-coloured regions decreases as the Cr value increases. The red line illustrates how the rate of false rejection increases with the increasing values of Cr. For the data sample represented in Figure 5.5, the rate of false detection and the rate of false dismissal are extremely low for Cr values ranging from 128.2 to 128.5. However, in general we found that Cr values which are greater than or equal to 128.5 are suitable for isolating skin-colours of people of different ethnicity. In

https://etd.uwc.ac.za/

Figure 5.3 (a), the increasing rates of false dismissals are represented by the diminishing surface area of the extended thumb and/or the diminishing size of the hand projection. When the Cr value is increased, some segments of the hand and/or the head projections are gradually lost and eventually the whole region disappears altogether.

We investigated how different motion threshold values affect the rate of false motion detections and the rate of false rejections of true motion values. If an object is moving slowly, the current and the previous object position may overlap. Bearing in mind the fact that the brightness function for different image pixels that belong to a smooth surfaced object often assumes a uniform distribution, $D(x, y, t) = a$, it is obviously not easy to implement the difference image for isolating the moving object.

In general it is easy to identify false movements since they are mainly identified outside context of an object, but the process of identifying falsely dismissed moving regions is a bit complex. Although some of the falsely dismissed motion values are often contained within the region of interest, such regions are largely ignored in this work. We use object boundary information to delimit the desirable region, but it is not easy to identify a misclassified region which falls within the boundaries of an accepted region.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

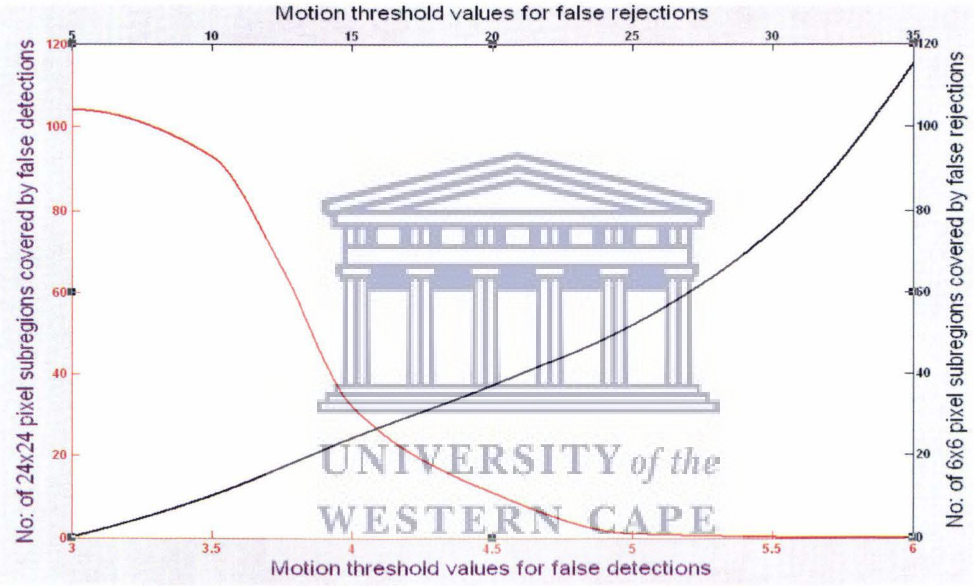
Table 5.2: Relationship between motion values and the sizes of falsely detected or falsely rejected moving regions

(a) The Effects of Motion thresholds on the magnitude of false motion detections

Motion Threshold Values	Estimated quantity of false motion regions (24x24 sub-regions)
Motion≥3	104
Motion≥3.5	93
Motion≥3.7	66
Motion≥4	32
Motion≥4.5	11
Motion≥5	1
Motion≥6	0

(b) Effects of Motion thresholds on the magnitude of false rejections of true motion values

Motion Threshold Values	Estimated quantity of false motion regions (24x24 sub-regions)
Motion≥5	0
Motion≥10	10
Motion≥15	24
Motion≥20	37
Motion≥25	52
Motion≥30	75
Motion≥35	115



UNIVERSITY of the
WESTERN CAPE

Figure 5.6: Relationship between motion threshold values and rates of false detections and false dismissals

Tables 5.2 (a) and (b) respectively show how different motion threshold values affect the rate of false motion detection and the probability of rejecting true motion values. The relationships between motion threshold values and the probability of rejecting true motion values and the relationship between motion threshold values and the rate of false motion detections are summarized in Figure 5.6. Unlike in the case of Cr values, there is no obvious range of motion values that simultaneously minimizes the rates of

https://etd.uwc.ac.za/

false detections and false dismissals. However, Figure 5.6 clearly shows that low motion values induce high rates of false motion detections and low rates of true motion dismissals. In this research we chose motion threshold values that minimize background clutter and also reduce rate of false rejections of true motion values.

Table 5.3: Relationship between edge threshold value and the quantity of falsely detected or falsely rejected edges

(a) The effects of edge threshold values on the quantity of falsely detected edges

Edges Threshold Values	Approx. no: of false edges (no: of 24x24 sub-regions covered)
Edge≥20	107
Edge≥22	93
Edge≥25	54
Edge≥26	17
Edge≥27	7
Edge≥28	3
Edge≥29	2
Edge≥30	1

(b) The effects of edge threshold values on the quantity of falsely rejected edges

Edges Threshold Values	Falsely dismissed edges (no: of 24x24 sub-regions covered)
Edge≥30	0
Edge≥35	5
Edge≥40	19
Edge≥50	53
Edge≥55	75
Edge≥60	96
Edge≥65	117
Edge≥70	127

UNIVERSITY of the
WESTERN CAPE

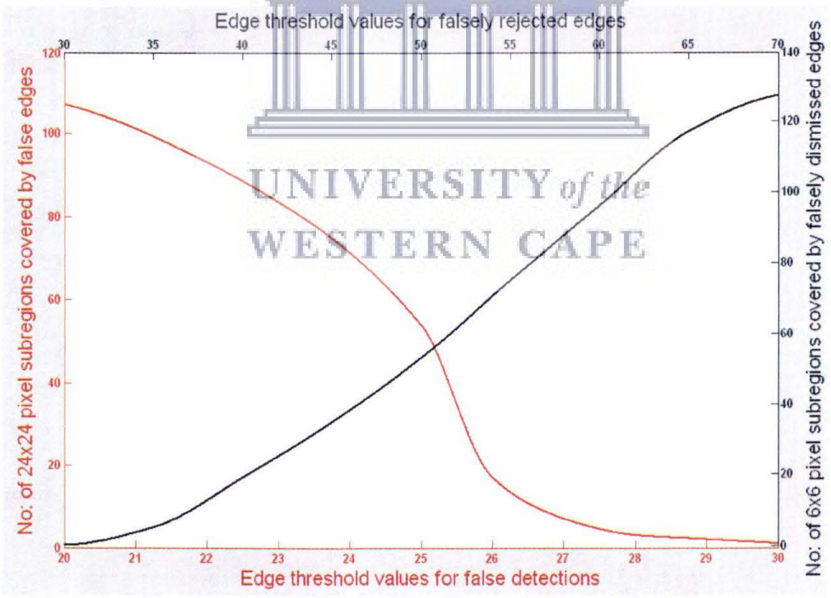


Figure 5.7: Relationship between edge threshold values and rates of false detections and false dismissals

https://etd.uwc.ac.za/

Tables 5.3 (a) and (b) illustrate how different edge threshold values affect the rate at which false edges are generated or the rate at which true edges are discarded. It is important to remember that there are no surrogates of truth for all segmentation results [93]. That is the reason why it is very difficult to distinguish between true and false edges. In this research, we mainly review the edges that lie along the boundaries of an object. We neither reject nor accept any edges that are found within a region of interest. However, all those edges that generated outside the confines of any known rough textured object are regarded as false edges. Many of such edges are generated by low edge threshold values (see Figure 5.3 (b)). Rapidly changing colour values also give rise to image edges, and although such edges do not constitute object boundaries, they are not necessarily false edges. The number of falsely detected edges is estimated by manually counting the number of sub-regions that contain false edges. Although the density of edge based pixels varies from sub-region to sub-region, all edge based sub-regions are assumed to contain the same number of edge-based pixels. As a result, the numbers of falsely detected edge-based pixels given in Table 5.3 (a) may be an overestimation of the true edge values. However, we are mainly interested in observing how different threshold values affect the rate of false detections and the rate of false dismissals. The number of falsely dismissed edges is approximated using the number of boundary based pixels that were discarded from the image data.

UNIVERSITY of the
WESTERN CAPE

Increasingly higher edge detection threshold values are associated with increasing rates of false dismissals and decreasing rates of false detections. These trends are similar to those illustrated in Figure 5.2. In this research, moderately low motion and edges threshold values are used for segmenting image objects. Low threshold values are preferred since they minimize the rate of false dismissals. False dismissals might cause loss of important object information, which often lead to total failure of the object recognition process.

https://etd.uwc.ac.za/

5.4 Segmentation results

In Section 3.2.2.1, we mentioned that single-cue based segmentation results are too coarse for gesture recognition systems. The single-cue based segmentation results shown in Figures 5.8, 5.9 and 5.10 further confirm this assertion. In these figures, the input image is given in the first row, while sample results for skin-colour, edge and motion based segmentation are given, in the same order, in each middle row. The bottom row shows the output of a multi-cue based segmentation process. In these diagrams all regions that are shaded in white represent the segmented image blobs and all rejected regions are shaded in black. Skin-colour based segmentation results clearly show that many non-skin objects are often classified as skin-coloured regions. In practice, not all skin-coloured objects actually represent true skin-regions. This observation also supports the postulation that under complex imaging conditions skin-colour alone cannot unambiguously identify the object of interest [54].

The motion history images represent the difference in intensity between consecutive pairs of image frames. Motion history images are represented in the extreme right of each middle row of Figures 5.8, 5.9 and 5.10. A careful analysis of the motion maps shown in these figures reveals that MHIs cannot reliably identify an object of interest. In fact, MHIs merely represent disparities in lighting intensities between each pair of corresponding image pixels, and hence the so-called 'motion maps' do not always represent the object motion. For instance, MHIs often represent fluctuating lighting conditions or other moving background scenes as if they constitute object motion. In Figures 5.8 (d) and 5.9 (d), the white pixels found around the chest regions are probably generated from the movement of some loosely fitting garments worn by the signer. Loose fitting garments are prone to shake under windy conditions or in response to body movements. In all motion based segmentation results, these irrelevant movements constitute image noise, which sometimes is often misconstrued for object motion. Depending on the chosen motion threshold values, the effects of image noise are sometimes very devastating. In order to alleviate the impact of image noise, a

UNIVERSITY of the WESTERN CAPE

https://etd.uwc.ac.za/

filtering technique that eliminates all small isolated image blobs is implemented in this research. Even in situation where image noise is significantly low, MHIs always represent the current and the previous object positions. If the current and the previous object positions overlap, MHIs represent the two positions with one big silhouette which sometimes have a faint or even a wide crack along the region of overlap. Such a representation does not actually represent the underlying object's 2D appearance. General knowledge of object shape is crucial for gesture recognition applications. Nonetheless, MHIs still provide important though crude, information about object motion. For instance, in Figures 5.8 (d), 5.9 (d) and 5.10 (d), the magnitude of motion exhibited by each object can be approximated by either estimating the area of overlap, which in some way represents the translation vector that transforms an image blob from its previous position to its current position.

