UNIVERSITY OF THE WESTERN CAPE

Upper Body Pose Recognition and Estimation towards the Translation of South African Sign Language

by

Imran Achmed

A thesis submitted in fulfillment for the
degree of Master of Science

in the

Faculty of Science
Department of Computer Science

February 2011

Declaration of Authorship

I declare that *Upper Body Pose Recognition and Estimation towards the Translation of South African Sign Language* is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Signed: _____

Date: _____

UNIVERSITY OF THE WESTERN CAPE

Abstract

Faculty of Science
Department of Computer Science

Master of Science

by Imran Achmed

Recognising and estimating gestures is a fundamental aspect towards translating from a sign language to a spoken language. It is a challenging problem and at the same time, a growing phenomenon in Computer Vision. This thesis presents two approaches, an example-based and a learning-based approach, for performing integrated detection, segmentation and 3D estimation of the human upper body from a single camera view. It investigates whether an upper body pose can be estimated from a database of exemplars with labelled poses. It also investigates whether an upper body pose can be estimated using skin feature extraction, Support Vector Machines (SVM) and a 3D human body model. The example-based and learning-based approaches obtained success rates of 64% and 88%, respectively. An analysis of the two approaches have shown that, although the learning-based system generally performs better than the example-based system, both approaches are suitable to recognise and estimate upper body poses in a South African sign language recognition and translation system.

Acknowledgements

I began my first year at university through the grace of God and only through His grace have I reached thus far. He has granted me the knowledge, wisdom and opportunity to reach a level of success that I am truly proud of.

My sincerest gratitude and appreciation is given to my supervisor Mr. James Connan. His continuous inspiration, guidance and encouragement has instilled in me the motivation to succeed. His patience, perseverance and the valuable time of his life he spent guiding me in my research is invaluable to me and for this I thank him.

During my research, Jacob Richard Whitehill has offered his time to teach me many things in this research. His sound advice and constructive comments, from the beginning of my research and until then end, was of great benefit. An enormous thanks goes to him for his generous assistance.

I would like to thank Prof. Richard Madsen and Abduraghiem Latief for their kind and insightful guidance on my statistical analysis. Thank you to Mehrdad and Jameel for their assistance and guidance. I would also like to extend my thanks to the individuals who offered their time to participate in this research.

A word of thanks to Telkom for sponsoring the entire duration of my studies. Without this sponsorship, it would not have been possible to obtain tertiary education.

I am indebted to my parents, who taught me the value of hard work and who tried their best to help me in my studies in whichever way they could, even though they never had the opportunity to study towards a tertiary education. This enormous support has pushed me to excel. I could not have reached this goal without them.

To my siblings, Chiara, Fozia and Riedewaan, thank you for your support and I hope I have set the bar. To Uncle Arnold and Uncle Andrew, your concern in my success has been a motivation to me, thank you.

Last but not least, a special thank you to my girlfriend, Tamlynne Meyer. She has been a helpful hand in the editing of this thesis. Her love, patience, support and understanding throughout my studies, has given me the momentum to see through each year. She is my source of inspiration.

Contents

List of Figures

List of Tables

Abbreviations

2D	- Two Dimensional
3D	- Three Dimensional
AI	- Artificial Intelligence
ASL	- American Sign Language
BSL	- British Sign Language
CPU	- Computer Processing Unit
DOF	- Degree of Freedom
EM	- Expectation Maximisation
FPS	- Frames Per Second
GMM	- Gaussian Mixture Model
GSL	- Greek Sign Language
H-Anim	- Humanoid Animation Working Group
HSV	- Hue, Saturation, Value
ISM	- Implicit Shape Model
JSL	- Japanese Sign Language
KPCA	- Kernel Principle Component Analysis
LibSVM	- Library of Support Vector Machines
LLE	- Locally Linear Embedding
LSH	- Locality Sensitive Hashing
MCMC	- Markov Chain Monte Carlo
MIT	- Massachusetts Institute of Technology
PC	- Personal Computer
RAM	- Random Access Memory
RANSAC	- Random Sample Consensus
RBF	- Radial Basis Function
RGB	- Red, Green, Blue
ROI	- Regions of Interest
RVM	- Relevance Vector Machine
SASL	- South African Sign Language
SGPLVM	- Scaled Gaussian Process Latent Variable Model

SVM	- Support Vector Machine
TSL	- Tint, Saturation, Lightness
UWC	- University of the Western Cape
YCbCr	- Luminance and Chrominance

Chapter 1

Introduction

1.1 Background & Motivation

Communication is an essential life skill used on a daily basis. It is a skill that is used to exchange or share information and, assist relationships and interactions amongst each other. It is the foundation that not only facilitates association and independence but also promotes a cohesive environment within societies.

In societies, not everyone is able to communicate in the form of spoken languages, especially individuals that are deaf. This divide is the main cause of the communication barrier between the deaf and the hearing communities. In South Africa, there are more than one million deaf individuals, of which 300 000 are deaf in both ears [45] and 600 0000 use South African Sign Language as their primary language to communicate [33].

There are different sign languages throughout the world, each with a grammar different to others. Examples of these include, British Sign Language (BSL) in Britain, Greek Sign Language (GSL) in Greece, American Sign Language (ASL) in America, Japanese Sign Language (JSL) in Japan and South African Sign Language (SASL) in South Africa. SASL is recognised as the official language for the deaf communities under the South African constitution [33]. Despite this fact, individuals from deaf communities are faced with severely limited educational services and socio-economic opportunities relative to the hearing person. This is an influential problem in South Africa that has led to the deaf being largely marginalised in society.

It is often a misconception by the hearing communities that there is a correlation between signed and spoken languages. This is not the case; sign languages are developed independently of spoken languages from the need to communicate amongst members of the deaf communities. Sign Language is a language on its own and developed to

fully express actions, emotions and objects in human-to-human dialogue. This is the only similarity that can be made with spoken languages. Another misconception is that linguistic interpretations of spoken languages can be easily applied to sign language [152][63]. These misconceptions has led to the assumption that a large number of deaf individuals can read and write, which is not true.

To alleviate this problem, skilled interpreters have been used to facilitate communication between these two different societies [33]. Their services, however, need to be arranged ahead of time and tend to be expensive, as there are insufficient skilled interpreters to assist each deaf person [47][63][45]. Unfortunately, most deaf individuals are still faced with a communication barrier. In cases where privacy is concerned, such as medical consultations, a deaf person may not be comfortable having an interpreter present.

On the other hand, the presence of an automated machine translation system that translates from sign language to a spoken language and vice-versa, would benefit the deaf community immensely. It would, in effect, improve the communication between the hearing and deaf communities. It would also be a solution to the shortage of skilled interpreters and a solution to the privacy issues.

At the University of the Western Cape, a research group has been formed that has proposed the development of the *Integration of Sign and Verbal Communication: South African Sign Language Recognition and Animation* project, of which this research forms part. This project is aimed at developing a component of such a machine translation system.

In order to understand sign language, body movements in the form of manual and non-manual gestures must be interpreted. Manual gestures consist of arm movements and locations, hand movements and locations, and hand shapes. Non-manual gestures consist of facial expressions such as a smile or frown, each conveying a very different meaning. To linguistically determine the meaning and translate to a spoken language, these gestures need to be analysed.

In the SASL project, a number of subsystems have been individually implemented in the form of sign language recognition [166][135][110][123] and rendering systems [163][44]. van Wyk [163] developed a full-body 3D human body model that renders sign language on a PC in an animated form. This research was followed by Ghaziasgar [44] that studied the feasibility of rendering sign language on a mobile phone. Whitehill [166] researched non-manual gestures and developed a robust automatic facial expression recognition and classification system. Naidoo [110] and Rajah [123] have both developed gesture recognition systems that track hand movements. Segers [135] developed a hand-shape gesture recognition system, however, this system only recognises hand-shapes when the

hands are placed directly in front of the camera, in more than 75% of its field-of-view. In the SASL project, a system that is able to locate the position of the hands from a distance, for further hand-shape recognition is still required. The importance of this requirement is noted by Schneider [134],

> "The palms of the hands should be held at a distance making it easy for the viewer to read both the fingers and lips simultaneously"

In addition to this, another area that is still required is the position of the arms.

1.2 Research Problem

This study is therefore aimed at developing a system that recognises the posture of an individual, in order to provide an estimate of the location of the different body parts used to perform sign language effectively. A goal of the SASL project is to incorporate a natural feel to the system by avoiding additional cumbersome equipment such as coloured markers, data gloves and data suits. This goal is conceivable with the use of computer vision techniques that employs image processing algorithms and does not require the users to be burdened by additional equipment.

The use of computer vision, however, draws attention to a fundamental factor to consider when developing such a system; it is important that the position of the hands and other body parts are estimated as accurately as possible.

In addition to the natural feel of such a system, another goal of the project is to allow the users to be free in an uncontrolled environment. Image processing in an uncontrolled environment is challenging and a challenge worth investigating.

The key aim of the SASL project is to have the entire system as a service on a mobile phone. In the study by Ghaziasgar [44], it is shown that mobile devices are well suited for capturing audio and video. Capturing video from a mobile device by holding it free is however unstable. This is an important element to consider when attempting to extract features consistently from visual input and another challenge that should be investigated.

1.3 Research Questions

Considering the challenges that are being faced, the research questions can be formulated as follows:

1. Can an upper body pose be estimated by matching an input image to a database of exemplars with labelled poses rendered by a 3D human body model in Blender?
2. Alternatively, can an upper body pose be estimated by learning a regression function from an input image onto the desired positions using skin feature extraction, Support Vector Machines (SVM) and a 3D human body model?
3. How do these approaches compare in accuracy?
4. Are these approaches suitable for a sign language recognition and translation system?

1.4 Research Objectives

These questions can be hypothesised and addressed with a number of techniques for each individual approach.

The first approach is to follow a template-based matching technique using the widely-used Chamfer Distance Transformation. To render such an approach feasible, the pre-processing techniques should involve a background subtraction method to isolate the individual in the scene. It should also generate a silhouette of the human body and use a distance approximation to find the best match from the database of exemplars.

The second approach is a novel proposition that applies feature extraction and artificial intelligence to the field of pose estimation. This approach requires a reliable means to extract the necessary features while discarding the unnecessary ones in an attempt to provide robust feature extraction. These features should be used along with artificial intelligence, such as SVMs, to predict an upper body pose.

An important fact to note is that humans are capable of instantly recognising a pose but are incapable of accurately measuring a pose by mere observation. Therefore, in both approaches, extensive use of a 3D human body model is used to estimate the main joint positions in the upper body, once the pose has been recognised.

1.5 Premises

In sign language, the entire body is not used. When communicating in sign language, only the upper body is used [39][133][8][33]. For this reason, focus is only placed on recognising and estimating the upper body limbs.

1.6 Thesis Outline

The remainder of the thesis is outlined as follows:

Chapter 2: *Pose Estimation*: This chapter describes the process of pose estimation and reviews existing literature in this field. It explores algorithms on various approaches that have significantly impacted the development of pose estimation systems and investigate studies where techniques have been developed to combine these approaches. These studies are compared and, the benefits and trade-offs thereof are highlighted.

Chapter 3: *Example-Based System*: This chapter describes and motivates the components that form part of this system. Image registration techniques required by the system, to prepare the images for feature extraction, are explained. These techniques include face detection and greyscaling, each of which are further explored to determine the most suitable algorithms available for these techniques. Feature extraction methods that include an edge detector and a distance approximation technique are also investigated and compared with closely related techniques. Finally, an overview of the 3D human body model used in both systems are presented.

Chapter 4: *Learning-Based System*: In this chapter the components that form part of the learning-based system are described. The chapter begins with an evaluation of the different colour spaces and introduces an innovative process that identifies skin-coloured pixels in an image. A comparison between simple background subtraction and probabilistic modelling techniques are discussed. The use of morphological operations is also described. In the final section, SVMs, the kernel functions and multi-class SVM techniques are explained.

Chapter 5: *Systems Implementation and Design*: This chapter provides details of the systematic implementation and design of both the example-based and learning-based system. It also provides a structural overview of each system and explains the procedure in which the algorithms in the previous chapters are combined to form the respective systems.

Chapter 6: *Experimental Results and Analysis*: In this chapter, the experiments performed on both the example-based and learning-based system are discussed. Experiments to identify a suitable kernel are also explained with the results thereof. The results obtained from each system is assessed and compared to related work. Finally, the outcome of the comparison between the two systems are discussed.

Chapter 7: *Conclusion and Future Work*: This chapter concludes the thesis and provides concluding remarks towards this research. It highlights the main contributions of the research and recommends directions for future work.

1.7 Summary

This chapter has been introduced with a background on sign language and the communication difficulties between the deaf and hearing communities. It has also described the SASL project and motivation of this study. The research problems, questions and goals were discussed as well as an outline of the rest of the thesis.

Chapter 2

Pose Recognition and Estimation

Traditionally, most pose recognition and estimation systems depend upon markers that are pre-attached to the person's body. These systems have significant disadvantages as they are conspicuous, expensive and impractical for a Sign Language application. It would be beneficial to provide an alternative solution that is marker-less. Systems that rely heavily on methods using image processing, computer vision and machine learning have become popular in the last decade and provides marker-less solutions. Many efforts that attempt to create novel systems or simply improve on existing systems, are inspired by advances in image processing, computer vision and machine learning.

In this chapter, existing literature that have contributed to the field of pose recognition and estimation are discussed. Existing literature can generally be categorized into three groups, namely, model-based, example-based and learning-based approaches. Algorithms that have significantly impacted the development of pose recognition and estimation systems using the various approaches are explored. Furthermore, for the sake of comprehensiveness, studies that have developed techniques to combine these approaches are investigated. A comparison study on these approaches is also done and, the benefits and trade-offs thereof are highlighted.

Particular emphasis is placed on the evaluation methods undertaken in the literature, where some researchers have opted for subjective evaluation and others objective evaluation methods. As this area of research mainly involves computer vision, many researchers have not explicitly stated their accuracies. Therefore, it is not possible to retrieve their accuracies when evaluating their results in some cases. The subjective evaluation methods performed by some researchers involves heuristic visual inspection to judge their results, while the objective evaluation methods, at times, involves ground truth data to determine the accuracies.

In the following subsections a brief definition on pose recognition and estimation is given, followed by an investigation on the pose recognition and estimation approaches.

2.1 Definition

The term pose recognition and estimation in this thesis refers to the process of recognising and estimating the position and orientation of a human body in single or multiple frames [101] . When estimated over multiple frames, the term human motion analysis or human body tracking is used. The objective of pose recognition and estimation involves determining the set of angles for each degree of freedom (DOF) of the joints in the human body model with respect to its local or relative coordinate system. Data captured from a single camera is represented in 2D with respect to a world coordinate system and later estimated in 3D using a 3D human body model with respect to its local coordinate system.

2.2 Pose Recognition and Estimation Approaches

Existing works in pose recognition and estimation are broadly classified into three groups: model-based, example-based and learning-based approaches. These approaches are briefly described below.

2.2.1 Model-Based Approaches

The model-based approaches involves complex model fittings and tracking frameworks [23]. These approaches assumes an explicitly known parametric model of the human body, and adopt this model with image measurements to determine and estimate the human pose that best fits the test image features [3][74][21]. The computational cost of such approaches are minimized by often employing constraints such as symmetry and degree of freedom on the models. Common features for the models includes angles between body parts and the lengths of the body parts [101]. The model-based approaches are further categorized into top-down and bottom-up methods.

2.2.1.1 Top-Down Methods

Top-down methods directly explore the high-dimensional pose space, along with the kinematic structure and corresponding constraints of the actual image observation, to reconstruct the predicted pose [68][87][35]. The probability distribution of the entire body

configurations is searched for using probability sampling techniques, such as Markov Chain Monte Carlo (MCMC)[50]. Link-joint models are roughly represented for each part of the human model. This is made up of 2D/3D geometric primitives, for example cylinders or rectangles, such that it can be fitted to the image to measure similarity [21].

The problem has been investigated by Taylor [156] in a top-down manner, by assuming corresponding points between the image and the model is provided. They also assumed that the relative lengths of each segment in the model is known and that the relationship between the points in space and the projections onto the image can be modelled as a scaled orthographic projection. Here these geometric constraints are used to estimate the individual's pose.

Similar work by Parameswaran and Chellepa [117], also applied geometric constraints to estimate an individual's pose. They use an isometric approximation where all human forms are assumed to have the same body part lengths when scaled. They also assume the torso twist is small such that the shoulder joints can be considered to have fixed coordinates. They are then able to recover the epipolar geometry of the camera and thus determine the joint positions. In other instances, smarter ways of searching or sampling means are required to efficiently explore the vast human pose space in a top-down fashion, such as the work of Maccormick and Blake [86].

Maccormick and Blake [86] have introduced an exclusion principle for tracking multiple indistinguishable body parts and also proposed a partition sampling algorithm. The approach is insensitive to the background and requires only a simple model of the body. The probabilistic exclusion principle, however, only makes use of edge-based measurements. The partition sampling algorithm makes use of particle filters and divides the state space into an arbitrary number of sub-spaces that corresponds to each of the body parts. The algorithm applies the sampling and evaluation once to each of the sub-spaces.

2.2.1.2 Bottom-Up Methods

Bottom-up methods, as opposed to top-down methods, do not use the whole body model to fit the test image. It instead fits the test image with a set of body part models [21][50][99][102]. These body part models are represented by either cylinders, rectangles or feature points, and by the geometric constraints between the parts. A list of body parts are first identified and pruned. The global geometric constraints between the parts are then used to assemble the body parts into the best possible full-body pose [21].

A bottom-up shape parsing method, where the partial body part shapes are more complete and guided by a parse tree, was proposed by Srinivasan and Shi [147]. In contrast

to previous bottom-up parsing, their functions for scoring when matching at every node do not illustrate a sub-tree independence attribute. They rather compare the shapes to the set of exemplars using the inner-distance shape context. They differ from normal parsing, which only makes use of the image features at the leaf level, by parsing multiple image segmentations at each level. A limitation to their method is their fixed parsing procedure that only starts from the lower body upwards. Furthermore, their results are qualitatively good but quantitatively poor.

By extracting potential body parts and grouping the parts into image segments in a form similar to the human body, it is possible to find people in images as in the work done by Ioffe and Forsyth [65]. By pruning the search in the image for possible body parts at an early stage, they are able to group the body parts more efficiently. Their results indicate a 49% false negative rate and a 10% false positive rate. Although kinematic constraints prove to be effective in discriminating people from non-people, the overall performance of their framework would benefit from a better body segment model such as looking for groups of features corresponding to body parts.

The strength to the bottom-up approach of Mikić et. al. [98], lies in the self-contained initialization procedure. They proposed a method of hierarchically tracking the human body by firstly detecting the head. This is followed by fitting a torso attached to the head. They then segment the remaining voxels in order to locate the upper and lower legs and arms. Their experimentation is done in a controlled environment and the results are based on the approach taken in Hunter [64] whereby the accuracies are verified by subjective evaluation.

The advantages of assembling parts using low-level segmentation have been proposed by Mori et. al. [107]. They use Normalized Cuts[1] to build torso and limb detectors which are verified using a variety of cues. To optimize the assembly of body configurations, they enforce global constraints, such as relative scales, symmetry of clothing, position and colours. To complement their approach they suggest applying artificial intelligence (AI) heuristic search methods, such as the best-first search method. They also suggest that by combining this approach with that of an example-based approach, better results will be yielded. In subsection 2.2.4 the efficacy and improvements, if any, in combining model-based with learning-based approaches in the general case as suggested by Mori et. al. [107], are investigated.

[1] A new graph-theoretic criterion for measuring the *goodness* of a portion of an image proposed by Shi and Malik [140]

2.2.1.3 Comparing Top-Down and Bottom-Up Methods

In general, the challenge with model-based approaches is that of initialisation. In the case of top-down approaches, it tries to minimize projection errors of kinematic models by either generating a large number of pose hypotheses upon which to estimate a pose [6][75] or by using numerical optimization methods [146]. With suitable initialization, this approach can produce accurate results [6] and decrease the search area as well as the search time [103], however, it can become complicated and computationally expensive [21]. It may furthermore become easily trapped in local minima [21].

Bottom-up approaches efficiently handles the high dimensionality of the human body by focusing on parts rather than the entire body [112]. This approach can handle a wider range of poses with less storage requirements. It provides better localization and is computationally inexpensive but suffers when the body part detectors fails [126].

It is these weaker points that has stemmed recent attempts to combine top-down and bottom-up approaches.

2.2.1.4 Combining Top-Down and Bottom-Up Methods

In Hua et. al. [61], a Monte Carlo simulation of the data driven belief propagation algorithm that makes use of the bottom-up visual cues, is proposed. Similar work by Lee and Cohen [76] also proposed a data driven algorithm based on the Markov Chain Monte Carlo method where proposal maps are used to generate possible 3D pose candidates during the search. Zhang et. al. [170] performed a hybrid setup, utilizing a combination of the top-down Markov Chain Monte Carlo method and the bottom-up local search to determine a 2D pose. Their hybrid setup explores the solution space efficiently and their results converge in a positive direction.

A multi-camera based approach combining top-down and bottom-up approaches to develop an efficient process to estimate the human body pose, have been implemented by Gupita et. al. [50]. This process is based on 2D likelihoods and epipolar geometry that prunes the search for likelihood regions in the 3D human body space. These likelihoods are collected only once in the image and combined in 3D using epipolar constraints. Their implementation achieved a 96% correct body part detection rate when the tolerance level for the joints error was 50% for the length of the limbs.

Another good example of a combination of top-down and bottom-up methods is, Felzenswalb and Huttenlocher [38], where a collection of colour-based part detectors that collectively represents the whole body are matched individually. The global configuration of these

body parts are found using the distance transformation. This is used to optimize a cost function in order to estimate the target body pose. Such a combination proves promising as it reduces the search complexity significantly [150] and has recently been extended to 3D body tracking from multiple views [143].

2.2.2 Example-Based Approaches

Example-based approaches utilize a database in which a group of training examples are stored with their corresponding pose or relative x, y and z coordinates. Given a query image, a comparison search on poses are performed and one or more poses with the closest matching features is returned [11] [105] [138] [151].

The challenge with example-based approaches is to accurately and efficiently search computationally expensive queries [21]. The objective of an example-based approach is to encode features from image observations that are used to identify poses from examples in a database [121]. Hayashi et. al. [53] adopted this approach and developed a system that uses a shape context algorithm on silhouettes. This algorithm treats a shape as a set of N points and is used to find correspondences between point sets. Parameters for an input image are estimated by computing the average of the body part positions corresponding to the shape context descriptors. Their system performs well for poses that have close neighbours in the database, however, it is constrained by the size of the database that contains approximately 1300 images. These images only cover a small subset of all possible human poses and is not sufficient for robust pose estimation.

The Fourier Descriptors method is a classical method to represent boundaries of shapes in object recognition and has progressed into a general method for extracting various shape forms [70]. In Poppe and Poel [122] the effectiveness of Fourier Descriptors used for recovering a pose is compared with 5 different lengths and 2 different sampling methods. They make use of silhouettes since it can be extracted relatively robustly from images. However, it should be noted that curve-based shape descriptors such as Fourier Descriptors are not feasible as silhouettes frequently change topology which causes the shape to have discontinuities [3]. In general, Fourier Descriptors also fail to achieve accurate results for images where the background is complex [60]. Their database consists of 45 656 silhouette images, generated using POSER[2] (Version 5), with their corresponding pose estimates. These poses, however, still fail to cover the entire pose domain. They achieved an average error per joint of approximately 16 to 17 degrees. Their system does not perform well when recovering poses using silhouettes with different body dimensions.

[2]POSER is a propriety 3D character modelling software developed by Curious Labs

Focusing more on the searching methods used in example-based approaches is the work of Shakhnarovich et. al. [138]. They introduced a new efficient search algorithm which employs hash functions in order to easily index and retrieve approximate nearest neighbours that are similar poses to that of a given query image with respect to both feature and parameter values. They implement edge direction histograms within a contour since they find much ambiguity in using silhouettes alone. Their database contains 150 000 training images rendered from a humanoid model using POSER. Their approach is similar to the hand pose estimation method proposed by Athitsos and Sclaroff [11], where estimation was based on a fast nearest neighbour search in the appearance domain.

Mori and Malik [106] stores a number of 2D images, from the CMU MoBo Database[3], consisting of individuals walking on a treadmill from multiple viewpoints. On each of the stored images, positions of the respective body joints are each manually marked and labelled. Given a query image, the shape context matching proposed by Belongie et. al. [13] together with a kinematic 3D body model, is used to compute a matching process to each stored image. Having found a corresponding stored image and given the 2D position of the joints, the poses are then estimated in 3D using the geometric algorithm in Taylor [156]. Most related to their implementation is the framework developed by Sullivan and Carlsson [154], that uses order structure to compare example of pose shapes to query images. Their results are visually expressed and shows that their deformable matching process performs well when edges are clear, particularly on the arms. In cases where the edges are considerably dissimilar from those of the stored image views due to clothing, the joint localization process fails.

These issues have directed many researchers to use a good image matching algorithm such as the Chamfer Distance algorithm that does not easily fail when edges are substantially different. This algorithm has proved to be an effective tool for many shape comparison fields. Cao et. al. [20] has followed this direction and proposed an approximate Chamfer Distance for identifying a pose, which achieves improvements in efficiency with slight less accuracy as compared to the exact Chamfer Distance. The approximate Chamfer Distance utilizes eigen approximations such that the distance transform can be represented in a low-dimensional sub-space. Their database consists of 14 964 images of a 3D model in various poses and angles. Their proposed implementation achieved better performance in terms of time and memory usages, however, the exact Chamfer Distance proved better in terms of accuracy.

An approach using distance level sets to skeletonise silhouettes have been implemented by Sminchisescu and Telea [145]. They proposed to design more consistent likelihood models for silhouettes in body pose estimation. Their likelihood model is composed of

[3]Carnegie Mellon University Motion of Body Database

an attraction term and an area overlap term that provides uniform model localisation. Their smoothing method for the silhouettes stabilizes the optimisation process used to estimate the pose. Both the likelihood model and the smoothing method are based on distance transform functions using level sets that allow for evolving boundaries. They do not examine their results but merely display it visually. From visual inspection, their results are satisfactory.

Body pose recognition and estimation research that makes use of the exact Chamfer Distance Transformation is that of Micilotta et. al. [96], where a variety of human upper body movements are stored in a database of 3D body configurations. The database is sub-divided into three databases, namely, hand position, silhouette and edge map database. The example in the database that yields the highest matching score is used to extract the 3D configurations for that particular pose. Their method reconstructs the upper body and locates the position of the hands such that the edges can be defined more clearly. This is followed by the Chamfer Distance Transformation to apply a matching process to identify the most likely pose from the database. They too express their results visually and show a positive result.

An example-based approach following a template-based matching technique using the Chamfer Distance Transformation similar to [20] and [96] is attempted and compared to a novel learning-based approach. When working in high dimensional space, such as human body movements, example-based approaches are often subjected to complications where there are not enough examples in the database to cover the entire body pose space. A solution would be to confine the pose space to a specific area such as sign language or pedestrian walking.

2.2.3 Learning-Based Approaches

Learning-based approaches are particularly appealing because various advanced machine learning techniques can be used and the performance of estimating a pose is fast enough for real-time applications. Most methods falling in this category follow a common structure. These methods do not assume an explicit 3D body model. Features are extracted from the original image and represented as vectors. A model is trained using these vectors and depending on a regression function that maps the data from image space to pose space, a pose is predicted [21]. There are many image features that have been used for pose recognition and estimation in particular. These include multi-scaled edge direction histograms [31], concatenated coordinates of edges [48], rectangular Haar-like features [125] [165], histogram-of-shape-context silhouette shape descriptors [3] and low level features such as Image moments (also referred to as Hu moments) of silhouette

images [130]. Similarly, there are many regression functions that have been used. Given a set of training examples, regression functions are learned from the set with the aim of globally or locally representing the relationship between the original image and the target pose. This should provide an efficient generalization to query images. Examples of these consist of Adaboost [125] [165], Relevance Vector Regression [3], Gaussian Mixture Models (GMM) [129], BoostMap [9], Bayesian Mixture of Experts [144], Expectation Maximization (EM) [130], RANSAC [94], and Support Vector Machines [128].

Addressing this problem of pose recognition and estimation from a learning-based approach is the work of Qiang et. al. [25] that applied an Implicit Shape Model-based (ISM) human detector, proposed by Leibe [77], to detect and segment the size and position of a human in an image. They extract silhouette features using a segmentation mask and the canny edge detector. Finally, they solve the 3D pose estimation as a regression problem using a Relevance Vector Machine (RVM) and ridge regression methods. Training data were created using POSER. Their implementation was tested on 50 frames created using POSER and 50 frames of real data, achieving an accuracy rate of 80%.

Instead of detecting the whole body, Ronfard et. al. [128] detect body parts using support vector classifiers that are trained based upon scale and orientation specific Gaussian derivative filters. The body part detectors comprise of 15 partially-aligned image rectangles. Their training set consist of 100 images taken from the MIT pedestrian database [4] and another 100 images for testing. When applying a geometric model with constant limitations for the body joints, they attained detection rates of 83% using SVM and 72% using RVM. However, when applying the learned geometric model, the detection rates slightly improved to 85% and 74% respectively. From a qualitative aspect, a greater number of the body part detectors were accurately placed on the test images and achieving 36% for RVM and 55% for SVM in this respect. By increasing the training size to 200 examples, the detection rates improved to 76% for SVM and 88% for RVM. Moreover, the body poses are correctly estimated in the test images, resulting in 75% for SVM and 54% for RVM.

A tracking framework proposed by Agarwal and Triggs [5] does not require explicit body models nor does it require prior labelling of body parts in images. Their approach, instead, estimate poses by making use of sparse Bayesian non-linear regression of joint angles against silhouette shapes that are extracted using histogram of shape context descriptors. Their framework uses a combination of regression methods to return multiple solutions to poses for each silhouette. That is, their regression is done over both

[4]Massachusetts Institute of Technology Pedestrian Database consisting of 924 images of pedestrians in ppm format

linear and kernel based functions using either ridge regression, Relevance Vector Machine or Support Vector Machines. Similar to [48] [56], they used POSER to generate data, relating to real human motion sequences, for a set of training and testing images. By applying these regression techniques, the average error measures for the full body using ridge regression, Relevance Vector Machine and Support Vector Machine resulted in 5.95, 6.02 and 5.91 degrees respectively. More recently, similar work by Sminchisescu [144] describe a mixture density propagation algorithm to estimate human motion.

Multiple local linear regressors are used in Okada and Soatto [115], to approximate a non-linear mapping from feature vectors based on histograms of oriented gradients to 3D poses. The histogram of oriented gradients use 8 orientation bins in 3x3 spatial cells to compute a consistent grid of overlapping regions. First pose clusters, that is a set of similar poses, are discriminated using the kernel SVM to predict the pose cluster that the particular pose is associated with. This is followed by using linear regressors to estimate the pose. In comparison to Poppe et. al. [121] who similarly used feature vectors based on histogram of gradients in an example-based approach, their approach yields better results. Using the same data (HumanEvaI dataset[5]) as in [121], the average mean errors of their approach is 37.98 as opposed to Poppe et. al. [121] with 42.85.

Early work by Rosales and Sclaroff [130] recover body poses from single images using a non-linear supervised learning architecture. Their architecture consists of a set of specialized forward mapping functions and a feedback matching function, estimated from body poses and their corresponding visual features. Image silhouettes used for training and their corresponding 2D joint configurations are generated using 3D motion capture data. These image silhouettes are used to compute Hu moments required for input. The learning problem is approached using the Expectation Maximization algorithm, that clusters the joint configurations in 2D by fitting a Gaussian Mixture Model. For each joint cluster, an inverse mapping is learned between the Hu moments and 2D joint configurations. During the feedback matching step, the most probable reconstructed configuration is selected which transforms the joint configuration back to the visual cue space.

In computer vision, particularly human body pose recognition and estimation, there is a large amount of work done in learning a low dimensional structure of a non-linear manifold embedded in a high dimensional space due to the large amounts of multi-variable data [46].

An example of this practice is the framework developed by Elgammal and Lee [36] that is based on learning view-based representations of the manifolds for a walking activity.

[5]Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion from Brown University

Their framework also learns non-linear mapping functions from the embedding space to both the 2D input space and the 3D body space. For such non-linear mapping functions, they adopt a Locally Linear Embedding (LLE) framework. This LLE framework is an unsupervised learning algorithm that maps the inputs of higher-dimensionality into a common global coordinate system of a lower-dimensionality [131]. Given a human silhouette, the body pose would be recovered in a closed form thus allowing it to recover the body configuration. They evaluated their approach using the CMU MoboGait[6] database, and achieved an overall correct classification rate of 93.05%.

An improvement to their work is that of Tangkuampien and Suter [155] that used Kernel Principal Component Analysis (KPCA) to train an optimal pose manifold using training sets of marker based human movements from the CMU motion capture database. Similar to Elgammal and Lee [36], a silhouette manifold is also trained, whereby marker-less based human movements is simply regarded as a mapping from the silhouette manifold to the pose manifold. Using LLE [131] reconstruction after training, unseen silhouettes are projected through the silhouette and the pose manifold. The estimated pose is then determined using the pose manifold to calculate the pre-image of the LLE reconstructed vector. Their results show that for 1260 test silhouettes, they obtain close reconstructions of the original body pose including an average error of 2.86 degrees per joint.

Unlike [36] and [155], Grochow et. al. [49] presented an inverse kinematics system that learns from previously seen poses. They define their inverse kinematics system as a maximisation of an objective function which states the suitability of a pose. Using different input data when training the model leads to different styles of inverse kinematics. They introduced a model called Scaled Gaussian Process Latent Variable Model (SGPLVM) that represents the probability distribution across all possible poses. Their inverse kinematics system is therefore able to form any pose, although poses similar to those in their training data are preferred by their system.

2.2.4 Combining Model-Based with Learning-Based Approaches

Both model-based and learning-based approaches have their relative strengths and weaknesses. By bringing together these approaches, it is possible to combine the advantages of both, while partially overcoming the disadvantages of either approach.

Thayananthan et. al. [157] improved upon Tipping and Faul's [160] bottom-up method by developing a multivariate generalization for training a sparse RVM regressor. Their system learns a one-to-many mapping that maps from feature space to state space. Furthermore, their system matches a set of image shape templates against the edge

[6] Carnegie Mellon University Motion of Body Database

map of an input image to obtain Hausdorff matching scores. To distinguish between the poses, these scores are mapped to different state-space regions by learning a set of RVM mapping functions. Each mapping function covers a wide pose space by selecting a small subset of the total set of shape templates. Their work is closely related to that of Sminchisescu et. al. [144] and Argarwal et. al. [4] [5]. In contrast to their work, Thayananthan et. al. [157] verifies the output of each mapping function using a likelihood estimation by projecting the 3D model.

A system combining a model-based approach with a learning-based statistical approach was proposed by Jaeggli et. al. [66]. Using a sparse kernel regressor, they are able to learn the relationship of the body pose and the image appearance. By adopting the LLE dimensionality reduction, the body poses are modelled on a low-dimensional manifold. Along with the image appearance, a non-linear dynamic model is learned from a previous model of possible body poses. Their training set consisted of 4 000 body poses of people walking. All the kernel regressors were trained with Gaussian kernels using the RVM algorithm. Their experiments are visually expressed and show a positive result.

Another study that combines the two approaches was proposed by Roberts et. al. [127], whereby probabilistic region templates used to detect body parts are presented. The likelihood ratios for single parts are learned from the dissimilarity score between the appearance distributions of the foreground and adjacent background. Moreover, the likelihood ratio is also learned for pairs of body parts with similar appearance. Their approach of partial configurations merges top-down and bottom-up approaches by allowing configurations of various dimensionalities to be compared. This is achieved by combining learned likelihood ratios computed only from the visible body parts. Despite having combined model-based with learning-based approaches and attaining a positive visual estimation result, their process is computationally expensive.

2.2.5 Combining Example-Based with Learning-Based Approaches

Little work has been done in combining example-based with learning-based approaches. Example-based approaches store a set of exemplar images with the corresponding known 3D postures. This is combined with a learning-based approach to learn a model to efficiently search exemplars similar to the query image. Ren et. al. [125] are able to control a 3D model to follow a human dancing motion by selecting the best local features from 2D silhouette images to estimate the yaw orientation and body configurations of the user. In contrast to Shakhnarovich et. al. [138] that implement edge direction histograms as feature vectors, they implement Haar wavelet-like features to compute feature vectors from the silhouette images. The selection of Haar wavelet-like features is

based on learning a set of hashing functions with the Adaboost algorithm. Estimating the yaw angles incorporates the Locality Sensitive Hashing (LSH) whereas estimating the body configuration depend on the stored temporal poses in a domain-specific database. By combining Adaboost and LSH, their solution allows for quick silhouette based yaw estimation, however, it is still limited due to the approach depending on a domain-specific database. First Adaboost is used to train a number of independent hash functions, followed by selecting the LSH functions randomly among them. The top 20 matches retrieved by LSH is used for yaw estimation. This yaw estimation obtained from their system is compared to ground truth motion captured data. Their system resulted in an LSH-aided yaw estimation of lower than 10 degrees for 73% of the images, lower than 20 degrees for 92.5% of the images and lower than 30 degrees for 98% of the images. Despite the results for the yaw estimation errors, their tests reveal the body configurations estimation to be robust.

2.3 Summary

In the preceding sections the approaches used when providing a possible solution for pose recognition and estimation, were described and analysed. As each of the approaches have their relative strengths and weakness, one of the fundamental concerns is, which approach would be more suitable for a sign language translation system. Unfortunately, little work has been done in combining the approaches, however, those studies that have attempted to combine the approaches were reviewed. Furthermore, the results obtained from these combined approaches are comparable to the three approaches individually. It is therefore not necessary to develop a combined approach but rather focus on developing a system using an individual approach that solves the pose recognition and estimation problem and, is practical as an application for sign language. When developing a system as an application for sign language, automatic initialization of the system is essential and since model-based approaches often suffer in this respect, it would not be feasible to pursue this approach. Therefore, an example-based and learning-based approach is rather considered for such a system.

In example-based approaches, various issues arise concerning feature-based measurements. These issues have directed many researchers towards good image matching algorithms. One that stands out is the Chamfer Distance algorithm that has proved to be an effective tool for many shape comparison fields. As for learning-based approaches, various advanced machine learning techniques can be used to develop real-time applications. Given these factors, it would be instructive to do a comparison between the two

approaches. Herewith, Chapter 3 and Chapter 4 is dedicated towards an example-based and learning-based approach respectively.

In this chapter an extensive study on model-based, example-based and learning-based approaches is provided. Studies where these approaches have been combined are also investigated. Furthermore the approaches have been compared and settled upon implementing an example-based approach as well as introducing a novel learning-based approach. Finally, the approaches will be compared in terms of accuracies and concluded on its suitability towards a South African sign language recognition and translation system.

Chapter 3

Example-Based System

This chapter discusses the components that form part of the example-based system for computing a template match of upper body poses. The objective of an example-based approach is to encode features from image observations that are used to identify poses from examples in a database [121]. Before these features can be extracted, image registration is required to prepare the images. Part of the image registration is face detection and greyscaling. In the following subsections these two techniques are investigated and followed by investigating the feature extraction methods consisting of an edge detector and a distance transformation technique. These techniques are explained and compared with closely related techniques. To compile the database, a 3D human body model is used to generate a large number of poses. An overview of the model is included in the subsections that follow.

3.1 Face Detection

The face detection method is a common method in many image registration techniques and is important in both the example-based and learning-based systems, for two reasons:

1. It is used to identify when an individual is present before the camera for the purpose of discarding all unnecessary previous frames.
2. It provides consistency by using the coordinates of the individual's face, along with the width and height of the individual's head in order to reposition the image such that each individual would be on the exact same position in the image.

For this method to succeed, the assumption is that each individual will face directly towards the camera. In sign language, it is considered rude and disrespectful to not

face directly towards the person with whom one is having a conversation with using sign language [139][148][91]. The above assumption is therefore a realistic constraint.

The OpenCV library's implementation of the face detection technique initially developed by Viola and Jones [165] is employed. This face detection method is a tree-based technique using Haar feature classifiers to build a boosted rejection cascade. The Viola and Jones [165] face detection uses AdaBoost at every node in the cascade in order to achieve a positive detection rate. This is possible by using a low rejection rate multi-tree classifier at every node in the cascade [16]. Their algorithm incorporates four innovative features [16]:

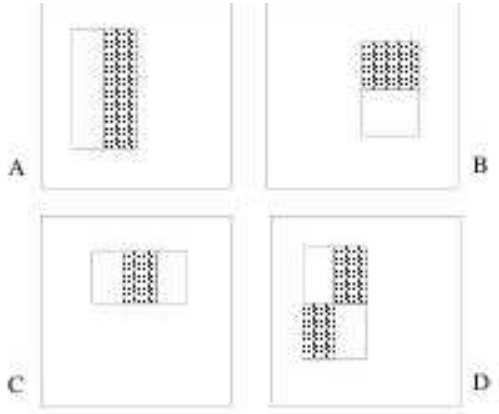
1. Haar-like wavelet features for input.
2. They introduce a new form to represent an image, referred to as an *integral image*.
3. A learning algorithm using statistical boosting, based on AdaBoost.
4. The organization of weak classifier nodes in a rejection cascade.

These concepts are discussed further in subsequent sections.

3.1.1 Haar-like Wavelet Features

Single wavelength square waves, where there is a single high interval and a single low interval, is referred to as Haar-like wavelets. A wavelet is a pair of light and dark rectangular regions that have identical size and shapes and are either vertically or horizontally adjacent [165]. These wavelets consist of three types of features, namely, a two-rectangle feature, a three-rectangle feature and a four-rectangle feature. An example of the features used in the face detection method are shown in Figure 3.1.

FIGURE 3.1: The Haar-like wavelet features used in the face detection method [165]



The two-rectangle feature is computed by differencing the sum of the pixels within the two regions. The value of the three-rectangle feature is determined by adding the pixels within the two outside regions and subtracting from the sum of the pixels in the centre region. The value of the four-rectangle feature is determined by computing the difference between the diagonal region pairs [165]. A threshold is given to the result and the value determined, indicates a presence or absence of Haar-like features. Computation of Haar-like wavelet features occurs at multiple image locations over multiple scales. To efficiently compute these features at multiple scales, Viola and Jones [165] introduced the integral image representation for images.

3.1.2 Integral Image

The integral image is an alternative representation for an image so that computation of the features can be done efficiently. The integral image is the sum of all the pixels to the left and above the corresponding pixel in the original image. The equation for the integral image is [165]:

$$integral\_image(x, y) = \sum_{x_i \leq x, y_i \leq y} original\_image(x, y) \tag{3.1}$$

where $integral\_image(x, y)$ is the integral image and $original\_image(x, y)$ is the original image. Furthermore, the following equation pair of recurrences is used where the cumulative sum of a row is represented by $sum(x, y)$ [165]:

$$sum(x, y) = sum(x, y - 1) + original\_image(x, y) \tag{3.2}$$

and

$$integral\_image(x, y) = integral\_image(x - 1, y) + sum(x, y) \tag{3.3}$$

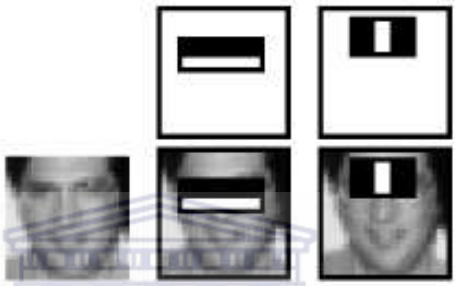
where $sum(x, -1) = 0$ and $integral\_image(-1, y) = 0$. Using the above equations, the integral image is computed with only a single pass over the image.

3.1.3 The AdaBoost Learning Method

In the Viola and Jones [165] face detection method, a modified AdaBoost algorithm is used that not only selects a small set of features but also trains the classifier. The purpose of the AdaBoost learning method, in its original form, is to create a *strong* classifier by combining many *weak* classifiers. Classifiers are defined as weak when such classifiers correctly recognises a feature more often than one who guesses. In Viola and

Jones [165] approach, weak classifiers are designed to select single rectangle features that distinguishes best between positive and negative input examples. For each Haar-like wavelet feature, a threshold-based binary classifier is built, such that the weighted training error is minimized [166]. Ultimately, the AdaBoost learning method selects a set of weak classifiers, combines them, and assigns weights to each classifier (sometimes referred to as *boosting*). At each interval of boosting, the single best weak classifier for that interval is chosen. The outcome would thus be a strong classifier consisting of a weighted combination of weak classifiers. An example of features selected by AdaBoost is shown in Figure 3.2.

FIGURE 3.2: Features selected by Adaboost for detecting the face [165]



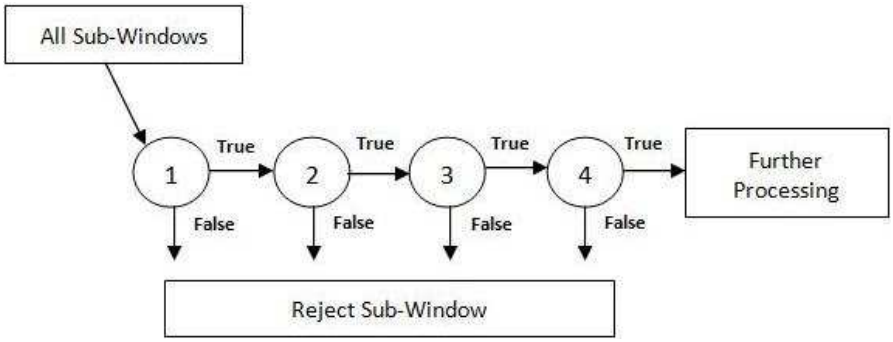
3.1.4 Organisation of Weak Classifier Nodes in a Rejection Cascade

The fourth component introduced by Viola and Jones' approach is a method that organises the combination of weak classifiers in a cascade structure that greatly increases the speed by only focusing on promising regions in the image. The reason behind only searching on these regions is to rapidly determine where a possible face might be in the image so that further complex processing is set aside for these regions. The overall detection process of these regions takes the structure of a degenerate decision tree and is referred to as a cascade.

The initial selected classifier is applied to the image regions and a positive result indicates a possible face might be in that region. This process is reiterated by a series of classifiers, each being increasingly more complex than the previous one. If at any point a negative result is obtained for an image region, it gets rejected and no further processing is performed on that region. If the image region obtains a positive result through all the classifiers, then there exists a face in the image region. The detection process is illustrated in Figure 3.3

The classifiers are selected by AdaBoost according to weights assigned to the classifiers. The classifiers with heavier weights are selected first in the cascade so that image regions that do not contain a face can be eliminated quickly.

FIGURE 3.3: The detection process for rejecting regions in the image [165]



3.1.5 Experimentation on the Face Detection Method

The face detection method presented in Viola and Jones [165] was tested on the MIT CMU [1] frontal face test set [132]. The test set from this database contains 130 images with 507 labelled frontal faces. Their results as well as the results for other published systems are illustrated in Table 3.1.

TABLE 3.1: False positive detection rates for Viola-Jones and other published systems on the same test set [165]

False Detections / Detector	10	31	50	65	78	95	167
Viola - Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2%	93.7%
Viola - Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Rowley - Baluja - Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman - Kanade	-	-	-	94.4%	-	-	-
Roth - Yang - Ahuja	-	-	-	-	94.8%	-	-

The face detection method is an important and required step in both the example-based and learning-based methods. It is therefore imperative to perform experimentation to ensure that the face detection method firstly works and secondly, that it is an accurate face detection method to deploy. To evaluate their method, a test set consisting of 1047 randomly selected images were used. These images were taken from the internet and other sources, with different background complexities, illumination, scales and camera variations. From the experimentation an 88.9% detection rate is obtained on frontal faces in the test set. This result is comparable to the results achieved by Viola and Jones [165] themselves. Some of the results obtained from the experimentation is shown in Figure 3.4

The results therefore confirm the results obtained by Viola and Jones [165]. The method has been shown to not only work but also achieve results of a high accuracy and is ideal for both systems proposed in this research.

[1] Carnegie Mellon University

FIGURE 3.4: Results obtained using the face detection method

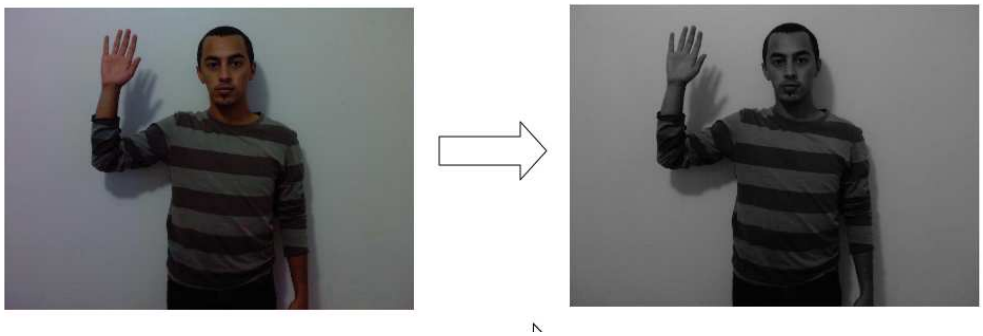
3.2 Greyscale Transformation

A colour image is formed by a combination of three values for each pixel, namely a red, green and blue pixel value. A greyscale image is formed by replacing these with a single value for each set of red, green and blue pixel. The greyscaled image holds the full information about the image intensity, which is conveyed by several shades of neutral grey, varying between the weakest intensity being black and the strongest being white. The purpose of transforming a colour image to greyscale is that less information is required for each pixel; hence decreasing the number of computations by a third [1]. It is also a pre-requisite for the edge detection method implemented in this research. In order to apply this transformation, certain ratios should be applied to the pixel values in the colour image. This transformation can be formulated in the following equation:

$$(R * 30\%) + (G * 59\%) + (B * 11\%) = GreyscalePixel \tag{3.4}$$

where R is the red pixel value, G is the green pixel value and B is the blue pixel value. The transformation is also visually presented in Figure 3.5.