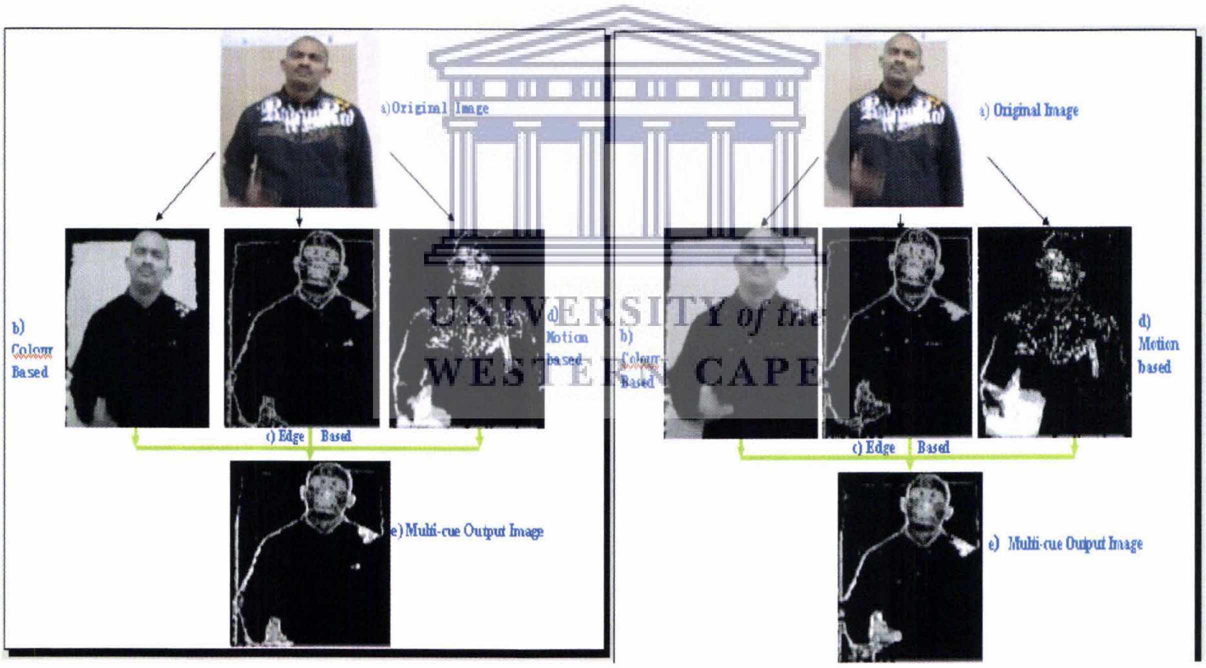
In Figures 5.8, 5.9 and 5.10, the central image represents the detected image edges. In Section 4.2 we mentioned that image edges do not always conform to object boundaries. The magnitude of an image edge largely depends on the intensity difference between a pixel and some of its neighbourhood pixels. Occlusion and object speed are some of the factors that affect the continuity of the edge. In regions where non-continuous or overlapping edges are concentrated, object boundaries are difficult or almost impossible to trace. Most edge detectors are not designed to control edge thickness. In our case, very thick edges are obtained around the face region wherever low threshold values are used. Thick edges that are found over the head region are presumably caused by the numerous wrinkles or other depressions that are commonly found on human faces.

UNIVERSITY of the
WESTERN CAPE

Our multi-cue based segmentation approach first assumes that all moving skin-coloured pixels constitute the interesting regions. Considering the fact that low threshold values were recommended in Section 5.3 the multi-cue approach may sound as if it ignores image noise. However, logical ANDing of edges, motion and skin-colour information help to eliminate some of the unnecessary regions. It can be seen

https://etd.uwc.ac.za/

from Figures 5.8, 5.9 and 5.10 that the results of a multi-cue based segmentation method basically contain less noise and few background objects in comparison with either motion or colour based segmentation results. Although the edge based segmentation results look comparatively similar to those obtained from a multi-cue segmentation, it must be remembered that edges alone cannot sufficiently segment image objects since edges do not always conform to continuous curves that characterize object boundaries. Besides, edges are often found in non-skin coloured regions yet it is absurd to assume that such an edge belongs to a face or a hand region. Logical ANDing of edge information with skin-colour cues help to eliminate some of undesired edges. Object motion is incorporated into skin colour information in order to discard some of the stationary skin-coloured regions. Stationary skin-coloured regions do not constitutes a dynamic gesture; hence no further processing will be required for such regions.

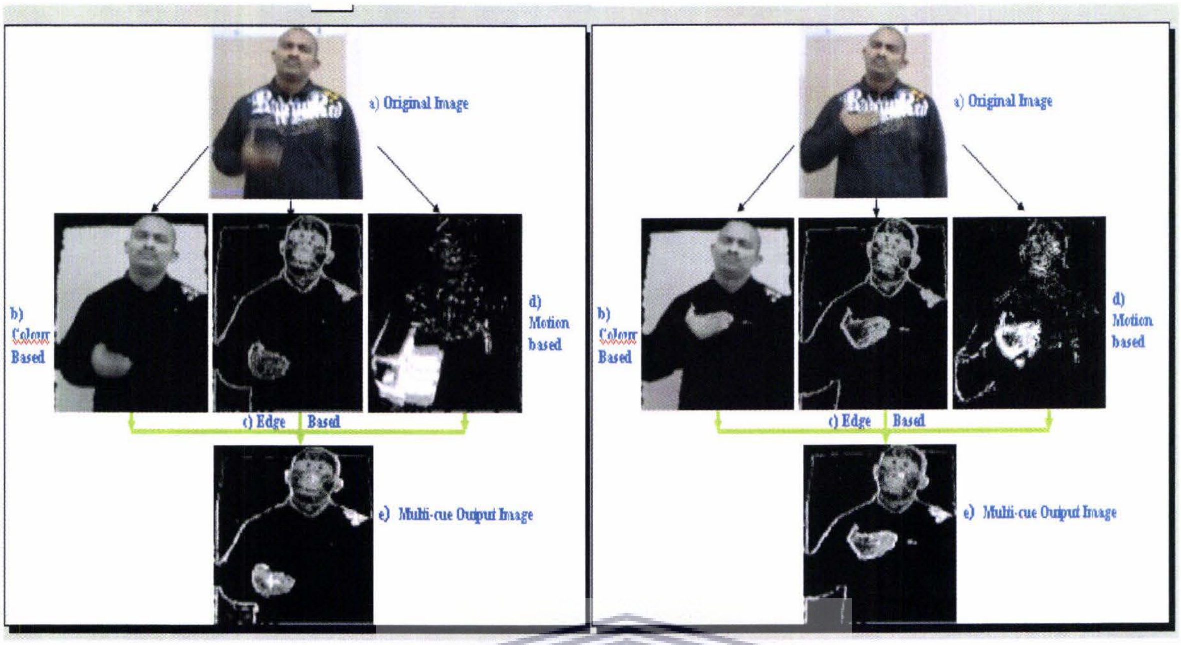


UNIVERSITY of the WESTERN CAPE

A) Frame 1 B) Frame 2

Figure 5.8: Segmentation results for frames 1 and 2

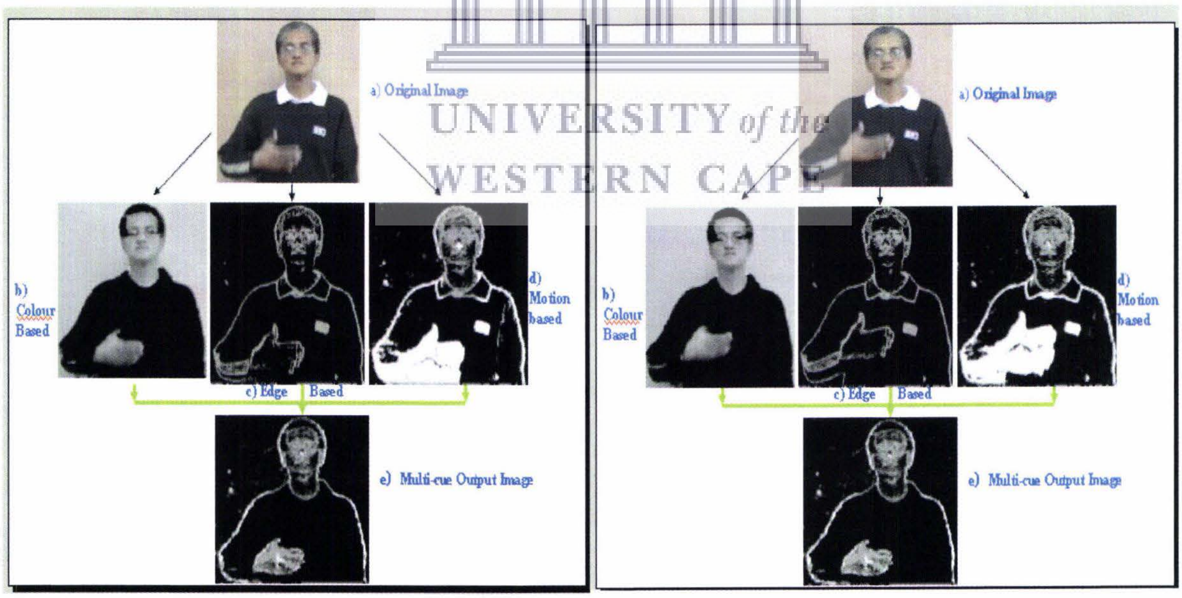
https://etd.uwc.ac.za/



A) Frame 3 B) Frame 4

Figure 5.9: Segmentation results for frames 3 and 4

UNIVERSITY of the
WESTERN CAPE



A) Frame 1 B) Frame 2

Figure 5.10: Segmentation results for frames 1 and 2

https://etd.uwc.ac.za/

5.5 An assessment of the segmentation results

Section 5.4 presents samples of the segmentation results that are produced by our system. However, it is not possible to tell whether these results are adequate for a chereme based gesture recognition system. Although sign language linguistics have shown that hand-shapes information, hand positions and hand movement patterns can reliably identify individual sign language gestures [11][15][53], adequacy of the hand segmentation process can only be confirmed using qualitative methods. According to Jayaram *et al.* [93], any analysis of the performance of any segmentation algorithm must take into account the application domain for which it is designed to serve and the accuracy of the low-level process of delineating the object boundaries. Each application domain is defined in terms of the task to be performed, *A*; the relevant 3D space over which the application draws meaning, *B*; and the imaging approaches implemented by the application, *I*. An evaluation of the performance of a segmentation algorithm must qualify both the low-level process of delineating object boundaries and the higher level process of recognizing the tasks at hand [93]. Jayaram *et al.* further argue that whereas machines outperform humans in such tasks as delineating object boundaries, the human eye's aptitude to recognize images is much higher than that of machines. In the next two subsections, we present a qualitative evaluation of the performance of the low-level segmentation and the hand-shape recognition processes.