FIGURE 3.5: The greyscale transformation performed on a colour image.



3.3 Edge Detection

A common factor present in pattern recognition, image segmentation and shape-based object recognition is feature extraction within images. These features usually tend to be

corners and edges [114]. An edge is a contour across which there exists a sharp change in image intensity. Edge detection algorithms are generally a high-pass filter that can be used to extract edge points in colour or greyscaled images and similarly generate binary feature maps [78]. Ideally, in shape-based object recognition, the edge of objects and sudden changes in colour should be represented by edge points in a contour form; however, edge detection algorithms are often affected differently by parameters such as illumination conditions, geometrical properties of objects, objects of similar intensities and particularly the level of noise contained in images. It is for this reason that various edge detection algorithms have been developed with the aim of finding the perfect one. Too many edge detection algorithms exist for all to be included here. Therefore the attention is focused on those that are well known and have been widely used. These algorithms may be grouped according to the following differentiation operators, namely, gradient-based, Laplacian of Gaussian and Gaussian-based edge detection algorithms.

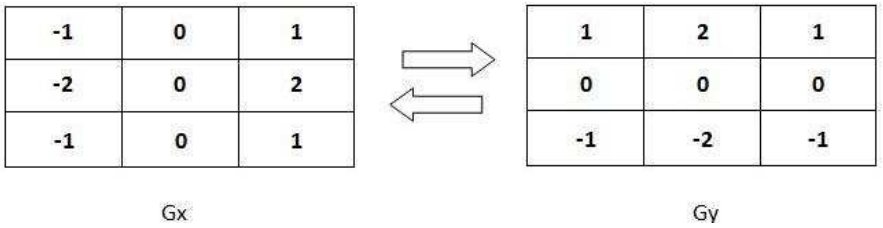
3.3.1 Gradient-Based Edge Detection Algorithms

Gradient-based algorithms find edges by locating the maximum and minimum in the first directional derivative of the image [100]. It consists of classical edge detection operators such as Sobel, Prewitt and Roberts.

3.3.1.1 Sobel Operator

The Sobel operator locates high spatial frequency regions corresponding to edges by performing a 2D spatial gradient measurement on images [89]. It comprises two 3x3 convolution kernels similar to the Prewitt operator as illustrated in Figure 3.6.

FIGURE 3.6: The 3x3 convolution kernel of the Sobel operator [89]



From Figure 3.6 it should be noted that kernel Gy, is simply kernel Gx rotated by 90 degrees. This design allows the kernels to maximally respond to vertical and horizontal edges, one for each orientation. The gradient magnitude for the operator is given by:

$$|G| = \sqrt{Gx^2 + Gy^2} \tag{3.5}$$

For a much faster computation, the approximate absolute gradient magnitude is used at each point in a greyscaled image and is given by:

$$|G| = |Gx| + |Gy| \tag{3.6}$$

Furthermore, the angle of orientation at each point gives rise to the spatial gradient and is given by:

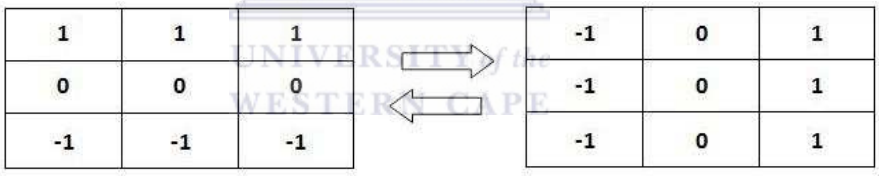
$$\theta = arctan(Gy/Gx) \tag{3.7}$$

The average operator used in the Sobel algorithm is more like a Gaussian and therefore better at removing white noise.

3.3.1.2 Prewitt Operator

The Prewitt operator shares similar edge patterns to the Sobel operator. The kernels maximally respond to vertical and horizontal edges, one for each orientation [89]. The gradient-based edge detector comprises two 3x3 convolution kernels as illustrated in Figure 3.7.

FIGURE 3.7: The 3x3 convolution kernel of the Prewitt operator [89]



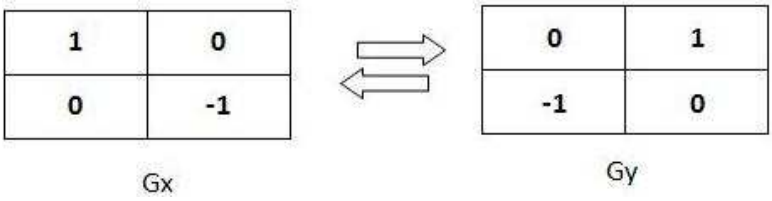
The operator is less susceptible to noise since it differentiates and averages in opposite directions; however, due to this average operation, the position of the edges might be changed.

3.3.1.3 Roberts Cross Operator

The operator locates high spatial frequency regions corresponding to edges by performing a more efficient 2D spatial gradient measurements on images [89]. Unlike the Sobel and Prewitt operator, the Roberts cross operator comprises of two 2x2 convolution kernels as illustrated in the Figure 3.8.

The smaller kernels make it more susceptible to noise and maximally respond to edges with a slope around 45 degrees, with a single kernel for each orientation. In each orientation, the kernels are applied to a greyscaled image to produce measurements for

FIGURE 3.8: The 2x2 convolution kernel of the Roberts cross operator [89]



the absolute magnitude of the gradient. This is given by the following equation:

$$|G| = \sqrt{Gx^2 + Gy^2} \tag{3.8}$$

Furthermore, the angle of orientation at each point gives rise to the spatial gradient and is given by:

$$\theta = arctan(Gy/Gx) - 3\pi/4 \tag{3.9}$$

3.3.2 Laplacian of Gaussian Edge Detection Algorithm

This algorithm was invented by Marr and Hildreth [90] who opted to combine Gaussian filtering with the Laplacian; hence, the Marr-Hildreth edge detection algorithm. Zero crossings of the Laplacian are searched for in the second derivative of the image with the aim of highlighting regions of rapid change in intensity, thereby finding edges in the image. Given a greyscaled input image, a 3x3 convolution kernel can be used to approximate the second derivatives of the Laplacian. There are three commonly used kernels [89] that are illustrated in Figure 3.9.

FIGURE 3.9: The three commonly used 3x3 convolution kernels of the Marr-Hildreth algorithm [89]

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

These kernels are very sensitive to noise since they approximate the second derivatives of the image. A Gaussian is therefore used to smooth the image as the smoothing helps to reduce the amount of error caused by noise. A 2D Laplacian filter is then applied to the image and given by the equation:

$$L(x, y) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2} \tag{3.10}$$

where $L(x, y)$ is the Laplacian of the image and $f(x, y)$ is the pixel intensity values. The edges can therefore be defined as the magnitude of the gradient vector at each spatial

location and the Laplacian of Gaussian function given by:

$$\nabla^2 LoG(x,y) = \frac{1}{\sigma^2}(\frac{x^2+y^2}{\sigma^2} - 2)e^{-(\frac{x^2+y^2}{2\sigma^2})} \tag{3.11}$$

where $LoG(x,y)$ is the 2D Laplacian of the Gaussian function centered on zero and σ is the Gaussian standard deviation.

3.3.3 Gaussian Edge Detection Algorithms

These algorithms are symmetric along the edges and the reduction of noise is achieved by smoothing the image similar to the Marr-Hildreth algorithm. The most significant Gaussian edge detection algorithm is the Canny algorithm which is considered to be the optimal Gaussian edge detection algorithm to use. It was invented by John Canny [19] at MIT [2] in 1986 and continues to outperform many newer algorithms that have been developed [109]. This algorithm is ideal for images corrupted by noise. Canny [19] formulated this algorithm according to three criteria. The first criterion is reliable detection with low error rate, i.e. true edges should not be missed and the probability of detecting non-edges should be low. The second criterion is good localisation of edge points, i.e. there should be a minimal distance between the edge points as found by the detector and the true edge position. The third criterion is to eliminate multiple responses and have only one response to a single edge.

Based on these criteria, the algorithm optimally smoothes the image using Gaussian filtering to reduce noise. It searches for the image gradient in order to locate areas with high spatial derivatives. It applies non-maximum suppression along these areas. Thus, if a pixel is not considered to be maximum, it is suppressed, achieving good localisation. Given a 3x3 region, if the value of a point is greater than the value of either points on the side of it, then it is considered maximum. Single edge points are then located in response to changes in image intensity by using hysteresis thresholding to further reduce the gradient array. Hysteresis thresholding is applied to the pixels which have not been suppressed. It makes use of two thresholds, a low threshold and a high threshold. If the magnitude of a pixel is below the low threshold, it is set as a non-edge. On the other hand, if the magnitude of a pixel is above the high threshold, it is set as an edge; however, if the magnitude of a pixel is between the low and high threshold, then it is set as a non-edge unless the pixel is the neighbour of an edge pixel.

[2] Massachusetts Institute of Technology

3.3.4 Comparison of Edge Detection Algorithms

Edge detection is an essential task for shape-based object recognition especially for the Chamfer Distance Transformation, for which it is a pre-requisite. It is therefore necessary to obtain true edges and keep false edges at a minimum. These algorithms, however, are often affected differently by factors such as illumination conditions, geometrical properties of objects, objects of similar intensities and noise levels contained in images. In this respect, these algorithms are compared, discussed and an overview presented in Table 3.2

TABLE 3.2: Advantages and disadvantages of the edge detection algorithms

Algorithm	Advantages	Disadvantages
Classical	• Edges are detected along with orientation • Simple to use	• Very sensitive to noise • Weak localisation and inaccurate
Laplacian of Gaussian	• A larger area around the center pixel is tested for edges • Good localisation	• The orientation of edges cannot be found due to the Laplacian Filter • Does not work well at corners and curves
Gaussian	• Good localisation • A good response to single edges • Not as sensitive to noise • A low error rate • Works well at corners and curves	• Time consuming • Computationally complex

Figure 3.10 presents a visual comparison of the edge detection algorithms by applying each algorithm on an image and presenting it alongside each other.

Based on the comparisons performed, the Canny edge detection algorithm is a more suitable technique to apply. This algorithm is the preferred choice since it has good localisation and a good response to single edges. It is not as sensitive to noise and has a low error rate. It furthermore works well on corners, curves and produces single

FIGURE 3.10: Edge detected images using the various algorithms for a visual comparison

continuous pixel thick edges. With more advanced computers being developed at present, the disadvantages to this algorithm can be eliminated. Adopting this algorithm in the example-based system is ideal for preparing the images for the matching process.

3.4 Silhouette Shape Matching

In the study on human body pose recognition and estimation, multiple visual cues such as texture, colour and shape are used to recognise the human body. When texture and colour is not present, most bodies can still be recognised by their body geometry alone. A common approach is to search for a shape pattern in an image. This concept gives rise to the task of shape matching. There are several shape matching techniques that have been used. In some of these techniques, such as the work of Hayashi et. al [53] and, Sullivan and Carlsson [154], a set of point correspondences are found and set up between shapes followed by a transformation that adjusts the shapes.

This procedure forms the principle of methods such as shape context matching [53] or iterated closest points [154]. These methods, however, perform well only for shapes that have close neighbours in the database [53] and fail when points are considerably dissimilar from those of the stored image views [154]. Another exemplary technique often used in shape matching is Fourier Descriptors which is used to represent boundaries of shapes, however, it is not feasible for human body pose recognition as silhouettes

frequently change topology which thus causes the shape to have discontinuities [122]. A more effective way for matching silhouette shapes is to measure the distance between points in a shape to identify a possible pose. A manner in which to measure this distance accurately, is to use the Euclidean Distance Transform.

3.4.1 Euclidean Distance Transform

The Distance Transform is an operation which creates a distance map by computing the distance of non-edge points to the nearest edge points by allocating integer values to the edge and non-edge points on a single image. The purpose of the distance map is to use the mapping as a component when performing the matching feature. The Euclidean distance is defined as the length of a straight line between two fixed points [104]. In a binary image, these points are represented by low-level features such as edges. Given two binary images P(x,y) and Q(x,y), the Euclidean distance between the 2D points can be computed by the following equation:

$$D_{Euclidean}(\mathbf{i},\mathbf{j}) = \sqrt{(i_1 - j_1)^2 + (i_2 - j_2)^2} \tag{3.12}$$

Furthermore, the Euclidean distance must satisfy the following properties [62][32].

- if $D_{Euclidean}(\mathbf{i},\mathbf{j}) = 0$ then $\mathbf{i} = \mathbf{j}$ (reflexive property)
- if $D_{Euclidean}(\mathbf{i},\mathbf{j}) \geq 0$ then $\mathbf{i} \neq \mathbf{j}$ (non-negativity property)
- $D_{Euclidean}(\mathbf{i},\mathbf{j}) = D_{Euclidean}(\mathbf{j},\mathbf{i})$ (symmetry property)
- if $D_{Euclidean}(\mathbf{i},\mathbf{j}) \leq D_{Euclidean}(\mathbf{i},\mathbf{h}) + D_{Euclidean}(\mathbf{h},\mathbf{j})$ where $\mathbf{h}$ is a third point, then $\mathbf{i} \neq \mathbf{j} \neq \mathbf{h}$ (triangle inequality rule)
- There is only one $D_{Euclidean}(\mathbf{i},\mathbf{j})$ for the points $\mathbf{i}$ and $\mathbf{j}$

From this set of properties, the reflexive property is of importance. When this property holds, it indicates an exact match between two images. Although the Euclidean Distance Transform provides an accurate and exact measurement for shape matching, the computational cost is relatively high and more importantly it fails when the exact matching pose of the query image is not found in the database [12][104][141][43]. Instead, several efficient algorithms have been developed to compute integer approximations of the Euclidean distance. Each of these approximation algorithms differ in terms of accuracies relative to the exact Euclidean distance. The approximation algorithms are discussed in the subsequent sections.

3.4.2 City-Block Distance Transform

The city-block distance or sometimes referred to as the Manhattan distance, is defined as an approximation in which the distance between two points is the sum of the absolute differences of their positions [104]. The distance between the 2D points is measured by the number of vertical and horizontal steps required to traverse an image grid. This is computed by the following equation:

$$D_{cityblock}(\mathbf{i}, \mathbf{j}) = |i_1 - j_1| + |i_2 - j_2| \tag{3.13}$$

The city-block distance is easy to compute since it can be recursively accumulated by considering the 4 nearest neighbours to each pixel. The distance is therefore measured as the minimum number of the 4 nearest neighbours; however, the diagonal distances are over estimated since the diagonal neighbours count as 2 steps instead of $\sqrt{2}$ [12].

3.4.3 Chessboard Distance Transform

Another commonly used approximation is the Chessboard Distance Transform. The chessboard distance is defined on a vector space whereby given two vectors, the distance is the maximum of the absolute value of the difference between the points in a vector. The distance between the 2D points is measured by the number of steps a *king* on a chessboard requires to traverse an image grid [104]. This can be computed by the following equation:

$$D_{chessboard}(\mathbf{i}, \mathbf{j}) = max(|i_1 - j_1|, |i_2 - j_2|) \tag{3.14}$$

Similar to the city-block distance, the chessboard distance is easy to compute since it can be recursively accumulated and differs by considering the 8 nearest neighbours to each pixel. The distance is therefore measured as the minimum number of the 8 nearest neighbours, and unlike the city-block distance, the diagonal distance are under-estimated since the diagonal steps count as a single step [12].

3.4.4 Chamfer Distance Transform

Similar to other approximation algorithms, the Chamfer Distance Transform is characterised by a local mask based on the Chamfer metric. It is defined as a transformation process which uses edge points from two different images and calculates the best fit between the two, by simply minimising a generalised distance between them [14]. To

formalize this definition, given the set $I = \{x_i\}_{i=1}^{N}$ whose elements are the edge points of a silhouette in the database and given the set $J = \{y_i\}_{i=1}^{M}$ whose elements are the edge points of a query image, where N and M denote the number of points in the set I and J, respectively. The Chamfer distance from I to J can be computed as the mean distance of points in I to the nearest points in J and given by the equation [158]:

$$D_{Chamfer}(I, J) = \frac{1}{N} \sum_{x_i \epsilon I} min_{y_i \epsilon J} ||x_i - y_i|| \tag{3.15}$$

The technique assigns values to every pixel with respect to its distance from a given pixel to the nearest edge pixel. The process involves scanning an image twice, by using a mask. The 3x3 mask is the most commonly used and have been further extended to 5x5 and 7x7 masks by [14]. Borgefors [14] recommends using the (3:4) approximation for the 3x3 mask and the (5:7:11) approximation for the 5x5 mask. The 7x7 mask, however, shows no significant improvements [14][104]. Therefore, the commonly used 5x5 mask with the (5:7:11) approximation is preferred. The (5:7:11) approximation is expressed in the following equation:

$$\begin{aligned} V_{i,j}^{k} \quad = \quad & minimum(V_{i-2,j-1}^{k-1} + 11, V_{i-2,j+1}^{k-1} + 11, V_{i-1,j-2}^{k-1} + 11, V_{i-1,j-1}^{k-1} + 7, \\ & V_{i-1,j}^{k-1} + 5, V_{i-1,j+1}^{k-1} + 7, V_{i-1,j+2}^{k-1} + 11, V_{i,j-1}^{k-1} + 5, V_{i,j}^{k-1}, V_{i,j+1}^{k-1} + 5, \\ & V_{i+1,j-2}^{k-1} + 11, V_{i+1,j-1}^{k-1} + 7, V_{i+1,j}^{k-1} + 5, V_{i+1,j+1}^{k-1} + 7, V_{i+1,j+2}^{k-1} + 11, \\ & V_{i+2,j-1}^{k-1} + 11, V_{i+2,j+1}^{k-1} + 11) \end{aligned} \tag{3.16}$$

On applying the mask to the edge image, the resulted distance transformed image will consequently correspond to an approximation of the distance from a pixel to the closest edge pixel [1].

3.4.5 Comparing Approximate Distance Transforms

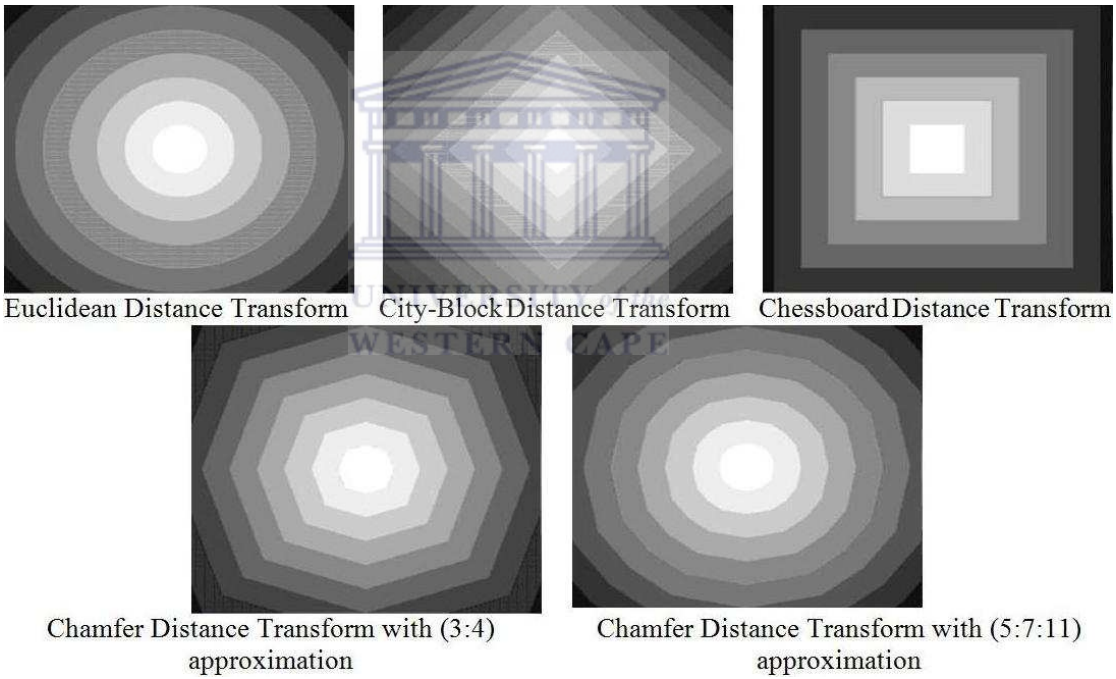
It is important that the approximate distance transform algorithm used in the shape matching method is a close enough approximation of the Euclidean distance, as this will contribute to a more accurate measurement when performing the matching process [24]. In order to find an approximate distance transform with minimal error to the exact Euclidean distance, the upper limit for the difference between an approximate metric and the Euclidean metric can be used to perform a comparison. This upper limit has been computed in Cuisenaire [28] and is as follows:

- The difference between the city-block metric and the Euclidean metric is $-0.59M$

- The difference between the chessboard metric and the Euclidean metric is $0.41M$
- The difference between the Chamfer metric and the Euclidean metric is $0.08M$

where M is the absolute difference between two points. It should be noted that the upper limit for the Chamfer metric is more preferable than the upper limits for the chessboard and the city-block metrics. Furthermore, when using the (5:7:11) Chamfer approximation, the upper limit is further reduced to $0.02M$ [28]. A more accurate distance measure can also be obtained by using a larger mask size since more terms can be compared. A 5x5 mask size is found to provide an acceptable trade-off between approximation accuracy and computational complexity [12]. Given a single point in the centre of an image, the distance transforms relative to this point is compared and illustrated in Figure 3.11. From the comparisons performed, it is more beneficial to use the (5:7:11)

FIGURE 3.11: The distance transforms relative to a single point in an image [104].



Chamfer approximation with a 5x5 mask. This will provide a closer approximation to the Euclidean Distance and a more accurate component in the matching technique.

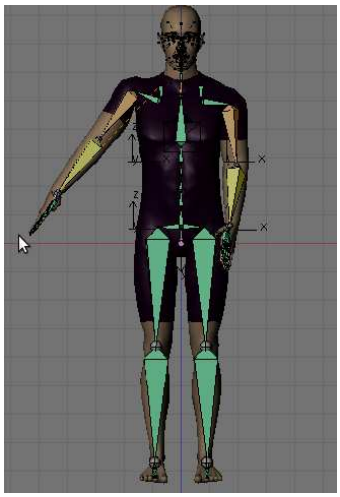
3.5 3D Human Body Model

Human body models are commonly used in computer graphics as a tool for creating images of human figures [92]. These models are often referred to as avatars, animated characters, humanoids or virtual humans. The appearance, shape and kinematic properties of the human body are incorporated into the human body models. The human body

is formed by body parts where each body part is linked to another by joints and each joint has a specific degree-of-freedom (DOF). The DOF describes the extent to which the movement of the joints in the body are limited. A combination of DOF in the body model depicts a pose relevant to the human body. This research uses 3D coordinates rather than 3D angles to represent the human body pose as the intention is not to encode the global orientation of the human body. Additionally, based on the assumption that the person will be facing the camera and performing sign language, the hip joint which is located at the origin point will be fixed and therefore have a constant 3D coordinate. The articulated nature of the human body is thus imitated by the body models in either 2D or 3D skeletal structures. In order to develop a correct pose estimation system, the human body model framework needs to produce physically valid posture estimates [97]. For postures to be physically valid, a number of constraints are assigned to the 3D model according to the DOF of the human.

POSER, which is a software package that renders and animates human figures in 3D form similar to the way humans move, is often used by many pose estimation systems [122][138][25][5]. An alternative 3D model was developed by Van Wyk [163], a former student in the South African sign language group. He used various standards and open technologies to develop an open framework that models and animates virtual 3D human body models. The framework was developed with the aim of effectively visualizing sign languages. The 3D body model consists of a mesh model, material, textures and bones. The mesh model is made up of a net of connected polygons defined by 3 or more vertices. The material in conjunction with the texture specifies how the 3D model will be visualized, indicated by Figure 3.12. The bones collectively specifies the skeleton of the 3D model. The framework incorporates standards and technologies which includes Python, Blender, H-Anim and MakeHuman. The H-Anim standards was extended in his system and adopted to effectively perform sign language movement.

FIGURE 3.12: The material and texture of the 3D model.



This research uses his system as it enables one to easily manipulate any joint into any position as required. It is possible to simultaneously retrieve the x, y and z coordinates for the respective joints, thus providing ground truth data. The main focus is on finding the positions of the shoulders, elbows and wrists, as these are the primary joints used when performing sign language. Figure 3.13 illustrates a single pose performed by the 3D model.

FIGURE 3.13: A single pose performed by the 3D model

Furthermore, it is possible to animate a Sign Language phrase while simultaneously estimating poses throughout the phrase, using Blender's interpolation solver.

3.6 Summary

In the preceding sections the components that make up the example-based system were investigated. A comparison was made between existing face detection algorithms and it was concluded that the Viola and Jones algorithm was the most suitable one. An experimental test was also performed to ensure the reliability of the algorithm, as the system requires an accurate detection of the face. This requirement ensures that each person can be moved to a consistent position as those stored in the database. The experiments indicated a high detection rate which merits the algorithm and proves the suitability thereof for the example-based system.