UNIVERSITY of the
WESTERN CAPE

5.5.1 Evaluating the low-level hand segmentation process

Generally, no surrogates of truth exist for low level segmentation results [93], and hence no universally acceptable yardsticks for quantifying the delineation process are available. Irrespective of the absence of universally acceptable yardsticks, this research qualifies the performance of the presented low level segmentation algorithm basing on the following attributes: the smoothness of the delineated boundaries; the amount of manual intervention required; and the volumes of falsely dismissed and falsely detected regions that are associated with each output image. In the contexts of this research, the

https://etd.uwc.ac.za/

segmentation results are considered smooth only if the delineated boundaries are not very rough and/or discontinuous. Erosion and/or dilation are some of the processes that contribute to rough boundaries. Table 5.4 describes the five categories that are used to classify object boundaries, magnitude of falsely detected regions and the rate of false dismissals. The number of sub-regions that are covered by the misclassified regions and/or the number of candidate hand regions that are obtained after processing one handed input gestures are also used as a measure of the effect of each of the two parameters on the overall segmentation results.

Table 5.4: Describing categories used to classify segmentation results

	Very High	High	Moderate	Low	Negligible
False Detections (No: of candidates)	12-20	9-11	6-8	3-5	0-2
Broken boundaries (No: of 6x6 pixel)	16-25	12-15	8-11	4-7	0-3
Jagged boundaries (No: of 6x6 pixel)	24-50	18-23	12-17	6-11	0-5
False Dismissals (No: of 6x6 pixel)	16-50	12-15	8-11	4-7	0-3

The other parameters are classified based on the criteria described below.

UNIVERSITY of the
WESTERN CAPE

The level of *manual intervention* is considered:

- *negligible* if the segmentation process can proceed on its own once the process is initiated;
- *low* if only one parameter must be adjusted whenever a new input image is processed;
- *moderate* if 2 or 3 parameters must be adjusted whenever a new image is used;
- *high* if between 4 to 10 parameters must be adjusted for each new input;
- *very high* if at least one parameter is adjusted at every processing stage.

The *overall segmentation result* is of:

- *very high* quality if

https://etd.uwc.ac.za/

    - all the boundaries of the output image are smooth;
    - there are no falsely dismissed regions;
    - the amount of image noise is negligible;
    - there is no manual intervention required
- *high* quality if
    - the rate of false detections, the rate of false dismissals, the nature of produced image boundaries and the level of manual intervention are either low or negligible.
- *moderate* quality if
    - either the amount of false detections or the amount of false dismissals are moderate
    - the number of jagged object boundaries is moderate
- *low* quality if
    - either the amount of false detections or the amount of false dismissals are high
    - the number of jagged object boundaries is high
- *negligible* quality if
    - if at least one of the observed characteristics is very high

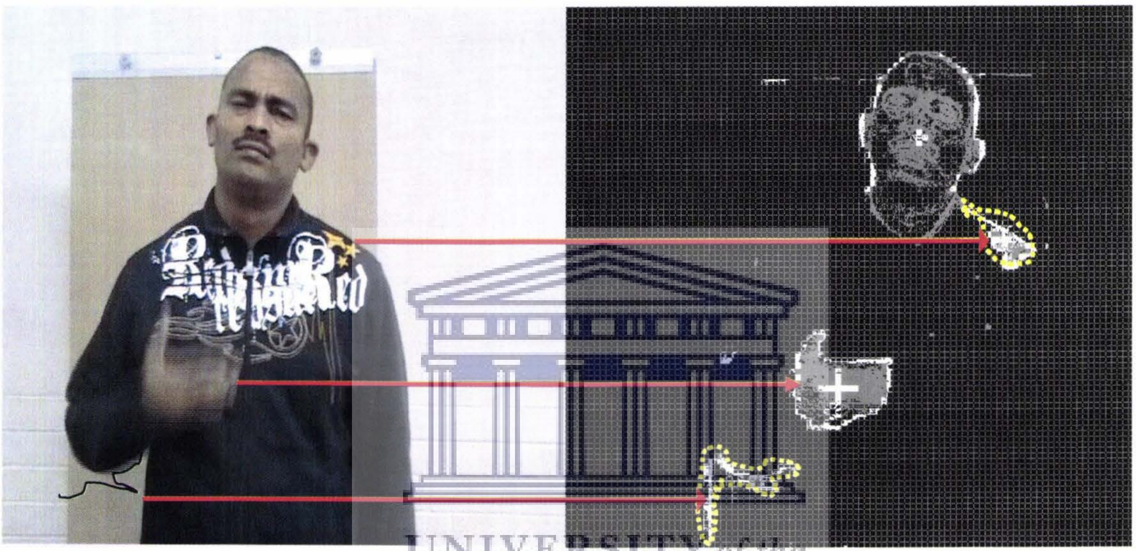
UNIVERSITY of the
WESTERN CAPE

Table 5.5 presents our assessment of the quality of the sample of segmentation results shown in Figure 5.11. Although some information about the extended finger has been lost, we are convinced that high quality segmentation is produced. The loss of some fingertip information is not brought about by any flaws in the segmentation algorithm, but is precipitated by the blurring effects suffered by the moving hand.

https://etd.uwc.ac.za/

Table 5.5: Observed characteristics of a multi-cue segmentation result shown in Figure 5.12

	Very High	High	Moderate	Low	Negligible
False Detections				✓	
Broken boundaries					✓
Jagged boundaries				✓	
Manual Intervention					✓
False Dismissals				✓	
Segmentation Result		✓			



UNIVERSITY of the

Figure 5.11: Segmentation results for a blurred input image

WESTERN CAPE

The images used in this research were extracted against a skin-coloured background. In this case, the skin-coloured background amplifies the boundaries of every moving object irrespective of whether the object is skin-coloured or not. However, some of the irrelevant boundaries are later discarded by the boundary tracing algorithm, and hence they do not affect the quality of the segmentation results. Coarse information that describes the possible hand configurations is incorporated into the segmentation algorithm in order to eliminate some of the moving background objects. The yellow dotted-lines, shown in Figure 5.11, outline some of the candidate hand regions. All

https://etd.uwc.ac.za/

image blobs that are characterized by non-persistent features and those which are almost stationary are eliminated.

A generalized notion of the accuracy of our low-level hand segmentation process can be obtained by scrutinizing the output results for at least 20 output samples that are extracted from images of 5 people. Here a small sample size is used since we realized that there is little variability between the different output samples. The observed parameter values for all the samples are summarized in Table 5.6.

Table 5.6: Observed attributes of a samples of 20 output images

	Number of images in each category				
	Very High	High	Moderate	Low	Negligible
False Detections			2	15	3
Broken boundaries				1	19
Jagged boundaries				7	13
Manual Intervention					20
False Dismissals			1	7	12
Segmentation Result		18	2		

UNIVERSITY of the
WESTERN CAPE

A weighted average of false detections, Av_{fd}, for n output samples is calculated as follows:

$$\mathrm{Av_{fd}} = \frac{\left(a.M_{VH} + b.M_{H} + c.M_{M} + d.M_{L} + e.M_{N}\right)}{n}$$

where a, b, c, d and e are the total numbers of data samples that contain very high, high, moderate, low and negligible numbers of falsely classified blobs respectively. The symbols, M_{VH}; M_{H}; M_{M}; M_{L} and M_{N} represent the median values for each range of values that are described in table 5.4. For instance, for the 20 output samples described in Table 5.6, the average class for all falsely detected regions is:

https://etd.uwc.ac.za/

$$Av_{fd} = \frac{(0\times16+0\times10+2\times7+15\times4+3\times1)}{20}$$

$$= 3.85$$

$$= 4 \text{ (rounded up to the next whole number)}$$

According to the category descriptions given in Table 5.4, the weighted average for falsely detected blobs lies within the 'low class' category. These findings coincide with the modal class for the false detections shown in Table 5.6. The weighted class for falsely dismissed regions, broken boundaries and/or the weighted class for jagged boundaries are computed in a similar way. In general, the weighted average category for manual interventions or for the overall quality of the segmentation results is considered to be equal to the modal class of each respective parameter.

5.5.2 Analyzing the hand tracking and recognition processes

The consistency of the segmentation algorithm can be inferred from the performance of the tracking and the hand-shape recognition algorithms. For each sequence of matched image blobs, we determine the number of correctly aligned blobs and the number of falsely tracked blobs. In the case of one handed gestures, a hand blob is falsely tracked only if a non-hand region is aligned to a hand region.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Table 5.7: Analysis of blob matching results

		Processed Frames (P)	Tracked Frames (T)	Lost Frames (L)	Falsely Tracked Frames (F)	Tracking Rate $\frac{T}{P} \times 100\ \%$	Consistency $\left(1-\frac{F}{T}\right) \times 100\ \%$
Person 1	Gesture 1	40	39	1	0	97.5	100
	Gesture 2	63	60	3	0	95.2	100
	Gesture 3	53	45	8	1	84.9	97.8
Person 2	Gesture 1	25	25	0	0	100	100
	Gesture 2	56	54	2	0	96.4	100
	Gesture 3	44	44	0	0	100	100
Person 3	Gesture 1	43	43	0	0	100	100
	Gesture 2	41	41	0	0	100	100
	Gesture 3	33	33	0	0	100	100
Person 4	Gesture 1	21	21	0	0	100	100
	Gesture 2	48	48	0	0	100	100
	Gesture 3	31	31	0	0	100	100

Table 5.7 presents an analysis of the hand tracking results. Table 5.8 explains how blob matching results for person 4 gesture 1, shown in Table 5.7, were obtained from the feature values shown in Appendix B.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Table 5.8: Automatic blob matching process for data shown in appendix B

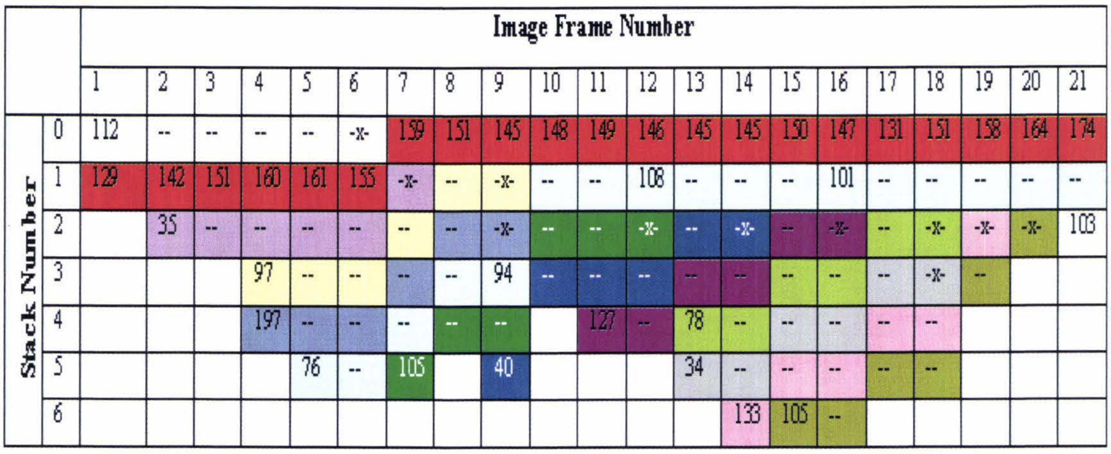


		Image Frame Number																				
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Stack Number	0	112	--	--	--	--	-x-	159	151	145	148	149	146	145	145	150	147	131	151	158	164	174
	1	129	142	151	160	161	155	-x-	--	-x-	--	--	108	--	--	--	101	--	--	--	--	--
	2		35	--	--	--	--	--	--	-x-	--	--	-x-	--	-x-	--	-x-	--	-x-	-x-	-x-	103
	3				97	--	--	--	--	94	--	--	--	--	--	--	--	--	-x-	--		
	4				197	--	--	--	--	--		127	--	78	--	--	--	--	--			
	5					76	--	105		40				34	--	--	--	--	--			
	6														133	105	--					

Symbols:

-- Continue from the last given value contained in similarly shaded squares

-x- Next frame is the termination point if the current blob sequence does not find a match

UNIVERSITY of the
WESTERN CAPE

In Table 5.8, the surface area of each candidate hand region is represented by a numerical value that corresponds to a particular stack number that is produced by each image frame. The red coloured sequence is the only one in which one blob from every image frame is successfully matched to all other correctly aligned blobs from the previous frame sequences. Manual verification of the red coloured sequence shows that only true hand blobs are successfully matched in this sequence, hence the blob matching and the blob tracking processes are 100% consistent for this particular output sample. In Table 5.8, unsuccessfully matched blob sequences are marked with -x- at one end. All blob sequences that have non-persistent features are eliminated from the stack list. Whenever a blob sequence with a higher sub-index value of a stack list identifier, *X[i]*, is discarded, the algorithm reduces by 1 all the sub-index values of all stack list identifiers which are greater than the sub-index of the eliminated stack list identifier. Besides, whenever a new non-matching blob is added to the stack list, the sub-index of the last stack list identifier increases by 1. Spontaneous increases and/or decreases in sub-index values explain the irregular patterns shown in Table 5.8.

https://etd.uwc.ac.za/

Irrespective of whether the same hand-shape is maintained during the execution of a gesture, the surface area covered by each of the aligned image blobs often changes from frame to frame. For instance, the surface area of the image blobs that constitutes the red coloured sequence (shown in Table 5.8) ranges from 129 to 174 pixels. In dynamic gestures, changes in blob surface areas simply reflect the different hand projections that are either caused by the changing camera viewpoints or by the presence of image noise at a particular imaging position.

5.5.3 Verifying the hand-shape recognition processes

As a way of verifying the identity of hand-shapes that constitute each aligned sequence of candidate hand regions, feature sets that describe each of the aligned image blobs are fed into the SVM. The training and the testing data samples are extracted as explained in Section 4.7. Blob classification is achieved by implementing a SVM with a linear kernel [14]. The SVM kernel is often fine tuned in order to make the best trade-offs between the function that describes the shape of the separating surface and the function that determines the smoothness of the object surface. Table 5.9 shows the different recognition rates that were achieved for either the person-dependent or person-independent situations. According to Table 5.9, out of four different gestures, the SVM based system achieved the lowest accuracy represented by a sensitivity (SE) of 0.8204 and a specificity (SP) of 0.7604. This low classification rate was observed for gesture 2 only. For other gestures the SE and SP values are much higher. This demonstrates that the SVM system is capable of accurately recognizing the hand gestures that were processed by our system. The performance of the system which is presented in this thesis can be further improved by improving the feature selection process.

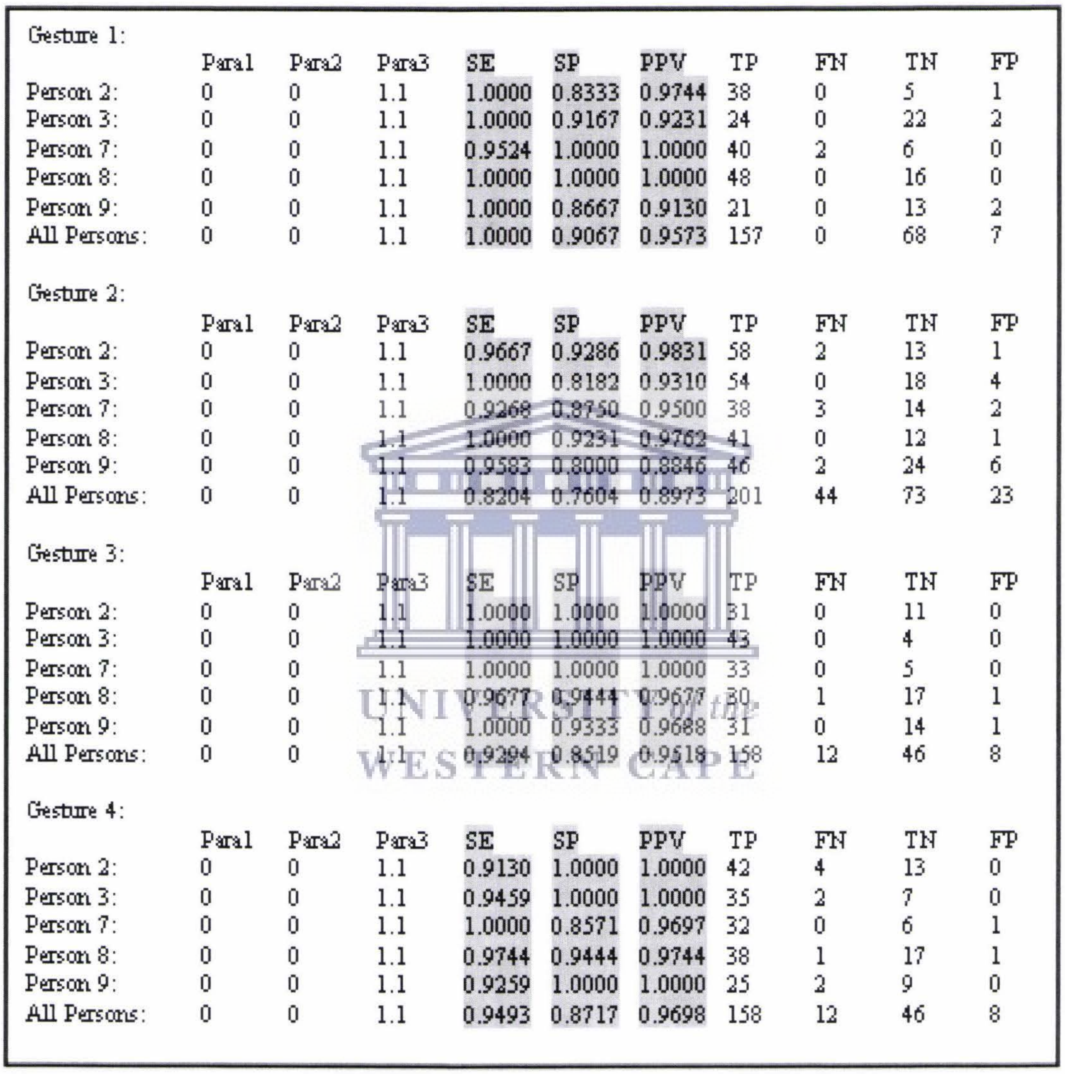
People do not always perform the same gesture in the same manner, hence the reason why person-independent recognition rates are a bit lower than the person-dependent rates. Lower person-independent hand-shape recognition results make it very difficult

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

to build person-independent gesture recognition systems. In most situations the same person often execute the same gesture in a slightly different manner, though the amount of gesture variability is often quite low.

Table 5.9: Person dependant and person independent SVM based hand-shape recognition rates



Gesture 1:

	Para1	Para2	Para3	SE	SP	PPV	TP	FN	TN	FP
Person 2:	0	0	1.1	1.0000	0.8333	0.9744	38	0	5	1
Person 3:	0	0	1.1	1.0000	0.9167	0.9231	24	0	22	2
Person 7:	0	0	1.1	0.9524	1.0000	1.0000	40	2	6	0
Person 8:	0	0	1.1	1.0000	1.0000	1.0000	48	0	16	0
Person 9:	0	0	1.1	1.0000	0.8667	0.9130	21	0	13	2
All Persons:	0	0	1.1	1.0000	0.9067	0.9573	157	0	68	7

Gesture 2:

	Para1	Para2	Para3	SE	SP	PPV	TP	FN	TN	FP
Person 2:	0	0	1.1	0.9667	0.9286	0.9831	58	2	13	1
Person 3:	0	0	1.1	1.0000	0.8182	0.9310	54	0	18	4
Person 7:	0	0	1.1	0.9268	0.8750	0.9500	38	3	14	2
Person 8:	0	0	1.1	1.0000	0.9231	0.9762	41	0	12	1
Person 9:	0	0	1.1	0.9583	0.8000	0.8846	46	2	24	6
All Persons:	0	0	1.1	0.8204	0.7604	0.8973	201	44	73	23

Gesture 3:

	Para1	Para2	Para3	SE	SP	PPV	TP	FN	TN	FP
Person 2:	0	0	1.1	1.0000	1.0000	1.0000	31	0	11	0
Person 3:	0	0	1.1	1.0000	1.0000	1.0000	43	0	4	0
Person 7:	0	0	1.1	1.0000	1.0000	1.0000	33	0	5	0
Person 8:	0	0	1.1	0.9677	0.9444	0.9677	30	1	17	1
Person 9:	0	0	1.1	1.0000	0.9333	0.9688	31	0	14	1
All Persons:	0	0	1.1	0.9294	0.8519	0.9518	158	12	46	8

Gesture 4:

	Para1	Para2	Para3	SE	SP	PPV	TP	FN	TN	FP
Person 2:	0	0	1.1	0.9130	1.0000	1.0000	42	4	13	0
Person 3:	0	0	1.1	0.9459	1.0000	1.0000	35	2	7	0
Person 7:	0	0	1.1	1.0000	0.8571	0.9697	32	0	6	1
Person 8:	0	0	1.1	0.9744	0.9444	0.9744	38	1	17	1
Person 9:	0	0	1.1	0.9259	1.0000	1.0000	25	2	9	0
All Persons:	0	0	1.1	0.9493	0.8717	0.9698	158	12	46	8

UNIVERSITY of the
WESTERN CAPE

The column values shown in Table 5.9 are explained as follows:

1) Para1: represents parameter *t* of SVM light [63]. This parameter represents the type of kernel, and in particular 0 represents a linear kernel.

https://etd.uwc.ac.za/

2) Para2: represents parameter *g or gamma* of the RBF kernel of the SVM light.
3) Para3: represents parameter *j* of SVM light that gives a measure by which training errors on positive examples outweigh negative examples.
4) SE = sensitivity
5) SP = specificity
6) PPV = positive predictive value
7) TP = true positive
8) FN = false negative
9) TN = true negative
10) FP = false positive

5.6 An assessment of the segmentation results

In this chapter a qualitative assessment of the segmentation results produced by our protype system is given. The given assessment shows that inspite of the fact that perfect segmentation results could not be achieved, the produced results adequately describe the sampled gestures. The adeaquacy of the segmentation and the tracking processes was tested by passing the aligned blob sequences into a SVM. The high recognition rates achieved by the SVM confirmed the adeaquacy of our segmentation results for gesture recognition purposes.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Chapter 6

Discussion and Conclusion

6.1 Background

In an endeavour to build a gesture recognition system that recognizes a large number of dynamic SASL gestures, we developed a prototype system that isolates the SASL cheremes. A chereme based dynamic gesture recognition system is designed based on the linguist notion of a sign language gesture. Sign language linguists argue that every sign language gesture is derived from a small set of cheremes that are peculiar to each sign language. The hand-shape, the hand movement patterns, and the hand positions are the major components of each hand gesture. However, computer vision algorithms have not yet matured enough to support comprehensive segmentation of moving hands. Besides, 2D projections are adversely affected by the occlusion problem to such an extent that 2D views hardly provide sufficient information that distinguishes the characteristics of various objects [14]. Consequently, most available vision-based gesture recognition systems often use crude estimates of hand-shape information. The use of crude prime gesture information seriously compromises the gesture recognition process. This thesis presented an improved method of extracting the hand-shape chereme. This thesis also describes how the hand position cheremes are extracted using our prototype system. Though the hand movement patterns were not automatically classified in this work, nonetheless the HMM which effectively represents the relationship between consecutive segments of an object of interest [45] can be used to achieve this. Section 6.2 compares the features of the old and the new gesture recognition systems.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

6.2 Characteristics of the new gesture recognition systems

The characteristics of the new gesture recognition system are better understood by comparing them to those of other gesture recognition systems. Table 6.1 summarizes how the new and the old systems handle the common gesture recognition problems.