Furthermore, a description of the greyscale method is given. The Gradient, Laplacian of Gaussian and Gaussian edge detection algorithms are also compared. The Distance Transformation method in this system requires a *clean* edge detected image. In this regard, the Canny operator proved suitable. In order to match silhouettes with examples in the database, a distance transformation technique is needed. Although the

Euclidean Distance Transform provides an accurate and exact measurement for silhouette shape matching, the computational cost is relatively high and more importantly it fails when the exact pose to that of the query image is not found in the database [12][104][141][43]. Instead, efficient algorithms have been developed to compute integer approximations thereof. Thus, approximation algorithms are compared and the Chamfer Distance Transform is preferred since it provides closer approximations to that of the Euclidean Distance Transform. Furthermore, the (5:7:11) approximation with a 5x5 mask is used to achieve a more accurate distance measure. To generate a large number of poses for the database, the 3D model developed by van Wyk, that incorporates standards and technologies which includes Python, Blender, H-Anim and MakeHuman, is used.

In Chapter 5, the algorithms are combined and the system design as well as the use of the various techniques towards the overall system are described. In the following chapter the novel learning-based approach is discussed.

Chapter 4

Learning Based System

In this chapter the components that form part of the learning-based system are discussed. The basis of a learning-based approach is to extract features from an image and represent it as vectors, followed by training a model using these vectors and predicting a pose based on the trained model. The learning-based system consists of face detection, skin detection, background subtraction and morphological operations that collectively contribute to the feature extraction process. Face detection was discussed in the previous chapter and will therefore not be discussed here. In the skin detection process an attempt is made to identify a suitable colour space, based on the assumption from previous studies, that such a colour space exists. Simple background subtraction techniques are compared to probabilistic modelling techniques, in order to eliminate unnecessary objects that may interfere with feature extraction. This is followed by a description on morphological operations relevant to the system, so that features regarded as noise can be removed. The case of locating different body parts leads to a multi-class problem. Therefore, to train a model, SVMs are used to learn features. In the subsections that follow, SVMs, the use of kernel functions and the use of SVMs in multi-class problems are discussed.

4.1 Skin Detection

Skin detection is a process that identifies pixels in images or image sequences as either skin or non-skin pixels on the basis of pixel colour [17][67]. This process is of paramount importance in a number of applications and is especially useful in face detection, hand detection and even hand tracking in videos [59][167]. Skin colour information has become increasingly popular as it is not easily affected by partial occlusions, rotations or scaling of human body parts [26][69]. The idea behind detecting skin is that human

skin colours are distinct from the colours of most objects [111]. Detecting skin-coloured pixels, however, is a non-trivial task as colour pixels may vary due to factors such as viewing geometry, camera characteristics and changes in illumination.

When building a framework to detect skin, three basic steps are involved [69][164][111][37]:

1. A suitable colour space needs to be selected to represent the pixels in an image
2. Skin and non-skin pixels are modelled using an appropriate classification algorithm.
3. Pixels are individually classified as either skin or non-skin pixels.

A unique property to this learning-based system is that it differs from the conventional style of detecting skin by discarding the need for a classification algorithm and introduces a new framework to effectively detect skin. The new framework consists of three steps:

1. The face is located using the face detection algorithm, and the region around the nose is identified.
2. A suitable colour space needs to be selected to represent the pixels in an image.
3. Using the region around the nose, accurate skin colour information is extracted per individual and used to identify each pixel as either a skin or non-skin pixel.

Before further exploring the framework for skin detection, the colour spaces and colour transformations are explained. Colour spaces can be described as mathematically representing or storing colours in various ways [72]. Numerous colour spaces exist but many share similar characteristics. Therefore only the most widely proposed colour spaces for skin detection are explained.

4.1.1 RGB Colour Space

The RGB colour space is the default colour space for most image formats. It is an additive combination of red, green and blue pixel values. Any other colour space is simply obtained by performing a linear or non-linear transformation from the RGB colour space. This can be visualized as a 3D cube where red, green and blue form three perpendicular axes respectively. An advantage to using this colour space is its simplicity, however, it is not perceptually uniform. This means that distances in the colour space do not conform to a linear correspondence with human perception [37][164]. In addition to this, the red, green and blue channels are highly correlated, and the luminance and chrominance data are not separated. This makes it an unfavourable choice for colour-based recognition algorithms [37].

4.1.2 Normalized RGB Colour Space

In this colour space, the components are obtained by normalizing the red, green and blue pixel values by using a simple normalization formulation:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \tag{4.1}$$

and

$$r + g + b = 1 \tag{4.2}$$

where R, G, B are the red, green and blue pixel values and r, g, b are the normalized red, green and blue pixel values respectively. Since the sum of the normalized pixel values is 1, the third component can be omitted, as it does not hold any significant information and in effect reduces the space dimensionality [164][69]. In addition to this, the lighting effects are greatly reduced by performing the normalization [164][69][108].

4.1.3 TSL Colour Space

Applying a transformation on the normalized RGB colour space, results in a normalized chrominance-luminance colour space, known as the TSL colour space. It is similar to the HSV colour space and describes colour as Tint, Saturation and Lightness. The colour space can be formulated as follows (where Tint is a mixture of white with the dominant colour of an area):

$$T = \begin{cases} \frac{arctan(\frac{r'}{g'})}{2\pi} + \frac{1}{4}, & if\ g' > 0 \\ \frac{arctan(\frac{r'}{g'})}{2\pi} + \frac{3}{4}, & if\ g' < 0 \\ 0, & if\ g' = 0 \end{cases} \tag{4.3}$$

$$S = \sqrt{\frac{9(r'^2 + g'^2)}{5}} \tag{4.4}$$

$$L = 0.299R + 0.587G + 0.114B \tag{4.5}$$

where

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B} \tag{4.6}$$

and

$$r' = r - \frac{1}{3}, g' = g - \frac{1}{3} \tag{4.7}$$

where T, S, L represents the Tint, Saturation and Lightness pixel values. r' and g' represents a variant of the normalized red and green pixel values.

4.1.4 HSV Colour Space

The HSV colour space is a popular colour space for skin detection since it is based on the human colour perception [108][37][72]. It describes colour in terms of Hue, Saturation and Value (also known as Lightness or Intensity). Hue defines the dominant colour of an area, whereas Saturation measures the degree of the dominant colour of an area in proportion to its brightness. Value is related to the colour luminance thereby storing the brightness information. A mapping from the RGB colour space to HSV colour space is obtained via a non-linear transformation. This can be formulated as follows [82]:

$$v = max_{r,g,b} \tag{4.8}$$

$$s = \frac{max_{r,g,b} - min_{r,g,b}}{v} \tag{4.9}$$

$$h = \begin{cases} \frac{g-b}{6(max_{r,g,b}-min_{r,g,b})}, & if\ v = r \\ \frac{2-r+b}{6(max_{r,g,b}-min_{r,g,b})}, & if\ v = g \\ \frac{4-g+r}{6(max_{r,g,b}-min_{r,g,b})}, & if\ v = b \end{cases} \tag{4.10}$$

where h, s, v are the Hue, Saturation and Value pixel values and where $max_{r,g,b}$ and $min_{r,g,b}$ is the maximum and minimum between the red, green and blue pixel values respectively. An interesting property concerning the transformation from RGB to HSV is that the Hue component is invariant to high intensity at white light sources [164][69].

4.1.5 YCbCr Colour Space

The YCbCr colour space is often used in television media as well as various video compression purposes [52]. The colour space can be represented from the RGB colour space via a linear transformation. It defines colour as Y which is the luminance component computed by a weighted sum of the RGB pixel values. Cr and Cb, are the chrominance components computed by subtracting the chrominance component from the red and blue pixel values. This can be formulated as:

$$Y = 0.299R + 0.587G + 0.114B \tag{4.11}$$

$$Cr = R - Y \tag{4.12}$$

$$Cb = B - Y \tag{4.13}$$

where Y, Cr, Cb represent the luminance and chrominance components. This is another popular colour space for skin detection since it offers a simple transformation and explicitly separates the luminance and chrominance components. Furthermore, skin colours of different races are found to occur in the chrominance channels [37].It is therefore

possible to discard the Y component since the luminance is easily separable from the chrominance components.

4.1.6 A Suitable Colour Space for Skin Detection?

Many researchers primarily use colour space transformation to detect skin for the following reasons. First, it is assumed that a certain colour space transformation would increase the separation between skin and non-skin pixels. Second, it is assumed that invariance to illumination can be achieved. In order to ascertain whether these assumptions hold, studies have specifically been performed to investigate the effectiveness of colour space transformation for the purpose of skin detection [142][169][164][69]. These studies compared the colour space transformation with the default RGB colour space and shown that there is no significant performance improvements in the task of detecting skin, based on the above assumptions [142][169]. They suggest the colour space choice should depend on the skin detection methodology and application [69]. They also suggest that the colour space choice should depend on the format on which the image is obtained as well as the need for a specific colour space in post-processing steps [164][69]. Furthermore, they conclude that eliminating the intensity component in the colour spaces do not improve the discrimination of skin and non-skin pixels; however, they suggest that it may help to better generalize training data for a classification process [142][169].

The question on whether to use a colour space transformation is therefore left unanswered. Although many researchers choose a particular colour space, they can not justify their choice for that particular colour space. In addition to this, they can not justify that the chosen colour space is the most optimal colour space for skin detection.

Many researchers [37][29][30], however, agree with Forsyth and Fleck [41] that the Hue component in the HSV colour space has a restricted range on the human skin colour, which is formed by a combination of carotene, haemoglobin and melanin. Carotene has a distinctive yellow-orange colour and is mostly found in the palms and soles. Haemoglobin, which is the substance carrying oxygen in red blood cells, forms a pink-red colour in the skin. The primary determinant of skin colour is dependent on the amount and type of melanin found in the skin. There are two types of melanin, pheomelanin, which is a red colour and eumelanin, which is a very dark brown colour. A combination of these colours are easily distinguishable by the Hue component in the HSV colour space [37][29][30]. It is beyond the scope of this research to prove that there is an optimal colour space, but based on the research done, it is concluded that the Hue

component can be used to effectively determine skin colour amongst all races and skin colours [37][29][30].

4.1.7 Identifying Skin Pixels

In contrast to the standard framework for detecting skin pixels, the need for any machine learning algorithm is avoided. This is achieved by directly identifying the colour distribution of the skin pixels in the face, using only the Hue component of the HSV colour space and a colour histogram model. Using only the Hue component is an efficient way to identify skin pixels since only a one dimensional space is used and therefore requires less CPU instructions per pixel. The simplest way to identify a skin region in a colour space, is to specify a range within the space, using thresholds.

Most researchers implementing skin detection using thresholds make use of constant thresholds [108][118][7] and therefore fail to identify skin pixels amongst all skin colours. According to scientific studies, the skin colour diversity in South Africa as well as the sub-Saharan African populations, is the highest in the world [124]. It is therefore necessary to be able to identify skin pixels amongst all races and skin colours. To differ from other studies, the threshold values are changed in an adaptive manner according to the particular individual. This is achieved by locating the face using the face detection algorithm. When using the entire face to create a colour distribution of the skin region, it is negatively affected by facial hair, eyes, lips or spectacles. Therefore only the region around the nose is used to create a colour distribution of the skin pixels, with a radius of 10 pixels from the centre of the face. From experimentation based on trial and error, this radius is sufficient to determine the distribution. A colour histogram is implemented using the Hue component to generate the colour distribution. The colour distribution is indicated by a colour bar where the height of the bar indicates the number of pixels in an image that have that Hue colour, as shown in Figure 4.1.

FIGURE 4.1: The colour distribution of the area around the nose

In Figure 4.1 the red bar indicates the non-skin colour distribution in the region, while the orange bar indicates the skin colour distribution in the region. A filter is created according to the threshold values determined from the histogram and can be described as follows:

A pixel is determined to be a skin pixel if,

$$T_L(H) \leq Image(i,j) \leq T_U(H) \tag{4.14}$$

where T_L is the lower threshold, T_U the upper threshold, H the Hue histogram and $Image(i,j)$ the Hue pixel. If condition 4.14 is not satisfied then the pixel is determined to be a non-skin pixel. Using this filter, a value of 255, which is white, is given to skin pixels whereas a value of 0, which is black, is given to non-skin pixels, as illustrated in Figure 4.2.

FIGURE 4.2: Pixels considered to be skin colour are white, while pixels considered not to be a skin colour are black, with respect to the filter. The area circled in red is used to create the colour distribution.

A challenge that remains even though the Hue range can be found, is that many false detections from the background may occur if a controlled environment is not used. For instance furniture, leather and clothing resembling skin colour are falsely detected as skin, as it may possess the same hue range as the particular individual. To resolve this challenge, an adaptive background subtraction technique is used to eliminate such false detections. This technique is further explained in the following section.

4.2 Background Subtraction

Background is defined as a scene in which objects stay constant, whereas foreground contains objects of interest that frequently move within the scene [79]. Background subtraction is a process that involves separating the background and foreground from

a sequence of images in order to highlight objects of interest in the scene. In this application, the objects of interest are the limb movements of an individual performing sign language. To effectively and reliably obtain the objects of interest, the background subtraction algorithm should handle the following conditions [136]:

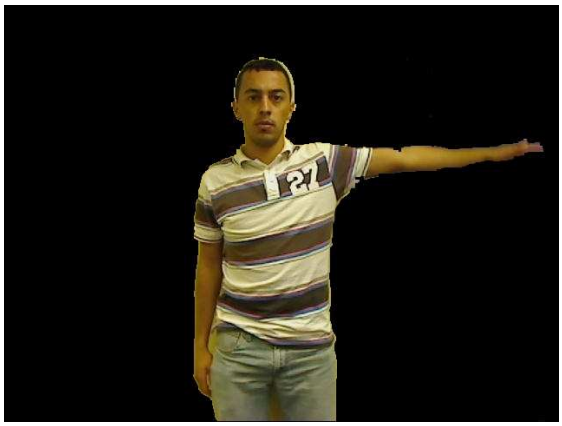
- Any sudden or gradual changes in illumination
- Avoid detecting *background* objects with high movement frequency such as moving tree leaves, rain or snow
- A quick reaction to changes in the scene

There exists a number of background subtraction algorithms varying from simple techniques to more advanced probabilistic modelling techniques. In the subsections below, only the dominant algorithms in this area are discussed.

4.2.1 Simple Techniques

The simplest form of background subtraction is to use a static reference image. A reference image is the image to which the current image will be compared. This comparison process can be described in terms of a binary classification problem where each pixel is assigned a label belonging to the background or foreground class [34]. To represent this formally, consider a pixel p in image $I(i, j)$. For each pixel p, it is assigned a label p_l where $l \epsilon$ {background, foreground}. Using this mask, background pixels are either set to black or white (in this case black) in order to highlight the object of interest which is the foreground object. This form of background subtraction is shown in Figure 4.3:

FIGURE 4.3: Simple background subtraction applied to an image



From Figure 4.3, it should be noted that although the object of interest is the particular individual, the clothing worn may negatively affect the skin detection process. For an

accurate skin detection process, false detections should be reduced as much as possible. To do so, all objects that may contribute to the false detections should be removed. A more effective way, to separate the background in cases such as these, is to use frame differencing which is another simple background subtraction technique.

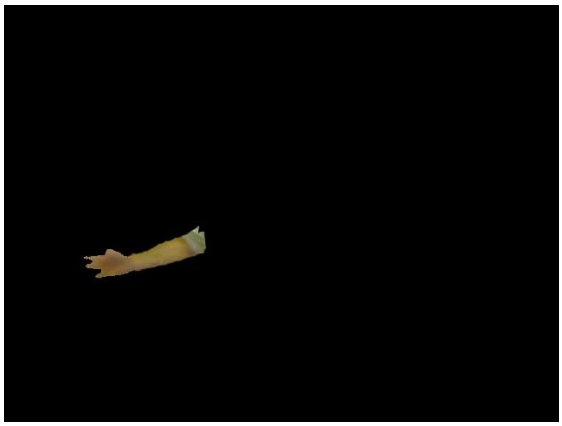
Frame differencing is similar to the static background subtraction technique but differs by continuously updating the reference image, which in effect is an updated background image. This image is either the previous frame in an image sequence or possibly several frames previously. The choice here depends on the object of interest's speed and the frame rate of the image sequence, as this method is greatly dependent on these two conditions [119]. A pixel is labelled as foreground if the difference between the pixel in the reference image and the corresponding pixel in the current image is above a certain threshold. Given two images, the frame difference can be computed by the following equation:

$$|I(i,j) - Ref(i,j)| > Th \tag{4.15}$$

where $I(i,j)$ is the current image, $Ref(i,j)$ is the reference image and Th is the threshold value.

Due to the nature of this application, it is important that information is not missed. It is for this reason that the reference frame is updated with the previous frame and that a low threshold is chosen, so that movement, when performing sign language, can be highlighted as seen in Figure 4.4:

FIGURE 4.4: Adaptive background subtraction applied to an image



From Figure 4.4, it is observed that the movement of the arms and its immediate surrounding, are regarded as foreground objects. This technique suits this application, since most objects that may negatively affect the skin detection process are removed, such as the clothes worn. It also highlights the area considered to be the object of interest.

4.2.2 Probabilistic Modeling Techniques

The adaptive GMM is a more advanced background subtraction technique whereby the values of background pixels are modelled as a mixture of adaptive Gaussians. For this technique, a mixture of adaptive Gaussians are used since multiple surfaces may appear in a pixel and lighting conditions may change [136][149]. The history of a pixel (i, j) at any time t can be formulated as [149]:

$$\{I_1, ..., I_t\} = \{I(i, j, x) : 1 \leq x \leq t\} \tag{4.16}$$

Given k Gaussian distributions, each pixel can be modelled by a mixture of these distributions. To evaluate the probability that a pixel may have a value I_t at time t, the following formula can be used [149]:

$$P(I_t) = \sum_{x=1}^{k} W_{x,t}\eta(I_t; \mu_{x,t}, \Sigma_{x,t}) \tag{4.17}$$

where $W_{k,t}$ is the estimated weight parameter of the k^{th} Gaussian component and $\eta(I, \mu_{k,t}, \Sigma_{k,t})$ is the normal distribution of the k^{th} Gaussian component represented by [149]:

$$\eta(I, \mu_{k,t}, \Sigma_{k,t}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_{k,t}|^{\frac{1}{2}}} e^{-\frac{1}{2}(I_t-\mu_{k,t})^T\Sigma_{k,t}^{-1}(I_t-\mu_{k,t})} \tag{4.18}$$

where $\mu_{k,t}$ is the mean and $\Sigma_{k,t} = \sigma_{k,t}^2 I$ is the covariance of the k^{th} Gaussian component. The number of distributions, k, are ordered based on the fitness value $\frac{W_{x,t}}{\sigma_{x,t}}$, and the background of the scene is modelled using the first M distributions where M is estimated as:

$$M = argmin_M(\sum_{x}^{M} W_{x,t} > Th) \tag{4.19}$$

where Th is the threshold, which is the minimum fraction of the background model. After the background has been updated, the foreground is detected by labelling any pixel found to be more that 2.5 standard deviations away from any one of the M distributions. If the test value matches the k^{th} Gaussian component, then it is updated as follows [93]:

$$\begin{aligned} W_{x,t} &= W_{x,t-1} \\ \mu_{x,t} &= (1-p)\mu_{x,t-1} + pI_t \\ \sigma_{x,t}^2 &= (1-p)\sigma_{x,t-1}^2 + p(I_t - \mu_{x,t})^T(I_t - \mu_{x,t}) \\ p &= \alpha P(I_t|\mu_{x,t-1}, \Sigma_{x,t-1}) \end{aligned} \tag{4.20}$$

where $w_{x,t}$ is the k^{th} Gaussian component and $\frac{1}{\alpha}$ is defined as the time constant that determines change. If the Gaussian component does not match the test value, then it is

updated as follows [93]:

$$\begin{aligned} W_{x,t} &= (1-\alpha)W_{x,t-1} \\ \mu_{x,t} &= \mu_{x,t-1} \\ \sigma^2_{x,t} &= \sigma^2_{x,t-1} \end{aligned} \tag{4.21}$$

If none of the components match the test value, then the least probable component is replaced by a new one with a low weight parameter, a high variance and the current value as its mean. After evaluating the Gaussian distributions, pixels that do not match, are classified as foreground and grouped using 2D connected component analysis.

4.2.3 Comparing the Background Subtraction Techniques

Each technique has its relative strengths and weaknesses. The selected technique depends on its usefulness and effectiveness towards a particular application. Not many work has been done on doing a thorough comparison, however, in Table 4.1, the strengths and weaknesses of these techniques are stated [119][149].

In this comparison the accuracy can not be compared since an unbiased comparison with a benchmark on these techniques is still required. Adaptive GMMs require a number of frames, to create the background model during initialization. It has intermediate performance and fails when light suddenly changes. These disadvantages are key factors required in this application. This technique is therefore not suitable for this application. On the other hand, although a simple technique, frame differencing has fast performance and does not fail when light suddenly changes, since the frame rate is the standard 25 frames per second (fps) and the previous frame in an image sequence is the updated background model.

Experiments show that a low threshold works well in this application, with an individual performing sign language at 25 fps. This technique removes the majority of objects that may falsely be detected as skin and highlights only the object of interest, which in this application, is the hands and arms. Any noise caused by objects with high movement frequency such as tree leaves are further removed by the skin detection process (A systematic integration of the techniques is discussed in Chapter 5). Furthermore, any slow moving objects in a scene, that are not part of the objects of interest, are removed using this process. In addition, any small noise in the highlighted regions of interest (ROI) are removed and *holes* in large areas of interest are filled using a morphological process, which will be discussed in the following section.

TABLE 4.1: The strengths and weakness of the background subtraction techniques [119]

Techniques	Strengths	Weakness
Simple Background Subtraction	• Simplicity • Fast performance • Background models are not constant as they change over time (excluding the static background subtraction)	• High memory requirements for average and median techniques (average and median techniques not discussed here) • Accuracy depends on the objects of interest's speed and the frame rate • Affected by slower moving objects in a scene
Probabilistic Modeling	• A different threshold is selected for each pixel and adapts with respect to time • Fast recovery for background models • Additional objects can form part of the background without destroying the existing background model	• Intermediate performance and memory requirements • Fails when light suddenly or drastically changes • Fails with improper initialization of the Gaussians • Accuracy affected when parameters are improperly selected

4.3 Morphological Operations

The adaptive background subtraction technique is an effective way to identify regions where movement has taken place; however, some noise or *holes* still remain due to inconsistent lighting. The noise, especially in large amounts, may affect the ability of the SVM to generalize well and it is therefore desirable to remove these abnormalities in binary images [116][80]. To accomplish this, morphological operations are used on binary images or greyscale images. The morphological operations referred to in this thesis are based on mathematical morphology, that is a non-linear approach based on set theory and the geometrical properties of images [85].

Morphological operations require both a binary image and a structuring element as input. The structuring element is an image processing element that determines the

effects on an image. It is generally 3x3 in size and specified by a pattern of elements relative to an origin at the centre pixel [116], as seen in Figure 4.5:

FIGURE 4.5: An example of a 3x3 structuring element [116]

0	1	0
1	1	1
0	1	0

Similar to a mask, the structuring element moves over an image and compares its elements with pixel values in the image, relative to the origin. There are two basic morphological operations, namely, erosion and dilation. Two additional morphological operations, namely, opening and closing, are derived from the fundamental properties of erosion and dilation. Further information on other morphological operations such as thinning, thickening, medial axis transform and, hit and miss transform can be found in [40]. In the subsequent sections erosion, dilation, opening and closing operations, which are relevant to this work, are discussed.

4.3.1 Erosion

Erosion is a process that erodes away the boundaries of image regions using a structuring element, thereby removing noise [116]. This causes image regions to shrink and *holes* within image regions to grow. The structuring element is superimposed on each pixel with its centre aligned with the current pixel. If each pixel in the structuring element corresponds to a foreground pixel, then the current pixel remains a foreground pixel, otherwise, the current pixel is set to a background pixel. Erosion of a binary image, the set A, by a structuring element, the set B, can be defined as [84]:

$$A \ominus B = \{x|(B)_x \subseteq A\} \tag{4.22}$$

where B_x is the set B translated by the vector x. Erosion on a greyscaled image causes dark regions in an image to expand and light regions to either shrink or be removed. This effect occurs mostly at image regions where there are rapid changes in intensity [162]. The dual operation of erosion is dilation, which means that applying erosion to a binary image is equivalent to applying dilation to the inverse of that image.

4.3.2 Dilation

Dilation is a process that expands foreground pixels in image regions using a structuring element, thereby causing image regions to grow and *holes* within image regions to shrink

[116]. Similar to the erosion process, the structuring element is superimposed on each pixel with the origin of the structuring element aligned with the current pixel. If every pixel in the structuring element corresponds to a background pixel, then the current pixel remains a background pixel, otherwise, the current pixel is set to a foreground pixel. Dilation of a binary image, the set A, by a structuring element, the set B, can be defined as [84]:

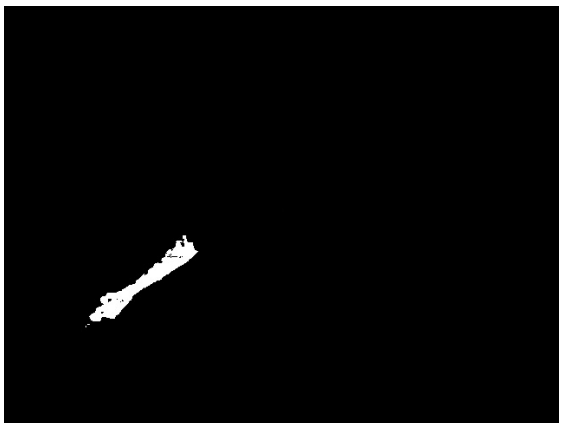
$$A \oplus B = \cup_{b \epsilon B} A_b = \{x | (B^s)_x \cap A \neq \emptyset\} \tag{4.23}$$

where B^s denotes the reflection of the set B and $(B^s)_x$ is B^s translated by the vector x. Dilation on a greyscaled image causes dark regions in an image to either shrink or be removed and causes light regions in an image to expand. Similar to erosion, this effect occurs mostly at image regions where there are rapid changes in intensity [162]. Dilation is often used to fill image regions, thereby enhancing the features. Using erosion and dilation operations in turn gives rise to two additional morphological operations, namely, opening and closing arithmetic operations.

4.3.3 Opening

Opening is a process that involves erosion followed by dilation using the same structuring element for both operations [54]. This operation removes the fine noise, smoothes the contours of objects and enhances the features in an image, illustrated in Figure 4.6:

FIGURE 4.6: Opening operator applied to a binary image to reduce noise



The opening operation of a binary image, the set A, by a structuring element, the set B, can be defined as [54]:

$$A \circ B = (A \ominus B) \oplus B \tag{4.24}$$

Using the opening operation on a greyscale image, leave regions larger than the structuring element unchanged while reducing the brightness of smaller regions [162]. This operation retains regions similar to the structuring element and is therefore useful for

segmentation. It is particularly useful in this system since it is able to remove noise while filling the *holes*, thereby enhancing the features in order to aid generalization by the SVM.

4.3.4 Closing

Closing is a process that involves dilation followed by erosion using the same structuring element for both operations [54]. This operation tends to fill *holes* in image regions but counters the effects thereof due to the erosion operation, if the entire image region or object is not filled. The closing operation of a binary image, the set A, by structuring element, the set B, can be defined as [54]:

$$A \bullet B = (A \oplus B) \ominus B \tag{4.25}$$

Using the closing operation on a greyscale image leave regions larger than the structuring element unchanged while increasing the brightness of smaller regions [162]. This operation may not prove useful in this application, as it increases the noise and negatively affects the features in an image.

An important property concerning the opening and closing operation is that both operations are idempotent. This means that each operation should only be performed once with the same structuring element since repeated operations results in the same output image and therefore contributes to unnecessary computation time [80][162].

4.4 Support Vector Machines

Past and present research have shown that there continues to be a growing interest in solving pattern recognition problems using Support Vector Machines (SVM). SVMs are derived from statistical learning theory and is a machine learning tool that initially classified data into two classes, but has been extended to support classification of multi-classes. In comparison to other classifiers, SVMs offer several advantages [166]. One such advantage is that training time is not affected by high dimensional feature vectors originating from large images. Another advantage is its use of kernel functions that offer the classifier both power and flexibility. This is achieved by substituting the linear kernel, which is the default kernel, with a radial basis function, polynomial, sigmoid or other more recent kernels such that the data points in a given classification problem may be separated more cleanly. Moreover, it allows SVMs to use linear classification techniques to solve non-linear classification problems.

In principle, SVMs are mathematical algorithms that aim to maximize a mathematical function, given a collection of data points [113]. Consider a set of data points consisting of two classes; it is possible to find a boundary that separates those two classes. Furthermore, consider a set of M training data points represented by $S = \{(x_1, y_1), (x_2, y_2), ..., (x_M, y_M)\}$ where each x_i, with $i = 1, 2, ..., M$, is a data point in $\mathbb{R}^n$ and each $y_i \epsilon \{\pm 1\}$ is the corresponding classification label which divide the data points into a positive and a negative class. Suppose also that the two classes $S^+ = \{x_i | y_i = 1\}$ and $S^- = \{x_i | y_i = -1\}$ are linearly separable in $\mathbb{R}^n$ such that at least one boundary can be formed between them [2].

This boundary which separates the two classes by a straight line, also referred to as the decision boundary, can be found by training an SVM [2], Figure 4.7(a).

FIGURE 4.7: (a)Linear classification (b) Non-linear classification [2]

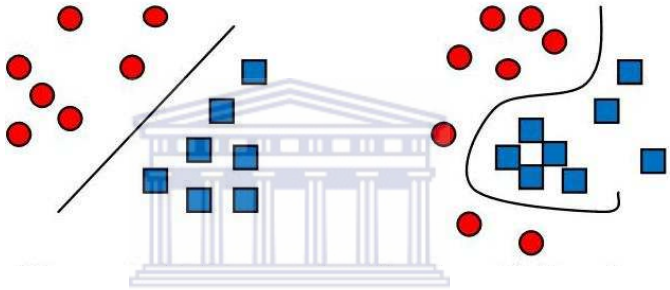
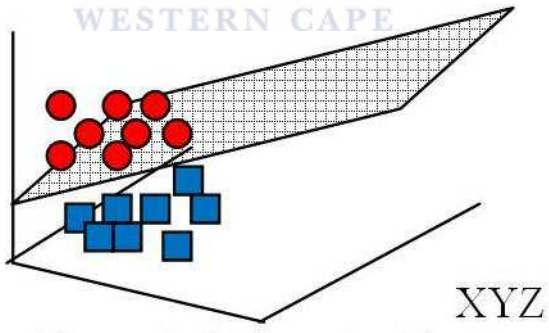


FIGURE 4.8: Linear classification of a plane [2]



In a higher-dimensional space, this is a geometrical concept of a plane, illustrated in Figure 4.8. It is generally referred to as a hyperplane and defined by the following equation:

$$f(x) = \mathbf{w} \cdot \mathbf{x} + b = 0; \ \mathbf{w} \epsilon \mathbb{R}^n, \ b \epsilon \mathbb{R} \tag{4.26}$$

where $\mathbf{w}$ is the normal vector and b is the interim term. Vector $\mathbf{w}$ of the decision hyperplane is defined as a linear combination of x_i with weights α_i as follows:

$$\mathbf{w} = \sum_{1 \leq i \leq N} \alpha_{\alpha i} x_i y_i \tag{4.27}$$

A simple rescale of $\mathbf{w}$ for all the points x_i (support vectors) lying on the respective hyperplanes holds that:

$$\begin{aligned} \mathbf{w} \cdot x_i + b &= 1 \\ \mathbf{w} \cdot x_i + b &= -1 \end{aligned} \tag{4.28}$$

d represents the distance between the decision boundary and the margin can thus be expressed as:

$$d = \frac{2}{||\mathbf{w}||} \tag{4.29}$$

The selection of the hyperplane is based on two factors. First, it should separate the data points clearly and second, it should have the maximum distance to the nearest data point from both classes. This distance is also referred to as the margin and the support vectors are the data points that are situated closest to the hyperplane. When there is a greater separation between the two classes, it is necessary to find the maximum margin as this allows an SVM to classify a new data point more accurately. To find such a hyperplane, the following conditions need to be met; the first condition is that all training data points should be classified correctly [2]. Hence $\mathbf{w}$ and b are to be estimated such that:

$$y_i(\mathbf{w} \cdot x_i + b) \geq 1 \; for \; y_i = 1 \tag{4.30}$$

and

$$y_i(\mathbf{w} \cdot x_i + b) \leq -1 \; for \; y_i = -1 \tag{4.31}$$

These two equations can be combined to give:

$$y_i(\mathbf{w} \cdot x_i + b) - 1 \geq 0, \; \forall \; i = 0, 1, 2, ..., N \tag{4.32}$$

The second condition is that the margins should be as large as possible. Maximizing equation 4.29 is the same as minimizing $\frac{||\mathbf{w}||}{2}$. Therefore, $f(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2$ should be minimized. Following this, the optimal hyperplane can be found by solving the optimization problem defined as:

$$\begin{aligned} &Minimize \; \frac{1}{2}||\mathbf{w}||^2 \\ &Subject \; to \; y_i(\mathbf{w} \cdot x_i + b) - 1 \geq 0, \; \forall \; i = 0, 1, 2, ..., N \end{aligned} \tag{4.33}$$

This problem can be solved, given the Lagrange multipliers $\alpha_1, \alpha_2, ..., \alpha_N \geq 0$ and the saddle point of the Lagrange function:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i(y_i(\mathbf{w} \cdot x_i + b) - 1) \tag{4.34}$$

Therefore, using the Lagrange function, the optimization problem can be translated to:

$$\begin{aligned} Maximize \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i, x_j) \\ Subject\ to \sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, i = 1, 2, ..., N \end{aligned} \tag{4.35}$$

The hyperplane that is selected is referred to as the optimal hyperplane or the maximum-margin hyperplane. The optimal hyperplane discriminant function, under this formulation is thus:

$$f(x) = \sum_{i \epsilon S} \alpha_i y_i (x_i x) + b \tag{4.36}$$

where S is the subset of support vectors corresponding to positive Lagrange multipliers. To summarize, SVMs use an optimal hyperplane to seperate data points by learning a decision boundary.

When classifying linear problems, a linear hyperplane with a maximum margin that separates the data points is found. In non-linear problems, a slightly different classification approach is used. The data points are mapped onto a higher-dimensional space known as the feature space, where an optimal hyperplane that separates the data points linearly can be found.

In reality, problems involving non-linear cases require more complex structures to find a decision hyperplane. In such cases, illustrated in Figure 4.7(b), the data points are unevenly distributed and non-separable compared to those in Figure 4.7(a).

In these cases, the classes are not linearly separable and the constrain of equation 4.32 cannot be satisfied. To solve such cases, a cost function that combines the margin maximization and the minimization of error criteria can be formulated. This is achieved by using a set of variables, ξ also known as slack variables. Hence the cost function can be expressed as:

$$\begin{aligned} Minimize_{\mathbf{w},b,\xi} \frac{1}{2} ||\mathbf{w}||^2 + C \cdot \Sigma_{i=1}^{N} \xi_i \\ Subject\ to\ y_i(\mathbf{w} \cdot x_i + b) \geq 1 - \xi_i \\ where\ \xi_i \geq 0 \text{ and } C \text{ are constants} \end{aligned} \tag{4.37}$$

The parameter C determines the trade-off between the amount of error to be tolerated and the margin maximization.

According to Mercer's theorem [159], in the mapping space, the dot product of the vectors can be equally formed as a function of dot products of the corresponding vectors

in the current space [161]. This equivalence can be expressed as:

$$\begin{aligned} K(x_i, x_j) &= \phi(x_i) \cdot \phi(x_j) \\ &= (x_i, x_i^2) \cdot (x_j, x_j^2) \\ &= x_i x_j + x_i^2 x_j^2 \\ &= x_i \cdot x_j + (x_i, x_j)^2 \end{aligned} \tag{4.38}$$

where the kernel function is represented by $K(x_i, x_j)$. This expression is true if and only if the following condition holds true for any function g:

$$\int g(\mathbf{x})^2 d\mathbf{x} \text{ is finite } \Rightarrow \int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \tag{4.39}$$

Without knowing the explicit form of ϕ, any data can be linearly separated in the higher-dimensional space by simply selecting an appropriate kernel function. Thus, the dual optimization problem can be defined as:

$$\begin{aligned} &Maximize \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ &Subject\ to \sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, \text{ where } i = 1, 2, ..., N \end{aligned} \tag{4.40}$$

It should be noted that drawing a complex curve is not suitable to separate data. As an alternative, it is possible to find an optimal hyperplane in the feature space that separates the data clearly and allow an SVM to accurately classify new test data. The decision function therefore becomes:

$$f(x) = \sum_{i \epsilon S} \alpha_i y_i K(x_i x) + b \tag{4.41}$$

where S are the support vectors.

4.4.1 Kernel Functions

Often in non-linear cases, a suitable hyperplane that separates the classes is required. To achieve this, a kernel function is used to map the data from the current space onto a higher-dimensional feature space. Following Mercer's theorem, four basic kernels are used by the SVM for training and classification, where r, d and γ are kernel parameters [57]:

- Linear: $K(x_i, x_j) = (x_i)^T \cdot (x_j)$

- Polynomial: $K(x_i, x_j) = (\gamma(x_i)^T \cdot (x_j) + r)^d$, where $\gamma > 0$
- Radial Basis Function: $K(x_i, x_j) = exp(-\gamma \cdot ||x_i - x_j||^2)$, where $\gamma > 0$
- Sigmoid: $K(x_i, x_j) = \tanh(\gamma \cdot (x_i)^T \cdot (x_j) + r)$, where $\gamma > 0$

Choosing a kernel is important as it influences the prediction capabilities of the SVMs [51][57]. Over the past few years research has been done to help choose an appropriate kernel for a given problem, given a specific set of features [83]. However, no standard method exists to find the most appropriate kernel [168]. Thus, selecting an appropriate kernel is often based on a trial and error procedure [27].

4.4.2 Comparing Multi-Class SVM Techniques

Although SVMs are inherently binary classifiers that handle 2-class problems, it can be applied to problems that require more than 2-classes, using a variety of techniques. A comprehensive study on these techniques are presented in [58], however, only three of the most common are explored. In general, multi-class problems are often solved using a combination of binary classifiers and a decision strategy to determine the class to which the input pattern belongs [137].

4.4.2.1 One-vs-Rest

In this technique, each SVM separates the data points of class i from the data points of the remaining classes in an M-class problem, for every $i = 1, ..., M$. Here, apart from class i, the data points from the remaining classes are combined to form a single class. Thus, M classifiers, in total are trained. In the testing phase, a test pattern is presented to all M classifiers and the class i with the maximum output value is determined as its label. Due to the large number of data points in each combination pair of classes, this technique results in slow training and testing times.

4.4.2.2 One-vs-One

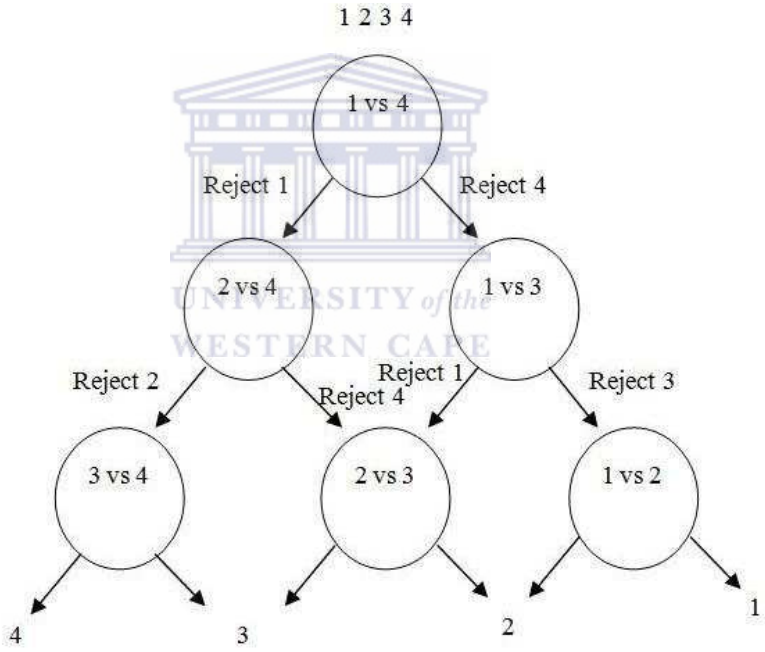
In this technique, $\frac{M(M-1)}{2}$ binary classifiers are trained using every binary pair-wise combination of the M classes, i.e. a classifier for each distinct pair (u, v) where $u \neq v$. Thus, every classifier is trained to differentiate between the two classes, using the data points in class u and v as positive and negative examples, respectively. Furthermore, to combine the classifiers, the Max-Wins algorithm [42] is used. In the testing phase, the algorithm is used to determine the class by selecting the class with majority of

the votes, voted by the classifiers [88]. Since the number of data points for each pair-wise combination of classes is smaller compared to the combinations in the one-vs-rest technique, it results in faster training times. Due to the large number of classifiers, especially if M is large, it results in slower testing times.

4.4.2.3 Directed Acyclic Graph SVM

The Directed Acyclic Graph (DAG) SVM algorithm was initially introduced by Platt [120]. In the training phase, similarly to the one-vs-one technique, $\frac{M(M-1)}{2}$ binary classifiers are trained using every binary pair-wise of the M classes. In the testing phase, the decision strategy is based upon a rooted binary DAG that consist of $\frac{M(M-1)}{2}$ internal nodes and M leaves, as illustrated in Figure 4.9:

FIGURE 4.9: At each node a class will be rejected until a single class remains [137]



Given a test pattern, beginning at the root node, where two classes (A and B) exist; if the pattern is classified as class A, then it does not mean that class A was selected but rather means that class B was rejected. Thus, from this node onwards, it will not be necessary to classify against class B again. Therefore, after classifying each consecutive node, a class in the node will be rejected. Hence, after $M-1$ steps, a single class which is the predicted class will remain. Using this technique, therefore results in faster training times compared to the one-vs-rest technique and faster testing times compared to the one-vs-one technique. This technique is therefore a suitable technique for multi-class SVMs.

4.5 Summary

In the preceding sections the components that make up the learning-based system are investigated. An attempt is made to find a suitable colour space for the skin detection process. Based on the research done, there is no optimal colour space for skin detection. In addition to this, research also shows that the intensity component does not improve the discrimination of the skin pixels in a specific colour space. Furthermore, it is agreed upon by many researchers that the Hue component in the HSV colour space has a restricted range on the human skin colour. In addition, a colour histogram is used to find the optimal range of the Hue component with respect to the skin colour of the individual, using only the region around the nose to determine the colour distribution.

Even though the Hue range can be found, false detections from objects in the background may occur if a controlled environment is not used. To eliminate such detections, a variant of the simple background subtraction technique is used to only determine regions where movement has occurred. This is achieved by continuously updating the background model with the previous frame. Due to any noise retained in the images and regions of interest in an image containing *holes*, morphological operations are used before placing the features in a vector. The opening morphological operation is suitable for this application since it is able to remove noise while filling the *holes*, thereby enhancing the features in order to aid generalization by the SVM. It should be noted that repeating the opening or closing morphological operation results in the same output image and therefore unnecessary computation time.

To learn these features, SVMs are used and since the problem is a multi-class problem, it provides a perfect solution. Furthermore, there is no standardised method to help choose an optimal kernel and is often based on a trial and error procedure. In Chapter 6, different kernel functions are evaluated with the aim of finding a suitable one.

In the following chapter the algorithms are combined and both the example-based and learning-based system are described as well as the use of the various techniques towards the overall systems.

Chapter 5

Systems Implementation and Design

This chapter focuses on the systematic design and procedures in which the systems are implemented and used. The algorithms discussed in Chapter 3 and Chapter 4 are combined in the respective systems and the links between the algorithms are shown. Each system will be individually described. Some algorithms are prerequisites for others and it is therefore important that the system design be emphasised. For the learning-based system, LibSVM [22] is used as it provides a simple and effective means to train and predict data using SVMs. Both systems consist of two separate phases described in the subsequent sections.

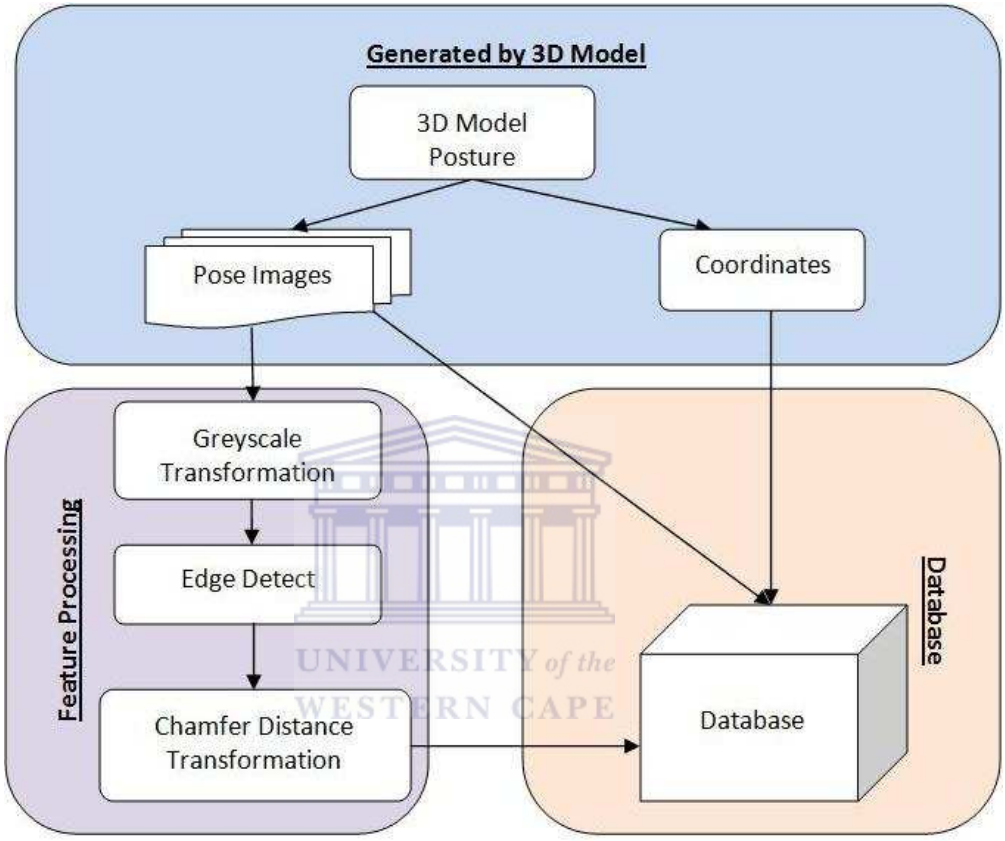
5.1 Example-Based System Design

The objective of an example-based approach is to encode features from image observations that are used to identify poses from examples in a database [121]. Before extracting these features from images, image registration is required. Following the feature extraction methods, the features are used to find a pose from the database that either matches or closely resembles the pose, to the extent that such a pose can be identified as the pose in the test image. To give an overview of the system, a systematic design is provided. The overall implementation of the system consists of two separate phases: a phase where the database is constructed and a phase where the system is tested. As the design of the two phases are different, they will be explained separately in the following sections.

5.1.1 Setting-up the Database

This section describes the construction of the database. This phase takes place iteratively as illustrated in Figure 5.1.

FIGURE 5.1: Example-based system design and implementation for setting-up the database.



Similar to Shaknarovich et. al. [138] and Poppe and Poel [122] that used POSER and Cao et. al [20] that used an alternative 3D model, this system uses a 3D model using Blender to generate a large number of sample images. Each generated image represents a human upper body posture used in sign language containing the most important joints such as the wrists, elbows and shoulders. In this system 12 660 images were generated and each image in its default colour space, RGB, is transformed to the greyscale colour space using transformation equation 3.4 and illustrated in Figure 3.5.

The greyscale image is a preparation step for the edge detection method since it highlights the intensity of an image in a one dimensional colour space. Using the Canny operator, a 3x3 mask is applied to each 3x3 region on the image. It begins in the top left of the image, moves from left to right and top to bottom. Following this path, the mask identifies regions containing rapid changes in intensity. These changes in intensity indicate the existence of an edge in the image. Pixels considered to be an edge are

given a value of 255 – white – whereas pixels considered not to be an edge are given a value of 0 – black. Given the edge detected image, a distance metric can be used to transform the image into a representative image that indicates the distance of a pixel to the nearest edge pixel. Similar to Cao et. al. [20] and Micilotta et. al. [94], the Chamfer Distance Transformation is also used. This is a suitable technique to use in this system as it provides a closer approximation to the Euclidean distance as compared to other approximate distance transforms. Using a 5x5 mask with a (5:7:11) approximation, the Chamfer Distance Transformation can be applied to an edge detected image.

Each edge image is scanned by proceeding with a forward scan, which begins in the upper left corner of the image. The forward scan moves from top to bottom and from left to right. The second scan is the backward scan, which begins in the bottom right corner of the image. The backward scan, in contrast to the forward scan, moves from bottom to top and from right to left. Throughout the scanning procedure, a mask constant is added to the pixel values and the minimum distance in the (5:7:11) neighbourhood is assigned to the centre pixel. Following this procedure, the local distances to an edge are represented by the pixel values. Thus, pixels closer to an edge will consist of a lower pixel value and pixels further from an edge will consist of a higher pixel value [2]. Consequently, each distance transformed image corresponds to an approximation of the distance from a pixel to the closest edge pixel.

The distance transformed images is stored in a folder on the computer and only the file locations are stored in the database. Additionally, the file locations of the example images and the 3D coordinates corresponding to the shoulders, elbows and wrists of each image are stored in the database, along with the file locations of the distance transformed images. Each entry in the database is given a key id for an easier reference and a *calculated_distance* entry initially set to null, illustrated in Figure 5.2.