Table 6.1: Comparing the new system with other vision-based gesture recognition systems

		New System	Other Systems
System Functions	Segmentation and component Labeling	Highly precise, efficient and cheap to implement	Efficient but computationally expensive
	Hand-shape recognition	Precise location (SVM based)	Coarse (Geometric model fitting)
	Blob Matching	Efficient	Efficient
	Gesture Recognition	Partially addressed. Gestures are identified from their basic components	Addressed. Each gesture is identified as a whole
	Object Occlusion	Not solved in 2D views	Not solved in 2D views
	Gesture Recognition	Both spatial and temporal gesture information is exploited	Mostly exploit temporal variations
	Real-time Processing	Not addressed	Addressed in some systems
	Overlap problem	Partially addressed	Depends on used approach

UNIVERSITY of the
WESTERN CAPE

6.3 Accomplished tasks

We manually isolated a number of hand-shape cheremes and some of these cheremes are given in Table 2.1. Manual identification of hand-shapes is based on a 3D world view of a gesturing hand. In SASL fingertip information is used to identify the different hand configurations. Each finger of a gesturing hand is either stretched, partially bent, totally clenched. Although it is easy to manually identify hand-shape cheremes, it is difficult to establish one to one mappings between 3D views and the 2D projections. Besides, 2D hand projections often fail to unambiguously identify the different object views. For instance the difference between cheremes H2 and H3, shown in Table 2.1, is

https://etd.uwc.ac.za/

not so apparent to the human eye. Since the perceptual ability of the human eye is much higher than that of machines [93], it is obviously more difficult for a computer to distinguish between these two cheremes. The ambiguities associated with 2D views hinder the successful implementation of a chereme based gesture recognition system since hand-shapes can only be classified after observing and comparing long sequences of consecutive 2D hand views. These are some of the challenges that computer vision algorithms must address, and hence the inherent limitations of most vision based gesture recognition systems. However, the hand segmentation algorithm presented in this study helps to improve the segmentation and tracking of candidate hand regions.

The following list summarizes some of the good features of the presented hand segmentation algorithm.

- *Robust method of combining and extracting low-level image cues*

  Skin–colour, motion and edge information are combined in such a way that the weak edge-based boundaries and the moving skin-coloured pixels are preserved. Such a design helps to minimize loss of important hand-shape information.

- *Enhanced boundary tracing algorithm*

  Unlike in other boundary tracing algorithms [71][21][1], the algorithm presented in this research does not use the start pixel to determine its exit point. Irrespective of whether image boundaries were corrupted by image noise and/or other processing modules, the new boundary tracing algorithm always successfully traces all external boundary pixels of a region. Besides, it requires far much less computations than most of its predecessors. For instance, the new algorithm traces the boundaries of most blurred surfaces, although in some cases minor hitches are encountered.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

- *Robust blob matching and tracking methods*

  The blob matching and tracking modules always aligns the best matching blobs. All blob sequences that do not match any other blob are discarded after particular frame intervals.

In general, this study managed to successfully segment the head and the hand regions. The location of the head and/or the neck region is used for building a relative body coordinate system from which the positions of each image blob that is obtained from each input image is approximated. Besides the centre of mass, other features such as: standard deviation; covariance matrices; the best fit ellipse; rectangularity; and some other geometrical information, are used to describe different image blobs. On average, the performance of the new hand segmentation and tracking algorithms is remarkably good. This argument is supported by the high recognition and tracking rates that were achieved by the algorithm (see Section 5.5.2).

6.4 Limitations of the study

UNIVERSITY of the
WESTERN CAPE

One of the major limitations of the new algorithm includes its inability to detect or to trace occluded regions. Since each image is taken from a single viewpoint, it is difficult to capture all the information that fully describes a three-dimensional object. The other problem associated with the new algorithm is its inability to isolate overlapping skin-coloured regions. These problems make it very difficult for the algorithm to effectively identify the candidate hand regions especially in situations where the hand overlaps the face and/or the neck regions. However, this problem can be solved by designing a boundary tracing algorithm that takes into account the gradual decrease (increase) in the magnitude of consecutive edge-based pixels when determining the most probable tracing direction. The envisaged boundary tracing algorithm must also take into account the tracing direction assumed by the last traced boundary pixels. This proposed boundary tracing algorithm would be based on a deformed model approach (see

https://etd.uwc.ac.za/

Section 3.2.2.2). In the algorithm, gradually increasing (decreasing) values for edge-based pixels would be used to represent the internal forces, $\xi_{\text{int}}(\phi(\tau))$; and the tracing direction assumed by last tracked boundary pixels would be used to enforce the external constraints, $\xi_{ext}(\phi(\tau))$. The proposed boundary tracing algorithm would identify the most probable boundaries of each output image irrespective of whether there is an overlap or not in the region of interest.

Another limitation of the presented hand segmentation algorithm lies in its dependence on the successful location of the head region. If the background is cluttered, there is no guarantee that this algorithm would always locate a person's head region. Similarly, if an image includes many individual views of different people, there is no guarantee that each isolated head region would be correctly associated with other blobs that describe the same object.

The SVM is an excellent tool for classifying objects. Although this research implements the SVM code that compares an unclassified dataset with only one of the known class at a time, other researchers implemented multi-class-based classification methods. For instance, in [14] the unclassified dataset is sequentially tested against all known classes. The unclassified dataset is given the same label as that of a class that best matches it. The multi-class approach is very important for the purposes of this research since it automatically identifies the most probable hand-shape chereme for any given sequence of candidate hand regions.

UNIVERSITY of the
WESTERN CAPE

6.5 Future activities

One of the objectives of this research was to identify the different cheremes. Although the hand-shape and position cheremes were successfully identified, the hand movement chereme was not explored. The hand movement patterns can be identified by implementing a HMM based learning and recognition module. In fact, most researchers

https://etd.uwc.ac.za/

have used the HMM to predict gesture patterns [19][32]. In essence all we need is to verify whether the features extracted in this research can sufficiently predict the hand movement patterns. Unlike the features used by other researchers [28][32], the feature set used in this research is more precise and hence more robust results were expected.

Although it was demonstrated that dynamic SASL gesture cheremes can be automatically isolated from video images of gesturing people, there is a need to ensure that the process of extracting cheremes would run in real-time. This can be achieved by either implementing hardware-based processing modules or by improving the efficiency of the image processing algorithm. A meaningful gesture dictionary can be written once all cheremes that constitute each gesture are extracted, and once the combination sequences of those cheremes have been identified. It is recommended that any extensions to this work should also further verify the applicability of the isolated cheremes to the automatic gesture recognition problem.

6.6 Concluding remarks

Adequately segmented hand regions provide sufficient information that enables extraction of some basic components of a gesture. However, computer vision has not yet matured enough to guarantee this. Despite this glaring setback, this research demonstrated that important gesture information can be extracted from hand blobs once the segmentation algorithm is designed to preserve the object boundaries. However, more work still need to be done in order to overcome the occlusion and overlap problems. Real-time chereme detection is vital for any chereme-based gesture recognition, and hence this is one of the challenges that need to be addressed in future.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

References

1. William K. Pratt;"Digital Image Processing"; Fourth Edition, Wiley Inter-science Publication; pp 579-622, ISBN: 978-0-471-76777-0; USA; 2007

2. Matthew Turk; "The Virtual Environment Handbook: Chapter 9"; Editor: Kay M. Stanney; Publisher: Lawrence Erlbaum Associates, Inc. [online] available: http://vehand.engr.ucf.edu/handbook/chapters/chapter9.pdf; [downloaded: 26 October 2007]; University of Central Florida

3. Cambridge University Engineering Department; "The Fundamentals of HTK"; COPYRIGHT 1995-1999 Microsoft Corporation. COPYRIGHT 2001-2002

4. Kjeldsen F. C. M.; "Visual Interpretation of Hand Gestures as a Practical Interface Modality"; PhD Thesis; Columbia University; 1997

5. Sagawa H., Takeuchi M. and Ohki M.; "Description and recognition methods for sign Language based on gesture components"; International conference on Intelligent user interfaces, pp 97-104; ACM 1997

6. Lee J. and Kunii T. L.; "Model-Based Analysis of Hand Posture"; IEEE Computer Graphics and Applications, Volume 15, No: 5; pp 77-86, Sept 1995

7. Jonathan G. Pearl; "Is Sign a Language?"; [online] available: http://www.cord.edu/faculty/sprunger/e315/SIGNPAPER.htm: [downloaded: 11October 2007]; Rice University; Fall Semester 1994

8. Sagawa H. and Takeuchi M.; "Learning Methods of Gesture Templates for Sign Language Recognition Based on Gesture Components"; Computers and their Applications, pp 344-347; 1998

UNIVERSITY of the WESTERN CAPE

https://etd.uwc.ac.za/

9. Hualu Wang, Shih-Fu Chang; "A highly Efficient System for Automatic Face Region Detection in MPEG Video"; IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 4, pp 615-628; 1997

10. Christopher Lee and Yangsheng Xu; "Online, Interactive Learning of Gestures for Human/Robot Interfaces"; IEEE International Conference on Robotics and Automation, Vol. 4, pp. 2982-2987, 1996

11. Wang Xu, B.A.; "A Comparison of Chinese and Taiwan Sign Languages: Towards a New Model for Sign Language Comparison"; Ma thesis, Graduate School of The Ohio State University; 2006

12. Lenman S, Bretzner L and Eiderback B.; "Computer Vision Based Recognition of Hand Gestures for Human Computer Interaction"; Technical report TRITANA - D0209, CID-report, Royal institute of Technology; Stockholm; Sweden, 2002

13. Sylvie C.W. Ong, Surendra 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; 2005

14. Pittore M., Verri A. and Campani M.; "Learning to Recognize Visual Dynamic Events from Examples"; International Journal of Computer Vision; Volume 38, No: 1; pp 35-44; 2000

15. Pavlovic V. I., Sharma R. and Huang T. S.; "Visual Interpretation of Hand Gestures for Human-Computer- Interaction: A Review"; IEEE Transactions on Pattern Analysis And Machine Intelligence Vol. 19, No 7; pp 677-694, July 1997;