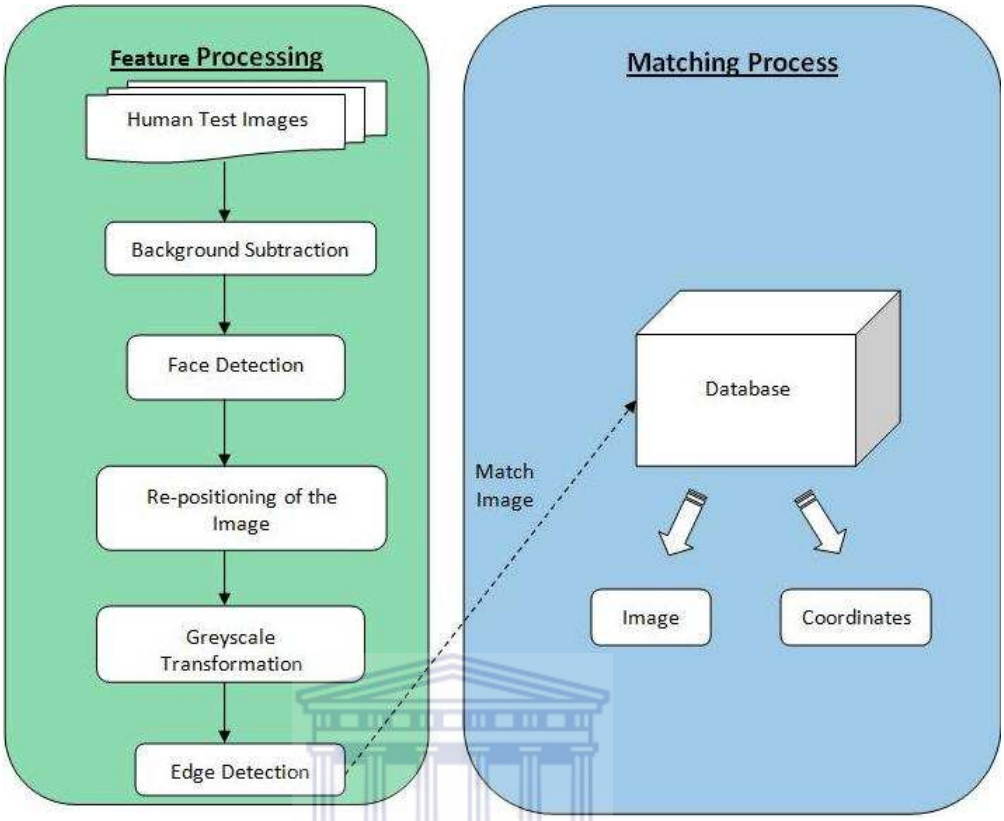
FIGURE 5.2: An overview of the database used in the example-based system

id	Original_Image	Chamfer_Distance Transformed Image	Wrists						Elbows						Shoulders						Calculated Distance
			Left			Right			Left			Right			Left			Right			
			x	y	z	x	y	z	x	y	z	x	y	z	x	y	z	x	y	z	
1	Original_1.jpg	Distance_1.jpg	6.4	2.5	0.1	4.1	1.1	0.5	1.1	1.5	0.6	0.7	0.8	0.2	0.03	0.2	0.0	0.04	0.2	0.0	260
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

5.1.2 Testing Setup

This section describes the procedure in evaluating the system from a given test set, using the database constructed in the previous phase. This phase takes place iteratively as illustrated in Figure 5.3.

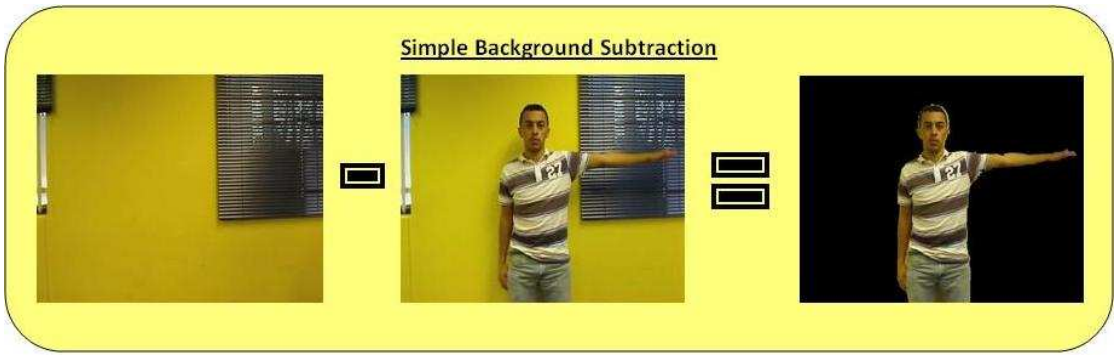
FIGURE 5.3: Example-based system design and implementation for testing.



The phase operates differently to the previous phase since test images may contain objects in the background that could affect the matching process. It is therefore necessary to isolate the person from the background so that only the person is used in the matching process. To separate the person from the background, a simple background subtraction technique that makes use of a static reference image is used. The reference image, which forms the background, is the image to which the current image will be compared. To obtain the reference image, a pre-recorded frame is used before the person enters the scene. The background subtraction is performed by subtracting each pixel in the reference image with the corresponding pixel in the current image. This operation highlights the region considered to be the foreground and sets all pixels considered to be the background as black, illustrated in Figure 5.4.

To further aid the matching process, it is desirable that each person be at a consistent location in the frame so that an overlap occurs with the template image. Each person may stand at different locations. Some people may stand more to the left while others may stand more to the right. Furthermore, each person may not be the same height, some may be short while others may be tall. To address this challenge and ensure that each person is at a consistent location in the image, the image is normalised so that consistency is achieved. The normalisation process is carried out as follows. After

FIGURE 5.4: The process involved using the simple background subtraction technique. A static reference image is subtracted from the current image to obtain the background subtracted image.



subtracting the background, a face detection method is used to detect the face in the image. By dividing the height and width of the face by two, the x and y coordinate of the centre of the face is obtained. To find a consistent point to which all other images should be re-positioned, such that the person is on a common point in the frame, let the centre pixel coordinate of the 3D model's face be (m, n) and let the centre pixel coordinate of the person's face in the test image be (x, y). Also, let the number of pixels by which to move the image in the x-dimension and y-dimension be X and Y respectively. This number of pixels can be calculated using the following equation:

$$X = m - x \tag{5.1}$$

$$Y = n - y \tag{5.2}$$

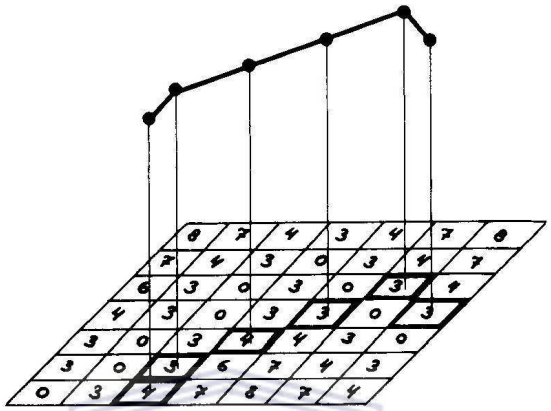
To ensure consistency, the image is repositioned according to the following conditions:

- If $X < 0$ and $Y < 0$, then move the image X pixels to the left and Y pixels upwards
- If $X < 0$ and $Y > 0$, then move the image X pixels to the left and Y pixels downwards
- If $X > 0$ and $Y < 0$, then move the image X pixels to the right and Y pixels upwards
- If $X > 0$ and $Y > 0$, then move the image X pixels to the right and Y pixels downwards

Following this operation, the greyscale transformation and edge detection method described in the previous phase is applied to the re-positioned image, illustrated in Figure 5.6. The operations mentioned thus far forms the image registration process. This

process is necessary as it contributes to a more reliable matching process. To match the test image with an example image in the database, the edge detected test image is superimposed on the distance transformed image. For every edge pixel in the test image, the corresponding pixel in the distance transformed image is summed, as shown in Figure 5.5.

FIGURE 5.5: An illustration of the method by which pixels are calculated from the distance transformed image [14]

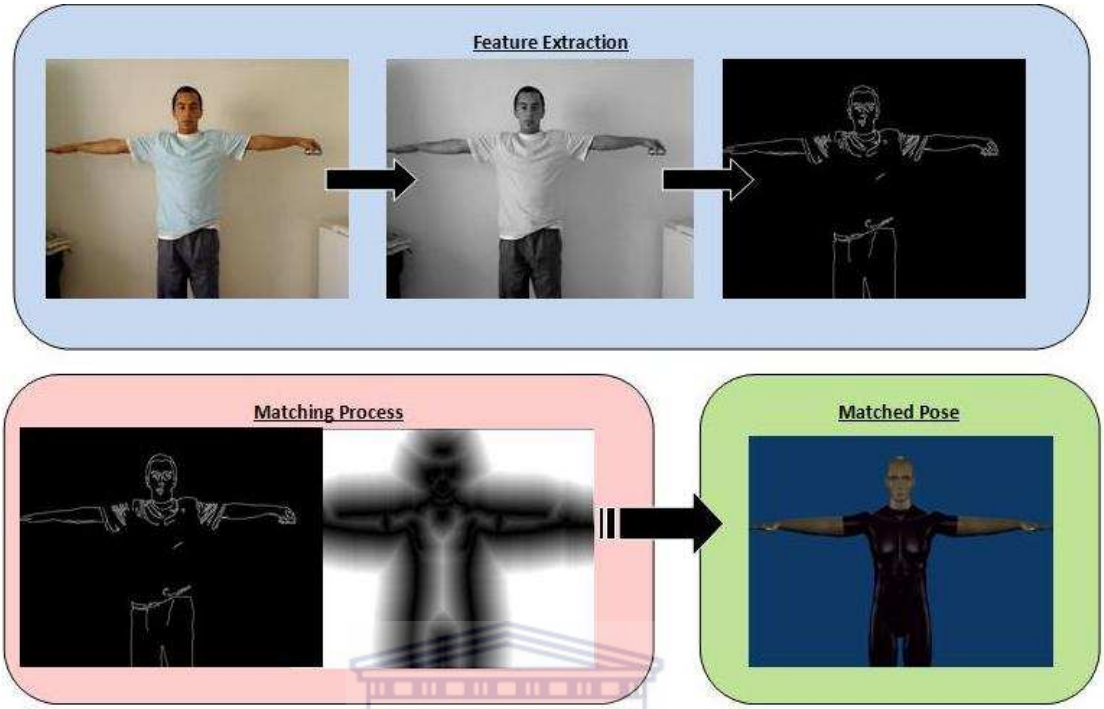


The sum for each distance transformed image is stored in the *calculated_distance* column in the database. After matching each image in the database, the distance transformed image with the lowest sum yields the best possible match, since pixels nearer to an edge will have lower pixel values. The example image corresponding to the matched distance transform image is then retrieved along with the 3D coordinates for the shoulders, elbows and wrists. The matching process is shown in Figure 5.6.

5.2 Learning-Based System Design

The aim of a learning-based approach is to extract features from an image and represent them as vectors. Thereafter a model is trained using these vectors, which is used to predict a pose. Before features can be extracted from images, image registration is required. The image registration procedure for the learning-based system is carried out differently to that of the example-based system. For each image, the features are represented as vectors and used to train an SVM. To find a pose, the SVM predicts the pose given a set of test vectors. A systematic design is provided to give an overview of the system. The overall system also consists of two phases: a training phase and a testing phase. The following sections explains these two phases.

FIGURE 5.6: A visual representation to the way the example-based system's matching process operates



5.2.1 Training Phase

This section describes the image registration procedure as well as the training process. Figure 5.7 illustrates the tasks to be carried out in this phase.

In this system, 2 individuals were used to perform the signs in the training phase. It is also possible to generate these signs using the 3D model. However, using different body types allows for better generalisation when training the SVM.

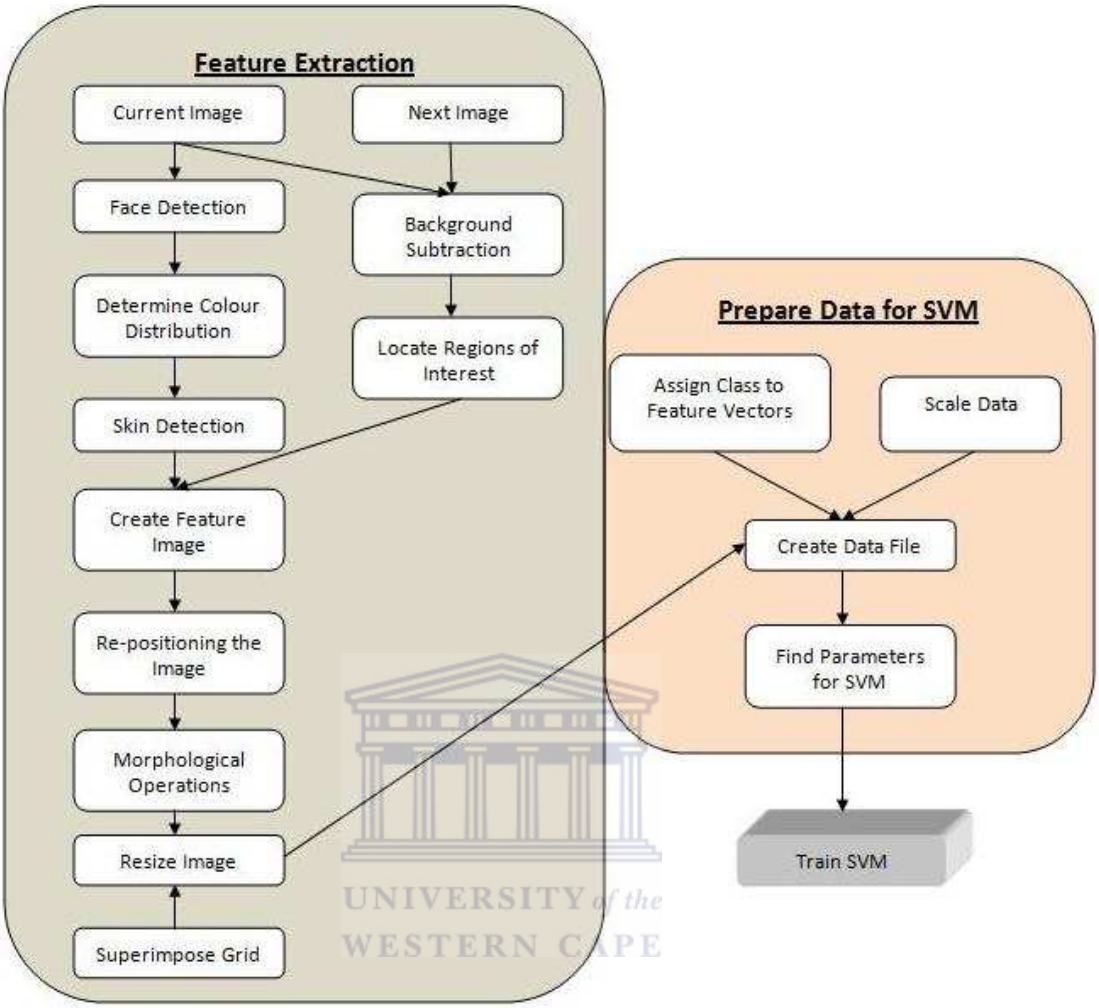
The first step in the image registration procedure is to detect the face in an image using the face detection method. After detecting the face, the centre of the face is obtained by dividing the height and width of the face by two. There are two reasons for this:

- To obtain a colour distribution of the individual's skin and,
- To reposition and normalise the image

As in the previous system, it is important that a consistent location be maintained. To address this challenge, each image is normalised so that consistency is achieved.

One of the unique attributes to this system and a very important one that differs from other skin detection methods, is that the region around the nose is used to determine the skin colour distribution. Using the coordinates for the centre of the face, this region

FIGURE 5.7: Learning-based system design and implementation for the training phase



around the nose can be found. A radius of 10 pixels was determined to be the optimal region after experimentation by means of trial and error, since it is less likely to contain any facial hair, colour objects or shadows that may affect the colour distribution as shown in Figure 4.2. By only using the Hue component, the colour distribution is not affected by intensity as this attribute is contained in the Value component of the HSV colour space. Using a colour histogram, the upper and lower threshold of the Hue component in the skin colour distribution is obtained. If condition 4.14 is satisfied when applying the thresholds, then a pixel is determined to be a skin pixel and assigned a value of 255. If condition 4.14 is not satisfied, the respective pixel is determined not to be a skin pixel and is assigned a value of 0, illustrated in Figure 4.2.

To eliminate false detections in the background that may fall in the same Hue range as the skin colour distribution, the skin detection method is only applied to regions of interest. These regions, in a sign language application, are the areas where sign language is performed. Hence, highlighting the areas where movement occurs is a means to

identify these regions. To identify where movement has occured, an adaptive background subtraction technique is used.

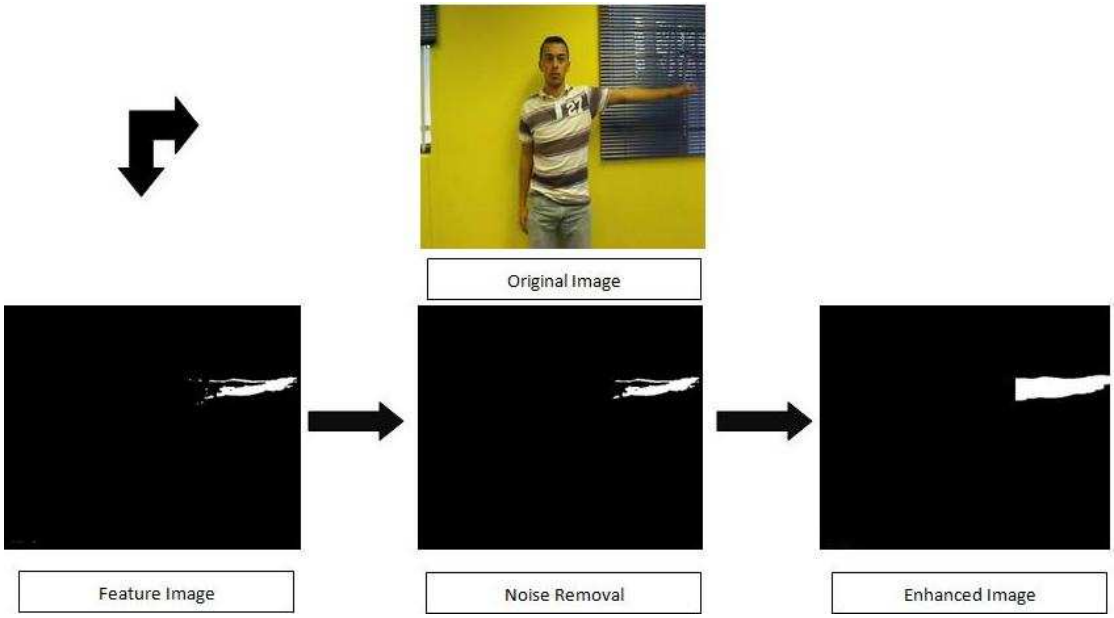
Using the image following the current image or a few images ahead, this form of background subtraction can be executed and the regions of interest can be located. When using the image following the current image, the difference between the two is less whereas using a few images ahead results in a larger difference. When the difference is small, the risk of detecting false regions of interest is less. Hence, it is opted to use the image following the current image. Applying the skin detection method only to the regions of interest results in an image consisting of pixels identified as skin with a greater part of false detections eliminated. These skin pixels in the image are the features that will be used to train the SVM. The image containing these features will be referred to as the feature image. Once the skin pixels have been detected, the feature image, rather than the original image, is re-positioned.

Due to the background subtraction technique being sensitive to noise and the skin detection method being sensitive to the Hue range, it is possible that the feature image may contain unnecessary features (noise). In the feature image, certain areas where skin have been detected may also contain discontinuities (holes). To compensate for these features, morphological operations are used. An ideal operation that forms part of the morphological operations technique is the opening operation. This operation involves the erosion operation followed by the dilation operation using a single structuring element for both operations. The erosion operation removes the noise followed by the dilation operation that fills the holes thereby enhancing the features, as illustrated in Figure 5.8.

Up to this point in the system, the images have been at a default size of 640x480 pixels. A large image contains more detail since it contains a greater number of pixels. This allows for more accurate execution of the previous techniques. When training on a very large number of features, the training and testing times are longer, especially when finding the optimal parameters for the SVM. Using an image size of 640x480 pixels amounts to 307 200 features. When training on approximately 1 500 images, the number of features amounts to 460 800 000 features. An efficient way to reduce the number of features, while retaining the essence thereof, is simply to resize the image. By means of trial and error, an image size of 40x30 pixels was shown to be suitable as it contains enough features to be distinguished from others. Resizing each image to 40x30 pixels is attained by averaging every 16 pixels into a single pixel. In each feature image, the information regarding the image height, width and channels is also contained in the image.

When creating the data file, this information is discarded and only the pixels (features) are used. An index is assigned to each feature, as illustrated in Figure 5.9. Furthermore,

FIGURE 5.8: The morphological operations used to remove noise and enhance the features.



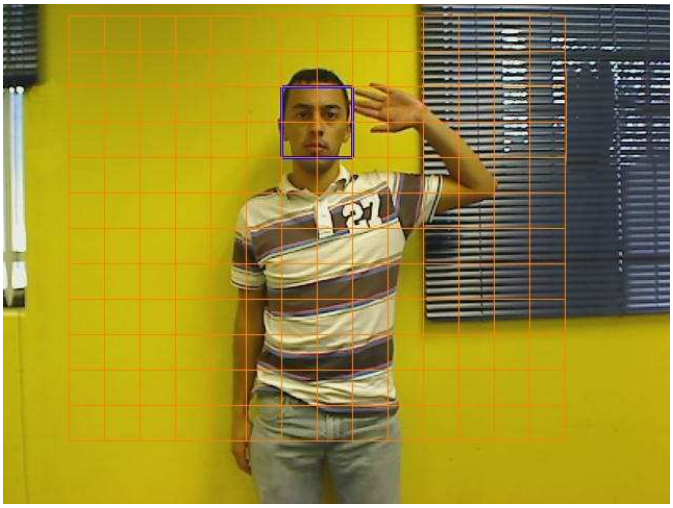
each feature vector (feature image) should be assigned to a class, also referred to as assigning a label. To estimate a pose using the 3D model, the position of the wrist can be used to build the posture. Before the position of the wrist can be used, the position needs to be identified. Thus, the purpose of the SVM is to identify the position of both wrists. It is a challenging task to randomly assign a label to a wrist by means of observation. A more effective and structured way to assign a label, is to superimpose a grid on the image. A grid consisting of 168 blocks, in effect, covers the entire pose space. Each block corresponds to a class in the SVM and, thus 168 classes are used, as illustrated in Figure 5.10. If a wrist is observed in block 131, it is assigned label 131. Since both wrists are to be assigned a label, a multi-class problem exists.

FIGURE 5.9: A representation of the data file without labels

```
1:0 2:0 3:0 4:0 5:0 6:0 7:255 8:255 9:255 10:0
11:0 12:0 13:0 14:255 15:0 16:0 17:255 18:0
19:255 20:0 21:0 22:0 23:255.....
..... 1180:0 1181:0
1182:0 1183:0 1184:0 1185:0 1186:0 1187:0 1188:0
1189:255 1190:255 1191:255 1192:255 1193:0 1194:0
1195:0 1196:0 1197:0 1198:0 1199:255 1200:255
```

Scaling the data is another element in preparing the data for the SVM. The advantage of scaling the data is to avoid features with a greater numeric range dominating those in a smaller numeric range [22]. Therefore, when creating the data file, a pixel with a value of 255 is converted to 1 and a pixel with a value of 0, remains 0. This ensures that

FIGURE 5.10: An illustration when the grid is superimposed on the image.



features are presented in a range of [0,1]. The next step in the system is to obtain the kernel parameters. Keerthi and Lin [71] have shown that using a linear kernel with a parameter C and a RBF kernel with parameters C and γ, have the same performance. Furthermore, Lin and Lin [81] have shown that the sigmoid kernel behaves similar to the RBF kernel with certain parameters. In addition, the polynomial kernel, compared to the RBF kernel, has more hyperparameters which increases the complexity of the SVM [22]. It is therefore reasonable to begin experimentation using the RBF kernel.

When finding the parameters for the RBF kernel, two parameters are needed, C and γ. To effectively train the SVM and accurately predict test data, the best C and γ parameters need to be selected for the given problem. An exhaustive approach to finding these parameters is to manually try each C and γ combination, where each parameter is an exponentially growing sequence. An alternative approach is to use the *grid-search* function in LibSVM that uses cross-validation. Cross-validation divides the training set into n equal sized subsets, where the classifier is trained on the $n-1$ subsets and tested on the remaining subset for each parameter combination [22]. Thus, the combination with the best cross-validation accuracy for the given problem is chosen. Finally, before training the SVM, the data file is presented in the format depicted in Figure 5.11.

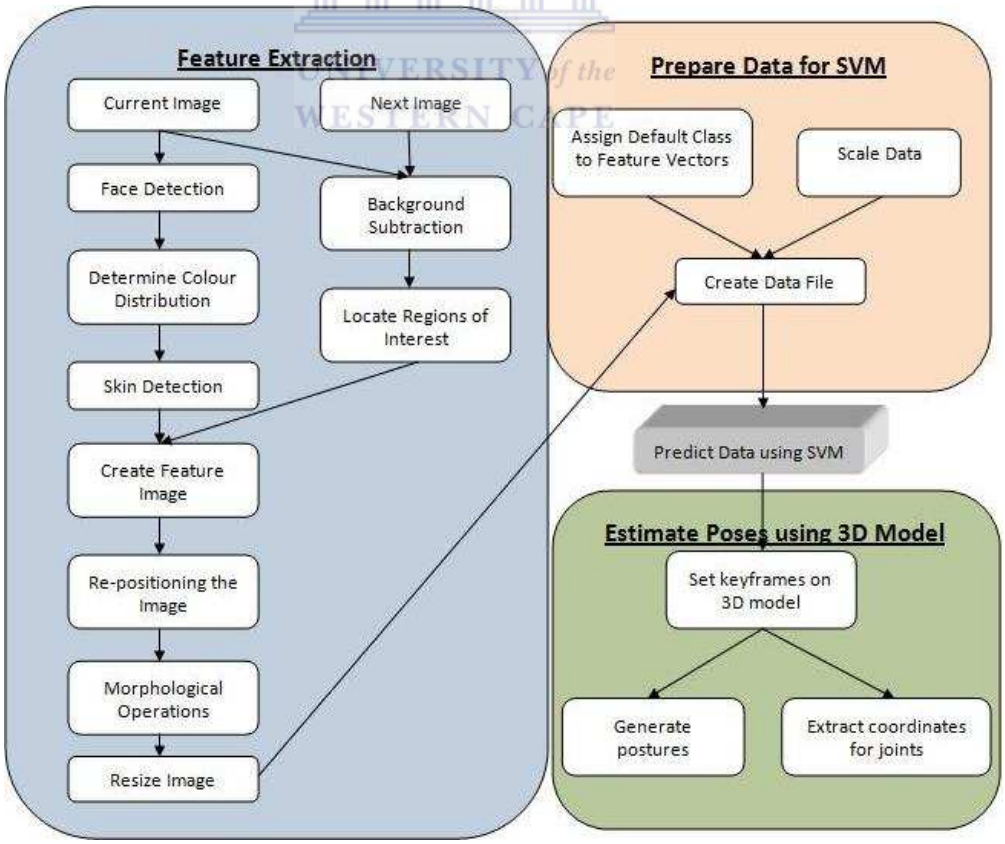
5.2.2 Testing Phase

This section describes the procedure for evaluating the learning-based system given a test set. This phase takes place iteratively as illustrated in Figure 5.12.

FIGURE 5.11: A representation of the data file with labels in the training phase.

```
132,135 1:0 2:0 3:0 4:0 5:0 6:0 7:255 8:255
9:255 10:0 11:0 12:0 13:0 14:255 15:0 16:0
17:255 18:0 19:255 20:0 21:0 22:0
23:255.....
..... 1180:0 1181:0 1182:0 1183:0 1184:0
1185:0 1186:0 1187:0 1188:0 1189:255 1190:255
1191:255 1192:255 1193:0 1194:0 1195:0 1196:0
1197:0 1198:0 1199:255 1200:255
131,136 1:0 2:0 3:0 4:0 5:0 6:0 7:255 8:255
9:255 10:0 11:0 12:0 13:0 14:255 15:0
16:0.....
```

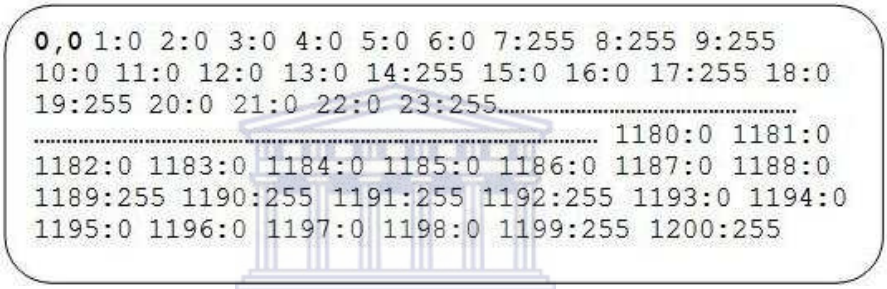

FIGURE 5.12: Learning-based system design and implementation for the testing phase.



When testing the system, 6 individuals were used, each performing 15 signs, i.e. 90 signs in total. The 6 individuals consisted of 3 males and 3 females so that an unbiased experimentation can be maintained.

The testing phase of the system follows the same image registration procedure and feature extraction methods as the training phase. It differs from the training phase immediately after the image has been resized. When creating the data file, the information regarding the image height, width and channels are discarded and only the pixels (features) are used. Furthermore, an index is assigned to each feature. For each feature vector a default label 0,0 is assigned to it, i.e. a label 0 for each wrist. This label refers to a test feature vector. Thus, before predicting the data, the data file for the testing phase is presented in the format depicted in Figure 5.13.

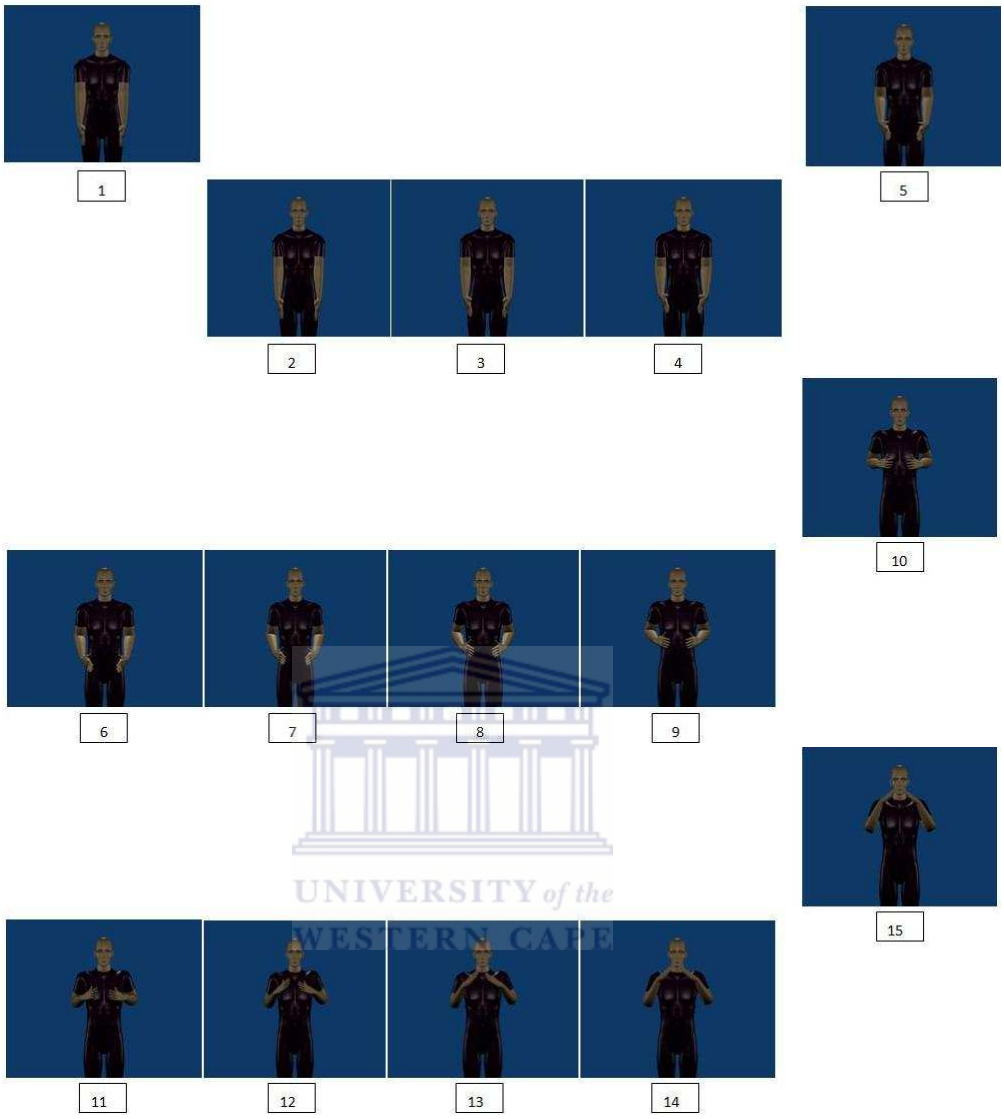
FIGURE 5.13: A representation of the data file with labels in the testing phase.



Using the trained model, the SVM predicts the label corresponding to each feature vector. For instance, for some unknown feature vector, a label 70,84 is predicted. This label is interpreted as class 70 for the right-hand wrist and class 84 for the left-hand wrist. According to the grid used in the training phase, 168 blocks exist; hence, 168 classes are used. If the wrist is predicted to be in class 70, this means the wrist is considered to be in block 70. Therefore, a predefined position is assigned to each block that is used to position the 3D model in the predicted posture. Hence, if the wrists are predicted to be in blocks 70 and 84 respectively, the 3D model is positioned according to the coordinates assigned to each of these blocks.

A feature present in Blender, and one of the reasons why it is used, is that keyframes can be created dynamically. For the set of keyframes that are created, Blender interpolates between them. Therefore, when poses are estimated in an entire sign phrase, only certain frames need to be used to set the keyframes on the 3D human body model in Blender. This allows the 3D human body model in Blender to automatically generate the postures from one keyframe to the next. For instance, as illustrated in Figure 5.14, if every fifth frame is predicted and then used to set the keyframes, the 3D human body model in Blender automatically generates poses for frames 2-4, 6-9, 11-14 and so forth.

FIGURE 5.14: Automatically generated poses using Blender for the in-between frames.



Thus an entire sign phrase can be reconstructed using fewer frames. It should be noted that even though the wrists are the only joints that are predicted, positioning the wrist at a certain position automatically positions the elbows and shoulders relative to that position. Inverse kinematic constraints on the 3D model ensures that the movement of the body parts are human-like. The most important feature of incorporating the 3D model in Blender with this learning-based system is that, along with the animation for each sign phrase, the 3D coordinates for the wrists, elbows and shoulders are automatically generated by the 3D model. Hence, the postures for an entire sign phrase can be estimated using this system.

5.3 Summary

In this chapter the design and implementation of the two systems were discussed, namely the example-based system and the learning-based system. Each system consisted of two phases and each was individually described.

In the example-based system, a large number of images are generated using a 3D model in Blender along with the coordinates of the wrists, elbows and shoulders. The file location of these images, the file location of the distance transformed images and the coordinates are stored in a database. To produce a distance transformed image, the generated images are greyscaled, edge detected and transformed using the Chamfer Distance Transformation. Prior to estimating, image registration is applied to the test images. This image registration consists of a simple background subtraction method, face detection and the repositioning of the image. The repositioned image is then greyscaled and edge detected. Using these edges, the corresponding pixel in the distance transformed image is summed and the image with the lowest sum yields the best match. The matched image along with the coordinates for the wrists, elbows and shoulders are retrieved.

In the learning-based system, 2 individuals were used for the training phase and 6 individuals for the testing phase. Both phases require similar image registration and feature extraction methods. These consist of face detection that is not only used to reposition an image but also determines the skin colour distribution of the person. The adaptive background subtraction method is used to locate regions of interest and, in conjunction with the skin detection method, creates the feature extraction method. This is followed by morphological operations to reduce noise and enhance features. To reduce the large number of features, the feature image is resized to a smaller scale. When creating the data file, the data is scaled to a range of [0,1]. The labels are manually assigned to each feature vector in the training phase, whereas a default label is assigned to each feature vector in the testing phase. In the training phase, a *grid-search* is computed to find the optimal kernel parameters using tools in LibSVM. These parameters are used to train an SVM model. In the testing phase, the trained model is used to predict the labels when presented with unseen test images. These labels are further used to set keyframes on the 3D model in Blender to automatically generate the postures from one keyframe to the next. Additionally, the 3D model is used to estimate the coordinates of the wrists, elbows and shoulders.

In the next chapter, these systems are evaluated, a comparison is made between the two and the results are discussed.

Chapter 6

Experimental Results and Analysis

In this chapter the two approaches are compared and the accuracies are determined. Before describing the experiments conducted, the database, sign language data, training data and test data are explained. Explaining the sign language data is important, not only to understand how it is linguistically used but also how it affects each system.

For all experiments, the measurement of accuracy is explained and the assessment of the outputs of each system are described. Before analysing the learning-based system, the optimal kernel has to be identified. Experimental analysis is performed on both the example-based system and the learning-based system. Experimental analysis is also performed on the effectiveness of the learning-based system when predicting a given set of frames. In order to find a subsystem for the SASL system, a comparison is discussed between the two.

The experiments are aimed at determining the success rate of the two systems as well as its suitability as an application for a sign language system. It is also aimed at identifying areas for future work.

6.1 Experimental Setup

6.1.1 Experimental Setting

In the experimental setting, a single webcam[1] was placed on a tripod and connected to a notebook. The notebook was used for its portability and only used to capture the

[1] Logitech Quickcam Webcam

video. The evaluation of the systems was carried out on an Intel i7 desktop computer.

The specifications for the desktop computer were:

- Processor: Intel(R) Core(TM) i7 CPU @ 3.20GHz
- Memory: RAM 3Gb
- Operating System: Ubuntu Karmic Koala

6.1.2 Sign Language Data

In South African sign language, no standard set of poses exist upon which these experiments can be based. Sign language consists of numerous varied poses each of which conveys a different meaning and each of which is equally important. As a subsystem of an automatic sign language translation system, it is important that poses which are used to express words in sign language, can be recognised. For experimental purposes on the effectiveness of the systems towards SASL, 15 South African sign language words were chosen. The distinct poses that form the words do not cover the entire sign language vocabulary, however, an immense effort was made to choose words which consist of poses that cover the vocabulary to a large extent, such that signs performed on the far left and far right of the body are all well represented.

These words were selected from the "Fulton School for the Deaf SASL Dictionary" [55]. An illustration of the 15 words are shown in Figure 6.1, followed by a brief description of each. The words are given an abbreviation and are referred to where necessary.

6.1.3 Database Generation

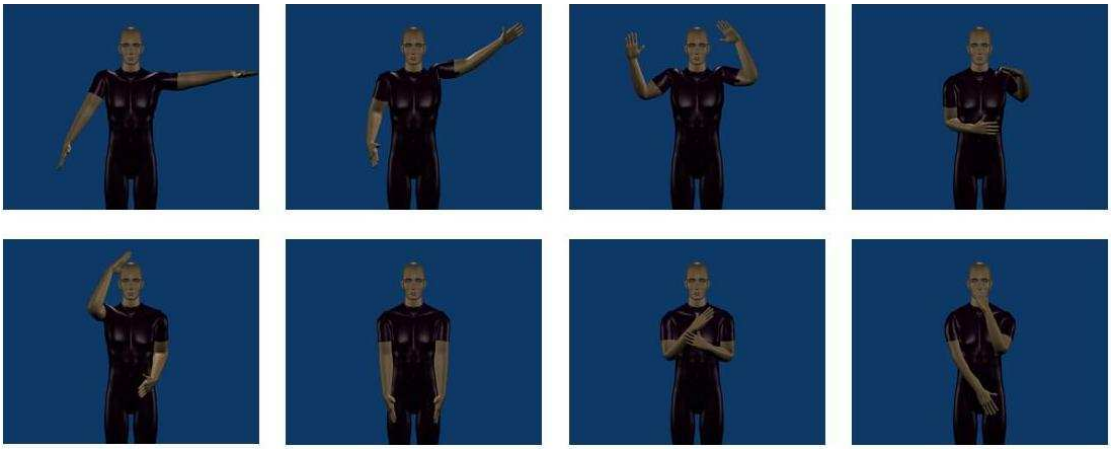
For the example-based system, a database containing upper body poses is required. To date, such a database is not publicly available. Similar to other researchers [122][138][25][5] that have generated their own set of poses using POSER or other alternatives, the 3D model discussed in section 3.5 is used to generate a large number of upper body poses. Hand shapes are beyond the scope of this research and is the subject of other research projects. For the purpose of these experimentations, the poses were performed with an *open* hand, as the objective is to find the positions of the wrist, elbow and shoulders regardless of the hand shape. The poses were generated so that a combination of the left and right arm is generated within the DOF of the human body. This resulted in a database consisting of 12 660 poses. A sample set from the database is illustrated in Figure 6.2.

FIGURE 6.1: An illustration of the sign words used in this experimentation

TABLE 6.1: A brief description of the sign language words used

Sign	**Abbreviation**	**Description**
Away	S1	Using the right hand, move to and fro away from right side of the body.
Bye-Bye	S2	Right hand is in front of the right shoulder while waving the hand inwards to the left and outwards to the right.
Cracker	S3	Both hands are in front of the chest, while moving hands away from each other to the sides.
Curtains	S4	Both hands are above the shoulder and moving towards the face and outwards again.
Dress	S5	Both hands are in front of the chest and move them down the sides of the body. When reaching below the hips, move hands away from the body.
Eat	S6	Using both hands, mimic an eating gesture similar to eating with chopsticks.
Left	S7	Raise left hand away from the left side of the body.
Light	S8	Move right hand above right shoulder just above the head.
Love	S9	Using both hands, cross wrists in the middle of the upper chest as one holds hands against oneself.
Right	S10	Raise right hand away from the right side of the body.
Run	S11	Using both hands on the side of the chest, move hands up and down imitating a running movement.
Toast	S12	Using both hands in front of the chest, put the right hand in front if the left hand. This is followed by moving the right hand upwards indicating the toast popping up.
We	S13	Right hand in front of the chest, move hand from the right shoulder across chest to the left shoulder.
Wide	S14	Raise right and left hand away fron the sides of the body.
Why	S15	Use right hand on left side of the chest and tap twice against chest.

FIGURE 6.2: A sample set taken from the database



6.1.4 Training Set for Learning-Based System

As previously mentioned, video data consisting of individuals performing the aforementioned signs in SASL are not publicly available. To obtain video data on which to train the learning-based system, two individuals were asked to perform these signs. The two individuals were a male and female, and are shown in Figure 6.3. They will be referred to as Person A for the male and Person B for the female. In total, this resulted in 30 signs where each sign consisted of a total number of frames that ranged from 50 to 104.

FIGURE 6.3: The individuals used, first for training and later for testing

Person A Person B

For each individual, the signs begin and end in the *neutral* state (hands on the side of the body). This ensures consistency in the sign language video as well as the subsequent analysis.

6.1.5 Testing Set for the Example-Based and Learning-Based System

In testing the systems, the same individuals used for the training phase were also used for the testing phase. It should be noted that the video data captured from these individuals in the testing phase were distinct from the video data in the training phase.

In addition to Person A and Person B, 4 individuals, 2 males and 2 females, were asked to perform the signs. This offers the opportunity to draw a conclusion based on whether the systems perform differently on different body types. The individuals also represent a wide range of skin colours so as to provide a good examination of the skin detection method proposed in 4.1.7. Henceforth, these individuals are referred to as Person C, Person D, Person E and Person F in order of appearance as shown in Figure 6.4.

FIGURE 6.4: The remaining individuals, each with a different skin colour, used for testing in this experimentation

Altogether, the 6 individuals performed 15 signs each beginning and ending in the *neutral* state for each sign. This resulted in 90 signs being performed. The signs performed by Person A and Person B in the test set are different to the signs performed for the training set. Thus, in the learning-based system, the testing set is disjoint from the training set and is considered to be *unseen* data. A complete list regarding the number of frames per sign and per individual is shown in Table 6.2.

6.1.6 Metric of Accuracy

For each system, a binomial experiment analysis is used to measure the accuracy. A binomial experiment involves a series of independent and identical Bernoulli trials [73]. A Bernoulli trial is a non-deterministic experiment with one of two outcomes, success or failure [73]. To consider an experiment as a Bernoulli trial, it should adhere to three criteria [153]:

1. There should only be 2 possible outcomes, either success or failure.
2. For each trial, an outcome should have a fixed probability, where p is the probability of a success and $q = 1 - p$ is the probability of a failure.
3. Each trial and outcome is independent of each other.

This experiment is similar to the scenario of obtaining a *six* when rolling a die, where a *six* is considered to be a success and everything else a failure. When analysing a frame in

TABLE 6.2: A list consisting of the number of frames captured per individual in each recording

Sign \ Subject	**Person A**	**Person B**	**Person C**	**Person D**	**Person E**	**Person F**
Away	104	82	129	98	167	118
Bye-Bye	121	84	98	103	123	108
Cracker	106	77	92	89	94	64
Curtains	106	75	134	112	102	92
Dress	69	90	91	90	103	86
Eat	80	69	93	90	90	87
Left	59	56	113	97	66	56
Light	110	91	117	88	104	65
Love	67	62	86	74	55	64
Right	71	61	104	86	61	48
Run	71	92	89	86	84	65
Toast	113	88	122	82	86	93
We	89	79	81	80	75	63
Wide	77	52	102	89	68	84
Why	50	56	77	68	62	49
Total	1293	1114	1528	1332	1340	1142

a sign, the same criteria is adopted. For the purpose of this experimentation, a pose that is recognised as a *match* is considered to be a success and a pose that is not recognised as a *match* is considered to be a failure.

When determining the significance of the results, the set of poses in each sign will be analysed separately. The binomial probability distributions for these results are calculated and listed in Appendix A. For each sign, a chi-square test is performed, which is an approximate test, using the total number of correct poses in each sign to determine if the proportion of successfully recognised poses in each sign are the same for the example-based and learning-based system. In addition, the McNemar's test is also performed, which is a direct significance test, using the number of correct poses against the number of incorrect poses when comparing the two methods.

6.1.7 Assessment Criteria

Four sets of evaluation were to be carried out. These consist of an evaluation on the performance of the example-based and learning-based system on every fifth frame, an evaluation on the performance of the different kernels using SVMs and an evaluation on the performance of the learning-based system on every frame. To prevent a biased assessment towards this research, 3 individuals were chosen to assess the output of both systems and are henceforth referred to as Assessor A, Assessor B and Assessor C. For

each evaluation, each assessor was randomly given five sets of output to assess. Each set of output consisted of the outputs for each of the 6 individuals. Instructions to the assessors were, given an input pose, if the output pose matched the input pose, then it should be labelled a success. Similarly, if the output did not match the input pose, then it should be labelled a failure. Evaluation of the systems was in the following order:

TABLE 6.3: Evaluation of example-based system on every fifth frame

Evaluation of example-based system on every fifth frame	
Assessor A	away, bye, cracker, curtains, dress
Assessor B	eat, left, light, love, right
Assessor C	run, toast, we, wide, why

TABLE 6.4: Evaluation of different kernels in SVM

Evaluation of different kernels in SVM	
Assessor A	eat, left, light, love, right
Assessor B	run, toast, we, wide, why
Assessor C	away, bye, cracker, curtains, dress

TABLE 6.5: Evaluation of learning-based system on every fifth frame

Evaluation of learning-based system on every fifth frame	
Assessor A	run, toast, we, wide, why
Assessor B	away, bye, cracker, curtains, dress
Assessor C	eat, left, light, love, right

TABLE 6.6: Evaluation of learning-based system on every frame

Evaluation of learning-based system on every frame	
Assessor A	away, bye, cracker, curtains, dress
Assessor B	eat, left, light, love, right
Assessor C	run, toast, we, wide, why

6.2 Results and Discussion

The number of frames for each sign, in the test set, range from 48 to 167 with 7749 frames in total. Due to this large number of frames, it would be time consuming to analyse each frame. If one considers the speed at which the webcam captures a frame, that is, 25 frames every second; it is clear that very little change occurs between every five frames. An example is shown in Figure 6.5. Hence, a more meaningful approach would be to analyse every fifth frame, thereby only looking at significant changes.

From the view point of a human observer, slight variations in the position of the body parts is not easily identified. It is, however, possible to estimate the positions by mere

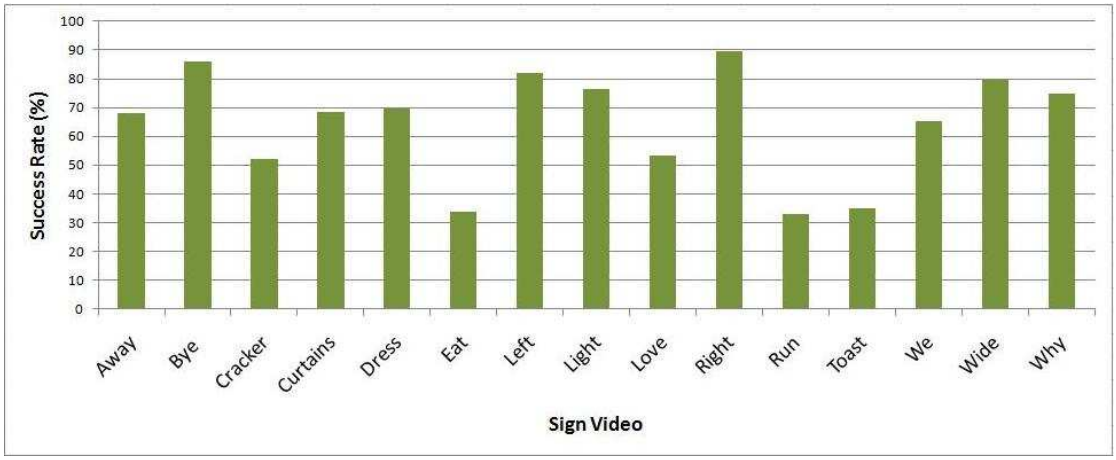
FIGURE 6.5: An example of five consecutive frames

observation. Thus, similarly to other researchers in this field [10][18][105], determining whether or not a pose has been successfully matched can only be done by means of a subjective evaluation.