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

16. William Stokoe; "The Study and Use of Sign Language"; Sign Language Studies, Vol. 1.4, pp 369-406; E-ISSN: 1533-6263, Print ISSN: 0302-1475, Gallaudet University Press; 2000/1

17. Merriam-Webster online dictionary; [online] available: http://www.merriam-webster.com/dictionary/; [downloaded: 21 October 2007]

18. A.L. Sexton; "Grammaticalization in American sign language"; Elsevier Journal of Language Sciences; Vol.21, pp 105-141, 1999

19. Thad Eugene Starner; "Visual Recognition of American Sign Language using HMM"; M.A. Thesis; Massachusetts Institute of Technology; Cambridge MA; February 1995

20. Cohen C. J.; "Dynamic System Representation and Recognition of Basic Oscillatory Motion Gestures, and Application for the Controlled of Accentuated Mechanism"; PhD Thesis; The University of Michigan; 1996

21. Danielsson P E; "An Improved Segmentation and Coding Algorithm for Binary and Nonbinary Images"; Image processing and Pattern Recognition, IBM Journal of Research and Development, Volume 26, No. 6, pp 698, 1982

22. Birchfield S; "Elliptical Head Tracking Using Intensity Gradients and Colour Histograms"; IEEE conference on Computer Vision and Pattern Recognition; pp 232, 1998

23. Holger Fillbrandt, Suat Akyol, Karl-Friedrich Kraiss; "Extraction of 3D hand shape and hand posture from image sequences for Sign Language Recognition"; Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures; pp 181; 2003

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

24. Nianjun Liu, Brian C. Lovell; "Hand Gesture Extraction by Active Shape Models"; Proceedings of the Digital Imaging Computing: Techniques and Applications (DICTA); pp 59-64; Digital Object Identifier: 10.1109/DICTA.2005.1578108; 2005

25. Richard Watson; "A Survey of Gesture Recognition Techniques: A technical report on TCD-CS-93-11"; Department of Computer Science; Trinity College; Dublin 2; UK, 1993

26. Suat Akyol and Pablo Alvarado; "Finding Relevant Image Content for mobile Sign Language Recognition"; Proceedings of IASTED International Conference on Signal Processing, Pattern Recognition and Application, pp. 48-52, 2001.

27. Bauer B. and Kraiss K-F.; "Towards a 3rd Generation Mobile Telecommunication for Deaf People"; Proceedings of 10th Symposium on Signal Theory Algorithms and Software for Mobile Communications, pp. 101-106, Germany, 2001

28. Gerhard Regoll, Andreas Kosmala and Stefan Eickeler; "High Performance Gesture Recognition Using HMM"; Springer-Verlag; Vol. 1371; Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction; pp 69-80; 1997

29. Benjamin D. Zarit, Boaz J. Super, Francis K. H. Quek; "Comparison of five Colour Models in Skin Pixel Classification"; Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems; pp 58; ISBN:0-7695-0378-0; 1999

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

30. Lifang Gu, Don Bone; "Skin Colour Region Detection in MPEG Video Sequences"; Proceedings of the 10th International Conference on Image Analysis and Processing; pp 898; ISBN:0-7695-0040-4; 1999

31. Stefan Eickeler, Andreas Kosmala and Gerhard Rigoll; "Hidden Markov Model Based Continuous Online Gesture Recognition"; Proceedings of the 14th International Conference on Pattern Recognition; Volume 2; pp 1206; 1998

32. Yanghee Nam and KwangYun Wohn; "Recognition of Space-Time Hand Gestures using Hidden Markov Models"; In Proceedings of the ACM Symposium on Virtual Reality Software and Technology, ACM Press, pp 51-58, 1996

33. Jorg Zieren, Nil Unger and Shat Akyol; "Hands Tracking from a Frontal View for Vision-Based Gesture Recognition"; Springer-Verlag; Vol. 2449**;** Proceedings of the 24th DAGM Symposium on Pattern Recognition; pp 531-539; 2002

34. Andrew D. Wilson and Aaron F. Bobick; "Real-time Online Adaptive Gesture Recognition"; Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems; pp 111-116; Greece, 1999

35. Curatelli Oscar Mayora Ibarra and Francesco Curatelli; "A brief introduction to speech analysis and recognition: *An Internet Tutorial"*; [online] Available: http://www.mor.itesm.mx/omayora/Tutorial/tutorial.html, [downloaded on: 2 Feb 2004]; Universita di ingegneria Biofisica ed electronica, 2000-05-05

36. Sturman D. J, Zeltzer D.; "Whole-hand Input"; PhD Thesis; Massachusetts Institute of Technology (1992)

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

37. H. Hienz, K. Grobel; "Automatic Estimation of Body Regions from Video Images"; I. Wachsmuth and M. Fröhlich (Eds.) Gesture and Sign Language in Human-Computer Interaction; Springer-Verlag, pp 135-145; 1998

38. Stan Birchfield; "KLT: An Implementation of the Kanade-Lucas-Tomasi feature tracker"; [online] Available: http://www.ces.clemson.edu/~stb/klt/; [downloaded on: 10 Dec 2007]; Clemson University; Department of Electrical and Computer Engineering; 2003

39. Adam C. Schembri; "Issues in the analysis of polycomponential verbs in Australian Sign Language (Auslan)"; Doctor of Philosophy in Linguistics (A Dissertation); University of Sydney; Australia; 2001

40. Rabiner L. R.; "A Tutorial on Markov Models and Selected Application in Speech Recognition"; Proceedings of the IEEE, Vol. 77, No. 2, pp 257-286; February 1989

41. Jie Yang, Alex Waibel; "A Real-time Face Tracker"; Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision; pp 142-147, 1995

42. Richard J. Qian, M Ibrahim Sezan, Kistine E. Matthews; "A Robust Real-Time Face Tracking Algorithm"; Proceedings of International Conference on Image Processing; (ICIP); Volume: 1; pp 131-135; 1998

43. Hirohiko Sagawa, Masaru Takeuchi; "Development of an Information Kiosk with a Sign Language Recognition System"; ACM Conference on Universal Usability; pp 149-150; ISBN:1-58113-314-6; 2000

44. Peng-Jui Ku, Lu-Yun Chen, Jie Zou; "Features Tracking and Shape & Structure from Motion"; ECSE 6650-Computer Vision, [online] Available:

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

http://www.ecse.rpi.edu/Homepages/gji/CV/Projects/tracking.pdf, [downloaded on: 2 Dec 2007]; Rensselaer Polytechnic Institute; Dec 2001

45. Yang He, Amlan Kundu; "2-D Shape Classification Using Hidden Markov Model"; IEEE Transactions on Pattern Analysis and Machine Intelligence; Vol. 13, No: 11; pp 1172-1184; 1991

46. Jianbo Shi, Carlos Tomasi; "Good Features to Track"; IEEE Conference on Computer Vision and Pattern Recognition; pp 593-600; 1998

47. Chai D, Ngan K N; "Locating Facial Region of a Head-and-Shoulder Color Image"; Conference of face and gesture recognition, pp 124-129, Nara, Japan; 1998

48. W. T. Freeman and C. D. Weissman; "Television control by hand gesture"; Proceedings of Workshop on Automatic Face- and Gesture-Recognition, pp. 179-183, June 1995.

49. Zhu Y, Ren H, Xu G, Lin X; "Toward Real-time Human-Computer Interaction with Continuous Dynamic Hand Gestures"; pp 544 – 549; 2000

50. Yamaoka K, Morimoto T, Adachi H, Koide T, Mattausch H J; "Image Segmentation and Pattern Matching Based on FPGA/ASIC Implementation Architecture of Real_Time Object Tracking", Proceeding of the 2006 conference on Asia South Pacific design automation ; pp 176-181; 2006

51. Alper Yilmaz, Omar Javed, Mubarak Shah; "Object Tracking: A Survey"; ACM Computing Surveys (CSUR); Volume 38, Issue 4, Article No. 13; ISSN:0360-0300; 2006

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

52. Awad G, Han Junwei, Sutherland A; “A Unified System for Segmentation and Tracking of Face and Hands in Sign”; 18th International Conference on Pattern Recognition (ICPR'06) pp. 239-242, 2006

53. Fabio Buttussi, Luca Chittaro, Marco Coppo; “Using Web3D Technologies for Visualization and Search of Signs in an International Sign Language Dictionary”; Proceedings of the twelfth international conference on 3D web technology; pp 61-70; 2007

54. Jonathan Alon, Vassilis Athitsos, Quan Yuan, Stan Sclaroff; "Simultaneous Localization and Recognition of Dynamic Hand Gestures"; IEEE Workshop on Motion and Video Computing (WACV/MOTION'05) - Volume 2, pp. 254-260, 2005

55. Ballard D H, Brown C M; “Computer Vision”; Prentice-Halldaidb, 1982

56. Russ J C; “The Image Processing Hand Book”; CRC Press; 1995

57. Argyros A A., Lourakis M I A.; “Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera”; The 8th European Conference on Computer Vision - ECCV, Springer-Verlag, vol. 3, pp 368-379, Prague, Czech Republic, 2004

58. Addmore Machanja and Vladimir B. Bajic; “Tracking Hands and Head Regions for Sign Language Recognition Purposes: With Applications to Telephone System for the Deaf”; Southern African Telecommunication Networks Applications Conference (SATNAC), ISBN: 0-620-37043-2, Cape Town, SA, 2006

59. Addmore Machanja, Vladimir B. Bajic; “Probabilistic Tracking of Moving Hands in Video Sequences with Applications to Dynamic Gesture Recognition on Mobile

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Handsets"; Southern African Telecommunication Networks Applications Conference (SATNAC), Sugar Beach, Mauritius, ISBN: 978-0-620 39351-5; 2007

60. T. Starner, J. Weaver, and A. Pentland; "Real-time American Sign Language recognition from video using hmms"; IEEE Trans. on Pattern Anal. and Machine Intell., vol. 12, no. 8, pp. 1371-1375, December 1998

61. John C. Davis; "Statistics and Data Analysis in Geology", Third Edition, ISBN 0471172758, Imprint c2002, New York

62. John A. Rice; "Mathematical Statistics and Data analysis"; Published by Wadsworth and Brooks/Cole Advanced Books and Software; ISBN 0534082475; 1988; Belmont; California

63. Thorsten Joachims; "Transductive Inference for Text Classification using Support Vector Machines"; Proceeding of International Conference on Machine Learning (ICML), pp 200-209; 1999

64. Jianbo Shi and Jitendra Malik; "Normalized Cuts and Image Segmentation"; IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 22, NO. 8, pp 888-905, AUGUST 2000

65. M. Wertheimer; "Laws of Organization in Perceptual Forms (partial translation)"; A Sourcebook of Gestalt Psychology, W.B. Ellis, ed., pp. 71-88, Harcourt, Brace, 1938.

66. Natan Peterfreund; "Robust Tracking of Position and Velocity with Kalman Snakes"; IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 21, NO. 6, pp 564-569, JUNE 1999

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

67. Paul Kuo, John Hannah; "Improved Chin Fitting Algorithm Based on an Adaptive Snake"; International Conference on Image Processing (ICIP), pp 205-208, Atlanta, USA; October 2006

68. Park J M, Looney C G, Chen H C; "Fast Connected Component Labeling Algorithm using a Divide and Conquer Technique"; Technical Report 2000, [Online] Available : http://cs.ua.edu/research/TechnicalReports/TR-2000-04.pdf, [downloaded: 11 March 2007]

69. Chang F, Chen C J, Lu C J; "A Linear Time Component-Labeling Algorithm using Contour Tracing Technique"; Computer Vision and Image Understanding, Volume 93, Issue 2, Elsevier, pp 741, Feb 2004

70. Rosenfield A, Pfaltz J L; "Sequential Operations in Digital Picture Processing"; Journal of the ACM, Volume 13 , Issue 4; pp 471-494,October 1966

71. Yi L, Tie-Qi C, Jie C, Anthony T, Jiaxin C; "An Advanced Machine Vision System for VFD Inspection"; 6th International Conference on Manufacturing; 6-8 Sept 2000

72. Sonka M, Hlavac V, Boyle R; "Image Processing, Analysis and Machine Vision"; Brooks and Cole Publishing; 1998

73. Huang S, Zhou M, Geng G; "An Improved Geometric Deformable Model for Color Image Segmentation"; Proceeding of the 16th International Conference on Artificial Reality and Telexistence; pp 513-517; 2006; China

74. El-Baz A, Shi H, Farag A A, El- Ghar M A., Eldiasty T, Gheneim M A; "2D and 3D shape based segmentation using deformable Model"; International Conference

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

on Medical Image Computing and Computer-Assisted Intervention; Published by Springer-Verlag, Berlin Helderburg, pp 821-829, 2005

75. Cullen Jennings; "Robust Finger Tracking with Multiple Cameras"; International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems; pp 152-160, Corfu, Greece, 1999

76. Gonzalez R. C. and Woods R. E.; "Digital Image Processing"; Prentice-Hall, International Edition, USA, 2002

77. Xu C, Pham D L, Prince J L; "Medical Image Segmentation Using Deformable Models, Handbook of Medical Imaging"; Volume 2: Medical Image Processing and Analysis, edited by J.M. Fitzpatrick and M. Sonka, SPIE Press, pp. 129-174, 2000

78. Huang X, Qian Z, Huang R, Metaxas D; "Deformable-model based Textured Object Segmentation"; Huang X, Qian Z, Huang R, Metaxas D; Deformable-model based Textured Object Segmentation;, In Proceedings Of EMMCVPR Workshop, pp 119-135, 2005 In Proceedings Of EMMCVPR Workshop, pp 119-135, 2005

79. Tang J, Kawato S, Ohya J, Nakatsu R; "Locating Human Face in a complex Background Including Non-face Skin Colors"; Journal of Electronic Imaging, Vol 12, Issue 3; Co published by IS&T--The Society for Imaging Science & Technology and SPIE--The International Society for Optical Engineering; USA; pp 423-430, 2003

80. Ribeiro H L, Gonzaga A; "Hand Image Segmentation in Video Sequence by GMM: a comparative analysis"; XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'06), pp. 357-364, 2006.

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

81. V Vezhnevets, V Sazonov, A Andreeva; "A Survey on Pixel-Based Skin Color Detection Techniques"; Proceedings of Graphicon 2003; pp 85-92, Moscow, Russia

82. Scott E. Umbaugh; "Computer Imaging: Digital Image Analysis and Processing"; ISBN: 0849329191, CRC Press, 2005

83. E. Davies; "Machine Vision: Theory, Algorithms and Practicalities"; Academic Press, 1990

84. Ribeiro H L, Gonzaga A; "Hand Image Segmentation in Video Sequence by GMM: a comparative analysis"; XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'06), pp. 357-364, 2006.

85. Russel Ahmed Apu and Marina L. Gavrilova; "Flexible Tracking of Object Contours Using LR-Traversing Algorithm"; International Conference on Computer Graphics, Imaging and Visualization, pp 503-513; 2006

86. Gerald Friedland, Kristian Jantz, Tobias Lenz, Fabian Wiesel, and Ra´ul Rojas; "A Practical Approach to Boundary Accurate Multi-Object Extraction from Still Images and Videos"; Eighth IEEE International Symposium on Multimedia, pp 307-316; 2006

87. Hamada Y, Shimada N, Shirai Y; "Hand Shape Estimation under Complex Backgrounds for Sign Language Recognition"; Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp 589 – 594, 2004

88. Tae-Yong Kim, Jihun Park and Seong-Whan Lee; "Object Boundary Edge Selection Using Normal Direction Derivatives of a Contour in a Complex Scene";

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

17th International Conference on Pattern Recognition, Vol. 4, pp 755-758,2004

89. Francois Destrempes, Max Mignotte; "A Statistical Model for Contours in Images"; IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 26, No. 5, pp 626-638, 2004

90. Sugiyana T, Kwan P W H, Toraichi K, Katagishi K; "A Contour Tracing Algorithm that Avoids Duplicate Tracing Common Boundaries between Regions"; The Journal of the Institute of Image Electronics Engineers of Japan (Academic Journal), Vol. 33, no. 4-B, pp 586-596, 2004

91. Hermann Hienz, Kirsti Grobel, Georg Offner; "Real-Time Hand-Arm Motion Analysis using a single Video Camera"; 2nd International Conference on Automatic Face and Gesture Recognition (FG '96); pp. 323; 1996

92. Mathias Kolsch, Matthew Turk; "Fast 2D Hand Tracking with Flock of Features and Multi-Cue Integration"; IEEE Conference on Computer Vision and Pattern Recognition Worksop, pp 158, 2004

93. Jayaram K. Udupa, Vivki R. LaBlanc, Hilary Schidt, Celina Imielinska, Punam K. Saha, George J. Grevera, Ying Zhuge, Pat Molholt, Yinpeng Jin, Leanne M. Currie; "A Methodology for Evaluating Image Segmentation Algorithms"; Proceeding of SPIE; Volume 4684; Medical Imaging: Image Processing; pp 266-277; 2002

94. Xiaojin Zhu, Jie Yang, Alex Waibel; "Segmenting Hands of Arbitrary Colour"; Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp 446; 2000

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

95. Zeungnam Bien, Jun-Hyeong Do, Jung-Bae Kim, Hyoyoung Jang, Dae-Jin Kim; "Hand Gesture As A Means Of Human-Friendly Interface/Interaction"; Iranian Conference On Fuzzy Systems, 2004

96. Francis Quek, David McNeill, Robert Bryll, Susan Duncan, Xin-Feng Ma, Cemil Kirbas, Karl E. McCullough, and Rashid Ansari; "Multimodal human discourse: gesture and speech"; ACM Transaction Computation on Human Interaction, Vol. 9, No: 3, pp 171–193, 2002. ISSN 1073-0516.

97. Francis K. H. Quek; "Toward a vision-based hand gesture interface"; Proceedings of the conference on Virtual reality software and technology, ISBN:981-02-1867-2; pp 17-31; Singapore; 1994

98. Yuanxin Zhu; "Vision-Based Recognition of Continuos Dynamic Hand Gestures"; [online] available: http://meru.cecs.missouri.edu/workshop/zhu-1.ppt: [downloaded: 18 October 2007]; University of Missouri-Columbia.

99. Francis Quek, David McNeilly, Robert Bryll, Susan Duncan, Xin-Feng Ma, Cemil Kirbas, Karl E. McCulloughy, Rashid Ansari; "Gesture and Speech Multimodal Conversational Interaction"; Vislab Report: Vislab 01-01; [online] available: http://vislab.cs.vt.edu/Publications/2001/PDFfiles/Queetal01-old.pdf; [downloaded 23 October 2007]; Vision Interfaces and Systems Laboratory; Virginia Tech

100. Jean-Luc Nespoulous, Paul Perron, Andre Roch Lecours; "The Biological Foundations of Gestures: Motor and semiotic Aspects"; ISBN 0898596459, Lawrence Earlbaum Associates Publishers, London, 1986

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

101. Elizabeth Johnston; “Lecture 15: Language in the Deaf World”; [online] available: http://pages.slc.edu/~ebj/IM_97/Lecture15/L15.html; [downloaded 26 October 2007]; Sarah Lawrence College

102. Bruno Müller Junior, Ricardo de Oliveira Anido; “Distributed RealTime Soccer Tracking”; Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks, pp 97-103, 2004

103. Robert Bayley, Ceil Lucas, Mary Rose; “Phonological variation in American Sign Language: The case of 1 handshape”; Cambridge Journals: Language Variation and Change, Vol. 14, pp 19-35; Cambridge University Press 0954-3945/02; 2002

104. Andrew David Wilson; “Adaptive Models for the Recognition of Human Gesture”; PhD thesis, Massachusetts Institute Of Technology; 2000

105. Sanjay Kr Singh, D. S. Chauhan, Mayank Vatsa, Richa Singh; “A Robust Skin Color Based Face Detection Algorithm”; Tankang Journal of Science Engineering, Vol. 6, pp: 227- 234, 2003

106. Srikanth Rangarajan; “Tutorial on Edge Detection”; [online] available: http://www.uweb.ucsb.edu/~shahman/AfED.doc”; [downloaded 30 November 2007]; University of California, Santa Barbara

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Appendix A: Letter of Consent

Bastion School for the Deaf
Newlands
Cape Town

Request for permission to use your pictures in my thesis:

Research Title: Towards a chereme-based dynamic South African Sign Language gesture recognition system.

Researcher's Name: Addmore Machanja

My work is aimed at building a firm foundation for building a sign language recognition system that recognizes a large vocabulary of sign language gestures. Such a system would help to eliminate communication breakdowns between the speech using people and the hearing impaired people. Signing is an important modality of human to human communication. Besides, technologies used in for building dynamic gesture recognition systems can also be adapted almost all human-computer interactions related applications.

The researcher has no intentions of tarnishing or abusing the personalities of all people who participated in this research. However, when explaining the research findings, often there is need to include some pictures of the people who participated in the research. I hereby write to formally seek your permission to allow me to include your pictures in my thesis or in any other publications that explains my research findings.

It was my pleasure to get an opportunity to work with you all.

With Regards

Addmore Machanja; PhD Student
Department of Computer Science
University of the Western Cape
P. Bag X17; Bellville; 7535
Cell No: 0723267113

Signature of Coordinator: .... Date ....
Bastion School of the Deaf
Signature of Participant 1............................ Date.......................
Signature of Participant 2............................ Date.......................
Signature of Participant 3............................ Date.......................
Signature of Participant 4............................ Date.......................
Signature of Participant 5............................ Date.......................
Signature of Participant 6............................ Date.......................

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Appendix B: Some of the features from person 4 gesture 1

```
IMAGE NUMBER:  1

Blob number:  1
centre of mass   (34,67): total 112
X: Standard Dev: 9.04   Variance : 81.73
Y: Standard Dev: 19.40   Variance : 376.19
XY: Variance    : -164.04
Theta = 0.84  alpha =449.38  beta = 8.54  axis_ratio = 52.60

Blob number:  2
centre of mass   (54,73): total 129
X: Standard Dev: 2.79   Variance : 7.76
Y: Standard Dev: 3.96   Variance : 15.70
XY: Variance    : -2.80
Theta = 1.46  alpha =16.59  beta = 6.87  axis_ratio = 2.41
```

```
IMAGE NUMBER:  2
Blob number:  1
centre of mass   (56,66): total 142
X: Standard Dev: 3.02   Variance : 9.11
Y: Standard Dev: 3.93   Variance : 15.46
XY: Variance    : -2.47
Theta = 1.48  alpha =16.31  beta = 8.26  axis_ratio = 1.98

Blob number:  2
centre of mass   (38,91): total 35
X: Standard Dev: 2.66   Variance : 7.09
Y: Standard Dev: 2.18   Variance : 4.74
XY: Variance    : 4.74
Theta = 0.86  alpha =10.80  beta = 1.03  axis_ratio = 10.50
```

```
IMAGE NUMBER:  3
Blob number:  1
centre of mass   (58,62): total 151
X: Standard Dev: 3.17   Variance : 10.04
Y: Standard Dev: 3.98   Variance : 15.87
XY: Variance    : -3.15
Theta = 1.43  alpha =17.25  beta = 8.67  axis_ratio = 1.99
```

```
IMAGE NUMBER:  4
Blob number:  1
centre of mass   (99,57): total 97
X: Standard Dev: 0.99   Variance : 0.98
Y: Standard Dev: 15.89   Variance : 252.58
XY: Variance    : 1.03
Theta = 1.57  alpha =252.58  beta = 0.98  axis_ratio = 259.02


Blob number:  2
centre of mass   (59,60): total 160
X: Standard Dev: 3.33   Variance : 11.12
Y: Standard Dev: 4.10   Variance : 16.84
XY: Variance    : -3.64
Theta = 1.41  alpha =18.60  beta = 9.35  axis_ratio = 1.99

Blob number:  3
centre of mass   (32,84): total 197
X: Standard Dev: 5.41   Variance : 29.30
Y: Standard Dev: 6.70   Variance : 44.92
XY: Variance    : 9.71
Theta = 1.41  alpha =49.58  beta = 24.64  axis_ratio = 2.01
```

```
IMAGE NUMBER:  5
Blob number:  1
centre of mass   (62,58): total 161
X: Standard Dev: 3.33   Variance : 11.07
Y: Standard Dev: 4.05   Variance : 16.42
XY: Variance    : -2.77
Theta = 1.46  alpha =17.59  beta = 9.89  axis_ratio = 1.78

Blob number:  2
centre of mass   (30,83): total 76
X: Standard Dev: 3.88   Variance : 15.07
Y: Standard Dev: 3.61   Variance : 13.00
XY: Variance    : 4.15
Theta = 1.23  alpha =18.31  beta = 9.76  axis_ratio = 1.88
```

```
IMAGE NUMBER:  6
Blob number:  1
centre of mass   (63,58): total 155
X: Standard Dev: 3.25   Variance : 10.55
Y: Standard Dev: 3.89   Variance : 15.14
XY: Variance    : -1.78
Theta = 1.51  alpha =15.75  beta = 9.94  axis_ratio = 1.58
```

UNIVERSITY of the
WESTERN CAPE

```
IMAGE NUMBER:   7
Blob number:   1
centre of mass   (31,80): total 105
X: Standard Dev: 4.58   Variance : 20.98
Y: Standard Dev: 10.25   Variance : 105.15
XY: Variance    : -8.02
Theta = 1.53  alpha =105.91  beta = 20.22  axis_ratio = 5.24

Blob number:   2
centre of mass   (63,58): total 159
X: Standard Dev: 3.30   Variance : 10.91
Y: Standard Dev: 3.96   Variance : 15.65

XY: Variance    : -2.00
Theta = 1.50  alpha =16.38  beta = 10.18  axis_ratio = 1.61
```

```
IMAGE NUMBER:   8
Blob number:   1
centre of mass   (64,57): total 151
X: Standard Dev: 3.14   Variance : 9.88
Y: Standard Dev: 3.95   Variance : 15.61
XY: Variance    : -2.07
Theta = 1.50  alpha =16.28  beta = 9.21  axis_ratio = 1.77
```

```
IMAGE NUMBER:   9
Blob number:   1
centre of mass   (99,74): total 40
X: Standard Dev: 1.64   Variance : 2.70
Y: Standard Dev: 3.00   Variance : 9.02
XY: Variance    : 0.50
Theta = 1.56  alpha =9.06  beta = 2.66  axis_ratio = 3.41

Blob number:   2
centre of mass   (64,57): total 145
X: Standard Dev: 3.03   Variance : 9.16
Y: Standard Dev: 3.98   Variance : 15.86
XY: Variance    : -2.27
Theta = 1.49  alpha =16.56  beta = 8.46  axis_ratio = 1.96

Blob number:   3
centre of mass   (30,83): total 94
X: Standard Dev: 4.68   Variance : 21.91
Y: Standard Dev: 4.36   Variance : 19.04
XY: Variance    : 2.59
Theta = 1.37  alpha =23.44  beta = 17.52  axis_ratio = 1.34
```

```
IMAGE NUMBER:   10
Blob number:   1
centre of mass   (64,58): total 148
X: Standard Dev: 3.10   Variance : 9.58
Y: Standard Dev: 3.93   Variance : 15.41
XY: Variance    : -1.47
Theta = 1.53  alpha =15.76  beta = 9.23  axis_ratio = 1.71
```

```
IMAGE NUMBER:   11
Blob number:   1
centre of mass   (32,76): total 127
X: Standard Dev: 5.74   Variance : 32.98
Y: Standard Dev: 14.94   Variance : 223.16
XY: Variance    : -44.81
Theta = 1.28  alpha =233.19  beta = 22.95  axis_ratio = 10.16

Blob number:   2
centre of mass   (64,58): total 149
X: Standard Dev: 3.12   Variance : 9.75
Y: Standard Dev: 3.99   Variance : 15.96
XY: Variance    : -1.70
Theta = 1.53  alpha =16.39  beta = 9.32  axis_ratio = 1.76
```

```
IMAGE NUMBER:   12
Blob number:   1
centre of mass   (64,58): total 146
X: Standard Dev: 3.08   Variance : 9.49
Y: Standard Dev: 3.91   Variance : 15.29
XY: Variance    : -2.01
Theta = 1.50  alpha =15.92  beta = 8.86  axis_ratio = 1.80

Blob number:   2
centre of mass   (30,83): total 108
X: Standard Dev: 5.04   Variance : 25.43
Y: Standard Dev: 4.68   Variance : 21.89
XY: Variance    : 3.85
Theta = 1.34  alpha =27.89  beta = 19.43  axis_ratio = 1.44
```

```
IMAGE NUMBER:   13
Blob number:   1
centre of mass   (33,73): total 78
X: Standard Dev: 5.80   Variance : 33.69
Y: Standard Dev: 16.19   Variance : 262.00
XY: Variance    : -56.84
Theta = 1.19  alpha =275.37  beta = 20.32  axis_ratio = 13.55
```

```
IMAGE NUMBER:   14
Blob number:   1
centre of mass   (32,79): total 233
X: Standard Dev: 5.05   Variance : 25.50
Y: Standard Dev: 11.69   Variance : 136.73
XY: Variance    : -4.02
Theta = 1.57  alpha =136.87  beta = 25.36  axis_ratio = 5.40
```

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/

Blob number: 2 centre of mass (64,58): total 145 X: Standard Dev: 3.15 Variance : 9.90 Y: Standard Dev: 3.80 Variance : 14.48 XY: Variance : -0.70 Theta = 1.56 alpha =14.58 beta = 9.80 axis_ratio = 1.49 Blob number: 3 centre of mass (99,87): total 34 X: Standard Dev: 1.12 Variance : 1.26 Y: Standard Dev: 3.15 Variance : 9.94 XY: Variance : 0.74 Theta = 1.52 alpha =10.00 beta = 1.20 axis_ratio = 8.32	Blob number: 2 centre of mass (64,58): total 145 X: Standard Dev: 3.12 Variance : 9.76 Y: Standard Dev: 3.84 Variance : 14.74 XY: Variance : -1.72 Theta = 1.52 alpha =15.28 beta = 9.22 axis_ratio = 1.66
IMAGE NUMBER: 15 Blob number: 1 centre of mass (99,57): total 105 X: Standard Dev: 0.99 Variance : 0.97 Y: Standard Dev: 16.26 Variance : 264.49 XY: Variance : 2.96 Theta = 1.54 alpha =264.52 beta = 0.94 axis_ratio = 281.96 Blob number: 2 centre of mass (65,58): total 150 X: Standard Dev: 3.07 Variance : 9.40 Y: Standard Dev: 4.07 Variance : 16.55 XY: Variance : -1.97 Theta = 1.52 alpha =17.06 beta = 8.89 axis_ratio = 1.92	IMAGE NUMBER: 16 Blob number: 1 centre of mass (30,82): total 101 X: Standard Dev: 3.96 Variance : 15.69 Y: Standard Dev: 6.31 Variance : 39.78 XY: Variance : 1.11 Theta = 1.57 alpha =39.84 beta = 15.64 axis_ratio = 2.55 Blob number: 2 centre of mass (65,58): total 147 X: Standard Dev: 3.13 Variance : 9.81 Y: Standard Dev: 3.84 Variance : 14.75 XY: Variance : -1.50 Theta = 1.53 alpha =15.17 beta = 9.39 axis_ratio = 1.62
IMAGE NUMBER: 17 Blob number: 1 centre of mass (65,58): total 131 X: Standard Dev: 3.21 Variance : 10.32 Y: Standard Dev: 4.17 Variance : 17.41 XY: Variance : -2.21 Theta = 1.51 alpha =18.04 beta = 9.69 axis_ratio = 1.86	IMAGE NUMBER: 18 Blob number: 1 centre of mass (64,58): total 154 X: Standard Dev: 3.14 Variance : 9.87 Y: Standard Dev: 4.04 Variance : 16.31 XY: Variance : -2.24 Theta = 1.50 alpha =17.01 beta = 9.17 axis_ratio = 1.86
IMAGE NUMBER: 19 Blob number: 1 centre of mass (64,59): total 158 X: Standard Dev: 3.22 Variance : 10.35 Y: Standard Dev: 4.04 Variance : 16.30 XY: Variance : -1.77 Theta = 1.52 alpha =16.79 beta = 9.86 axis_ratio = 1.70	IMAGE NUMBER: 20 Blob number: 1 centre of mass (64,62): total 164 X: Standard Dev: 3.30 Variance : 10.88 Y: Standard Dev: 4.05 Variance : 16.42 XY: Variance : -2.07 Theta = 1.51 alpha =17.11 beta = 10.19 axis_ratio = 1.68
IMAGE NUMBER: 21 Blob number: 1 centre of mass (99,73): total 103 X: Standard Dev: 1.05 Variance : 1.10 Y: Standard Dev: 13.13 Variance : 172.38 XY: Variance : 1.39 Theta = 1.56 alpha =172.39 beta = 1.09 axis_ratio = 158.76 Blob number: 2 centre of mass (63,67): total 171 X: Standard Dev: 3.38 Variance : 11.42 Y: Standard Dev: 4.39 Variance : 19.29 XY: Variance : -5.33 Theta = 1.34 alpha =21.98 beta = 8.72 axis_ratio = 2.52	

UNIVERSITY of the
WESTERN CAPE

https://etd.uwc.ac.za/