6.2.1 Example-Based System Results

In the example-based system, the input was a frame from the test set and the output was a pose retrieved from the database. The retrieved pose was the image with the least distance value. The number of times a successfully matched pose was achieved was recorded and divided by the number of frames evaluated in the sign. This resulted in an average success rate per sign. A complete list of the success rate of each sign per subject is shown in Table 6.7.

FIGURE 6.6: A graphical representation of the success rate for the example-based system on every fifth frame



From the results in Figure 6.6, six signs - "bye", "left", "light", "right", "wide" and "why" - obtained a success rate greater than 70%. These signs make up close to 50% of the signs tested. An additional three signs - "away", "curtains" and "dress" - obtained a success rate very close to 70%. Therefore 60% of the signs tested, obtained a success rate close to or more than 70%. This result is very encouraging. Three signs achieved

TABLE 6.7: Success rate for the example-based system

Sign / Subject	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	**Average**
Person A	75.00	100	81.82	68.18	100	41.67	86.36	100	71.43	100	46.67	34.78	88.89	68.75	100	77.57
Person B	68.24	68.75	68.75	73.33	66.67	30.71	66.67	84.21	69.23	61.54	26.32	33.33	87.50	63.64	91.67	64.04
Person C	81.25	95.24	57.89	70.37	63.16	36.84	100	79.17	55.56	100	33.33	44.00	50.00	94.44	56.25	67.83
Person D	62.50	90.48	27.78	65.22	55.56	22.22	100	72.22	26.27	100	33.33	27.28	43.75	88.89	57.14	58.18
Person E	62.50	81.82	23.08	42.12	61.11	33.33	75.00	61.54	61.54	100	23.08	31.58	53.85	82.35	60.00	56.86
Person F	60.00	80.00	52.63	90.48	71.43	38.89	64.29	61.90	36.36	76.92	35.29	38.89	66.67	78.57	84.62	62.46
Average	68.25	86.05	51.99	68.28	69.66	33.94	82.05	76.51	53.40	89.74	33.00	34.98	65.11	79.44	74.95	**64.49**

a success rate greater than 50%, in addition to three signs - “eat”, “run” and “toast” - that obtained a success rate less than 50%. No sign was completely unrecognised.

From Figure 6.1, it can be seen that signs that obtained a success rate greater than 70%, are signs that are performed away from the body, allowing for the edges to be more clearly defined. This in turn, allows for a more accurate distance measure with a lower chance of obtaining a mismatch. Furthermore, signs that have obtained a success rate between 60% and 70% are signs that are performed partially away from the body. Signs with a success rate less than 60% are signs that are mostly performed in front of the torso region. This indicates that occlusion of the torso by the arms may produce similar silhouettes. In addition, edges formed by the type of clothing worn, is often the source of a mismatch.

In terms of accuracy according to the subjects, Person A obtained a success rate greater than 70%, three subjects obtained a success rate above 60% and two subjects obtained a success rate close to 60%. Furthermore, the success rates across the different genders were similar with both male and female subjects obtaining a success rate close to or greater than 60%.

The average success rate of the system across all subjects and signs was 64.49%. In comparison to other related systems, the system performs marginally better but comparable to Cao [21] that achieved a success rate of 63% using their eigen-chamfer method with kernel principle component analysis (KPCA) re-ranking and fine search evaluation. Unfortunately, the results obtained by Sminchisescu and Telea [145], Micilotta et. al. [94] and Mori and Malik [106], are visually presented and therefore a comparison cannot be made with their success rate.

6.2.2 Learning-Based System Results

In the learning-based system the input was a video of a signed word from the test set. In this system a video is used as input since the adaptive background subtraction technique takes the difference of every two sequential frames in a video. The output is a label predicted by the trained SVM. Each predicted label with predefined coordinates is then set as a keyframe on the 3D body model in Blender. The keyframes are used as a start and end point so that the 3D model is able to interpolate between these two points. All frames are then extracted from the animation output where each frame consists of an individual pose. The number of times a success was obtained was recorded and divided by the number of frames evaluated in the sign. This resulted in an average success rate per sign.

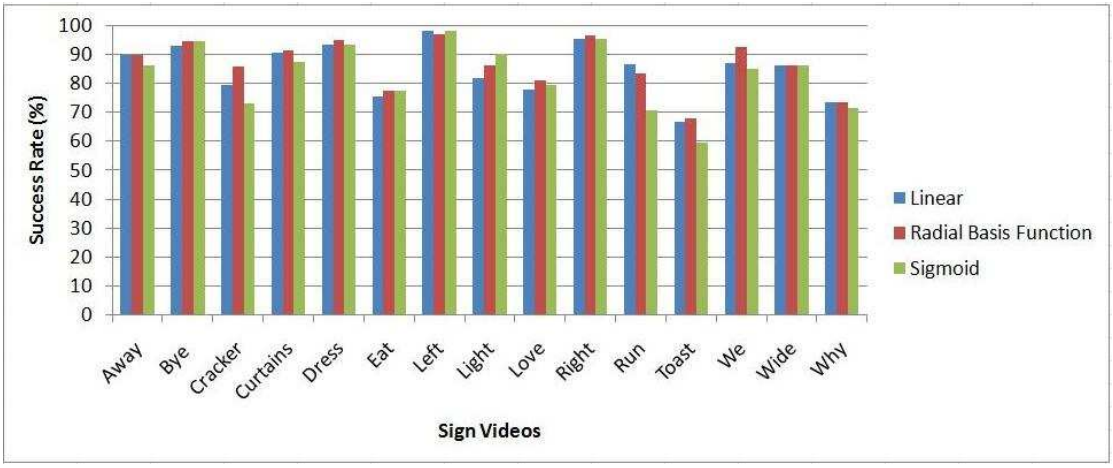
6.2.2.1 Kernel Suitability

Before this system can be evaluated, an appropriate kernel needs to be identified as it has a great influence on the predictive capabilities of the SVM [51][57]. Hsu et. al. [57] suggest that the RBF kernel is a reasonable kernel to begin experimentation in accordance with the theories proposed by Keerthi and Lin [71] and Lin and Lin [81]. The RBF kernel may be a reasonable choice but is not always the best choice since the kernel's performance is dependent on the target application and set of features [83]. Unfortunately, no standard method exists to find the most appropriate kernel [168] and selecting an appropriate kernel is often a process of trial and error [27]. Hence, these kernels are evaluated with the aim of finding the most appropriate one for this system.

Each kernel was trained on the training set discussed in section 6.1.4. Using the *grid-search* function in LibSVM, the parameters C and γ were obtained where C was 512.0 and γ was 0.0001220703125.

The test set used for this evaluation was the 15 signs performed by Person C. The same test set was used on all the kernels since a comparison was to be made between them. Conducting this evaluation using the polynomial kernel have not produced any significant results regardless of the degree used. Therefore, the results for this kernel are not included. The success rate for the linear, RBF and sigmoid kernels are listed in Table 6.8.

FIGURE 6.7: Comparing the success rate of the kernels



Analysing Figure 6.7, all kernels had comparable performances. However, the RBF kernel obtained either the same or a higher success rate than the other kernels in 12 of the 15 signs. On the sign "left" both the linear and sigmoid kernel obtained a better success rate than the RBF kernel. On the sign "light", the sigmoid kernel obtained a higher success rate that the other kernels, while on the sign "run", the linear kernel had the highest success rate. Overall, the RBF kernel had an average success rate of 86.44%

TABLE 6.8: Success rate of each of the kernels

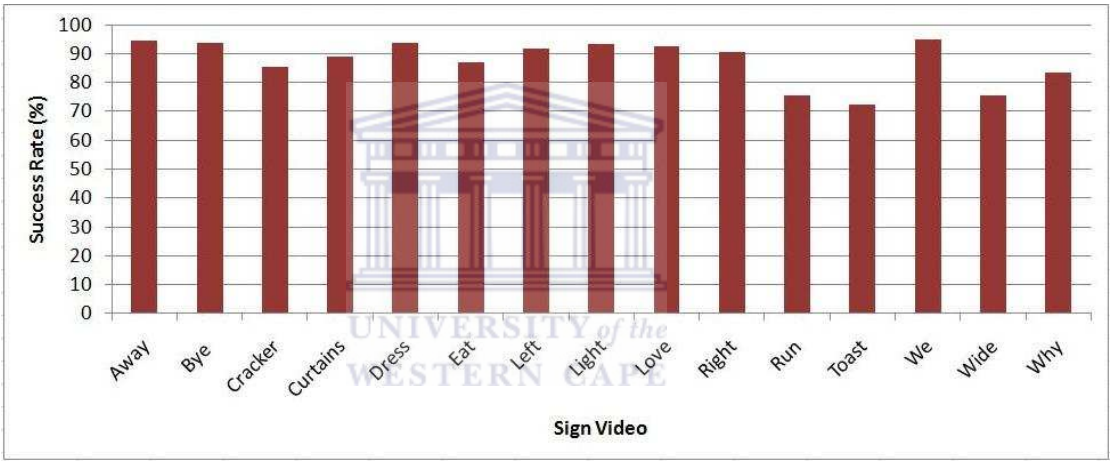
Sign / Kernel	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	**Average**
Linear	89.66	93.06	79.37	90.38	93.42	75.27	97.96	81.94	77.59	95.18	86.36	66.67	86.79	86.05	73.47	84.88
Radial Basis Function	89.66	94.44	85.71	91.35	94.74	77.42	96.94	86.11	81.03	96.39	83.15	67.71	92.45	86.05	73.47	86.44
Sigmoid	86.21	94.44	73.02	87.50	93.42	77.42	97.96	90.28	79.31	95.18	70.79	59.38	84.91	86.05	71.43	83.15
Average	88.51	93.98	79.37	89.74	93.86	76.70	97.62	86.11	79.31	95.58	80.10	64.59	88.05	86.05	72.79	**84.82**

compared to the linear and sigmoid kernel that had an average success rate of 84.88% and 83.15%, respectively. Hence, the RBF kernel is shown to be a more suitable kernel for this system and was therefore used for evaluation of the learning-based system.

6.2.2.2 Success Rate of Learning Based System

In order to perform a comparison with the example-based system, the same test data was used. In addition, every fifth frame of the video for each signed word was classified by the SVM. The output was sent to the 3D body model to animate the signed word. A complete list of the success rate of each word in this evaluation is presented in Table 6.9.

FIGURE 6.8: A graphical representation of the success rate for the learning-based system on every fifth frame



From the results in Figure 6.8, it can be seen that every sign obtained a success rate greater than 70%. In addition, the majority of the signs - eight signs - obtained a success rate above 90% with the highest success rate being 94.83%. Four signs - "cracker", "curtains", "eat" and "why" - obtained a success rate between 80% and 90%. Furthermore, three signs - "run", "toast" and "wide" - obtained a success rate between 70% and 80%.

Analysing the accuracy on a per subject basis, every subject obtained a success rate greater than 80% with Person D obtaining the highest success rate of 92.26%. The results indicated that the system performed equally well on the different body types and more importantly on the different skin colours of the subjects.

Two signs performed in front of the torso region - "toast" and "run" - obtained a lower success rate than other signs. However, three signs - "cracker", "love" and "we" - that are also performed in front of the torso region were amongst the signs that obtained the highest success rates. Therefore, performing signs in front of the torso region do not affect the learning-based system as much as in the example-based system.

TABLE 6.9: Success rate of the learning-based system on every fifth frame

Sign / Subject	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	**Average**
Person A	95.00	92.00	72.73	81.82	90.00	66.67	95.45	92.86	92.86	93.33	80.00	65.22	94.44	87.50	80.00	85.33
Person B	94.12	93.75	81.25	93.33	94.44	92.86	91.67	94.74	92.31	84.62	73.68	66.67	93.75	45.45	83.33	85.06
Person C	87.50	95.24	84.21	92.59	94.74	78.95	95.65	87.50	83.33	95.00	55.56	68.00	94.12	83.33	75.00	84.71
Person D	93.75	90.48	94.44	91.30	94.44	94.44	100	94.44	93.33	100	77.78	77.78	100	88.89	92.86	92.26
Person E	95.83	95.65	84.62	78.95	94.44	94.44	75.00	100	92.31	100	76.92	73.68	100	76.47	100	89.22
Person F	100	96.00	94.74	95.24	95.24	94.44	92.86	90.48	100	69.23	88.24	83.33	86.67	71.43	69.23	88.48
Average	94.37	93.85	85.33	88.87	93.88	86.97	91.77	93.34	92.36	90.36	75.36	72.45	94.83	75.51	83.40	**87.51**

Furthermore, variations in body sizes and structures between subjects did not affect the success rate of the system since the success rate between subjects was favourable as well as comparable to each other.

On the other hand, it is possible that the speed at which a signed word is performed can affect the success rate, whereby performing signs at a slower speed yields better results and performing signs at a faster speed negatively affects the results. This stems from the method of background subtraction used. If two frames are almost identical and have a small difference, then the number of features produced by the background subtraction technique will be few. Moreover, since morphological operations is a subsequent pre-processing step, the erosion operation further reduces the number of features. Thus, the number of features used to predict the pose would be very different from the features that would be required for a correct prediction. This in turn affects the ability to completely imitate a signed word since one incorrectly predicted pose causes the 3D body model to stray from the correct path to be followed. The system obtained a 100% success rate in 9 instances, thereby being able to successfully recognise and estimate a complete sign language word.

A comparison of the system with other learning-based systems, reveals this system has a higher success rate than Qiang et. al. that achieved a success rate of 80% using an ISM and RVM. It also achieves a much higher success rate than Ronfard et. al. [128] that used scale and orientation specific Gaussian derivative filters with a 75% success rate for SVM and 54% for RVM. In addition, it compares favourably to the LLE framework developed by Elgammal and Lee [36] with an overall classification rate of 93.05%. Unfortunately, other related systems do not define their success rate and thus further comparison with their work cannot be done.

6.2.2.3 Predicting Every Frame or Skip a Few?

The system is unique in that it offers the possibility to predict frames at a particular step size such as every frame, every third, fifth, tenth frame and so on. In rendering the output the 3D model interpolates between these frames. Thus, to evaluate the influence of the frame step size, the previous evaluation of section 6.2.2.2 is compared to the evaluation on the test set where every frame is predicted. In the previous case, every fifth frame was predicted and the 3D body model would interpolate the remaining frames. In this case, every frame is predicted and the 3D body model would interpolate between every frame rather that the span between every fifth frame. The results pertaining to the evaluation on every frame are displayed in Figure 6.9 and a complete list is shown in Table 6.10.

TABLE 6.10: Success rate of the Learning-Based System on every frame

Sign / Subject	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	**Average**
Person A	93.06	92.31	75.00	81.72	88.41	70.00	96.61	94.32	93.55	95.77	83.82	66.37	91.80	88.31	80.00	86.07
Person B	95.65	94.05	79.22	94.67	95.65	93.65	91.07	97.56	93.55	86.88	75.00	65.91	93.67	50.00	82.14	85.91
Person C	89.66	94.44	85.71	91.35	94.74	77.42	96.94	86.11	81.03	96.39	83.15	67.71	92.45	86.05	73.47	86.44
Person D	95.92	91.14	94.38	91.07	92.22	95.56	98.97	96.59	95.95	100	76.74	80.49	100	89.89	94.12	92.87
Person E	94.90	97.80	85.94	80.43	95.35	94.25	76.79	100	90.63	100	79.69	74.19	98.41	73.81	95.92	89.21
Person F	98.48	96.34	92.55	97.53	96.55	92.22	95.45	88.16	100	65.57	88.16	83.72	86.67	70.59	66.67	87.91
Average	94.61	94.35	85.47	89.46	93.82	87.18	92.64	93.79	92.45	90.77	81.09	73.07	93.83	76.44	82.05	**88.07**

FIGURE 6.9: A graphical representation of the success rate for the learning-based system on every frame

FIGURE 6.10: Comparing the success rate on whether to predict every frame or every fifth frame using a bar graph

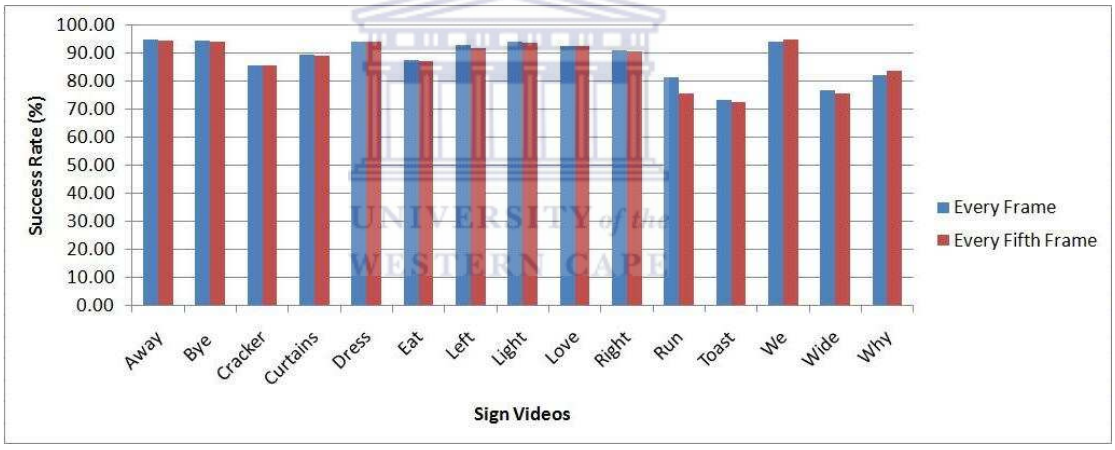
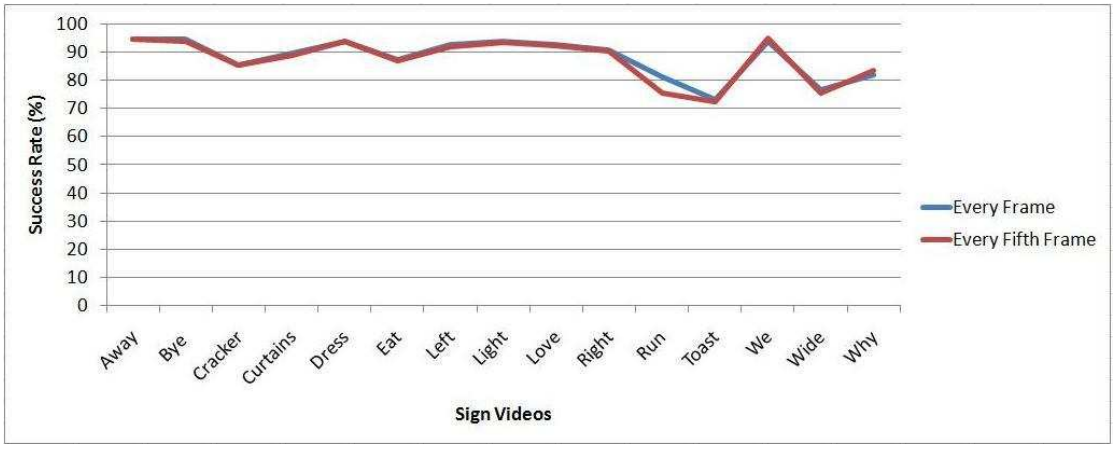


FIGURE 6.11: Comparing the success rate on whether to predict every frame or every fifth frame



From the comparison in Figure 6.11 and Figure 6.10, the success rate of the poses in each sign in both cases are very comparable. In two signs, "we" and "why", a marginally greater number of poses were correctly predicted when predicting every fifth frame; however, when predicting every frame in the video, eight signs had a marginally greater number of poses correctly predicted. Overall, predicting every frame obtained an average success rate of 88.07% and predicting every fifth frame obtained an average success rate of 87.51%. The results are very similar. Predicting every fifth frame performs five times faster than predicting every frame with a comparable success rate. Therefore, it is more advantageous to predict every fifth frame than every frame.

6.2.3 Example-Based System vs Learning-Based System

Both the example-based and learning-based systems have been shown to have very encouraging success rates. For investigative purposes, a comparison is made between the two. The example-based system follows a template-based method using the popular Chamfer Distance Transformation that has been shown to recognise body poses effectively. It has been used to adequately recognise individuals walking and a variety of body movements in related work. Additionally, a novel learning-based system is implemented to recognise upper body poses using a skin feature extraction method with SVMs.

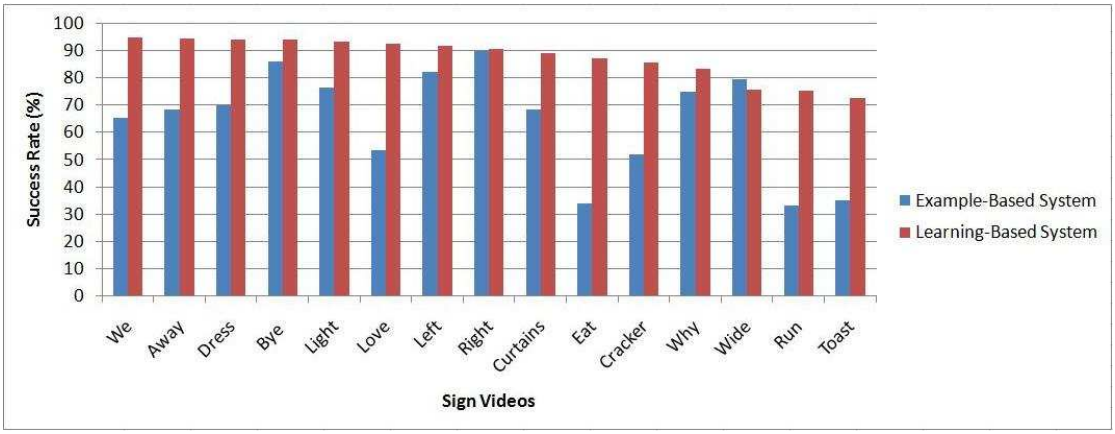
On average, the learning-based system has a success rate of 87.51%, which is greater than the example-based system, that has an average success rate of 64.49%. In Figure 6.12, a comparison of the success rates of the example-based and learning-based systems are shown. The order of sign language words has been sorted in descending order according to the success rate of the learning-based system. Analysing further, it can be seen from Figure 6.12 that the learning-based system has a low variation in the success rates across different signs with a range of 22.38% in the success rates as compared to the example-based system. The example-based system has a greater variation in the success rates across different signs with a range of 56.74% in success rates. When translating from sign language to English, it is important that the location of the wrist is accurate so that the gesture of the hand can be interpreted. For this reason, it is of greater importance that a lower variation across across different sign language words be maintained.

It was also shown that the example-based system is affected by the position of the hands in a particular pose. In specific, poses which are performed with the hands and arms in front of the torso region, have relatively lower success rates. However, this was shown to not affect the learning-based system as much. On the other hand, it was shown that the learning-based system is affected by the speed at which an individual performs sign language. This factor was shown to not affect the example-based system.

TABLE 6.11: Comparison of the example-based system against the learning-based system

Sign / Subject	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	**Average**
Example-Based System	68.25	86.05	51.99	68.28	69.66	33.94	82.05	76.51	53.40	89.74	33.00	34.98	65.11	79.44	74.95	**64.49**
Learning-Based System	94.37	93.85	85.33	88.87	93.88	86.97	91.77	93.34	92.36	90.36	75.36	72.45	94.83	75.51	83.40	**87.51**

FIGURE 6.12: Comparing the success rate of the example-based system against the learning-based system



In order to determine whether the difference between the success rates of the systems was statistically significant, chi-square and McNemar's tests were used. The results of the chi-square tests, illustrated in Table 6.12, show that there are five signs for which the differences in success rates are *not* significant. These signs are "bye", "left", "right", "wide" and "why" that each obtained a p-value above a 0.01 level of significance. Similarly, the results of the McNemar's test, illustrated in Table 6.13, show the same signs have obtained a p-value above a 0.01 level of significance. Thus, the same conclusion is drawn. The differences in success rates for these signs are *not* significant and, therefore, both systems perform equally well on those signs. Furthermore, in the other ten signs, both the chi-square and McNemar's tests have shown that the differences in success rates for those ten signs are significant.

From the comparison, it is concluded that, although the learning-based system generally performs better than the example-based system, both systems are suitable for upper body pose recognition and estimation.

6.3 Summary

In this chapter the experimental setup, including database generation, sign language data, training data and testing data were discussed.

Experimental analysis on the learning-based and example-based systems was conducted and the results show that the example-based system obtains lower success rates on poses which are performed with the hands and arms in front of the torso region. The results also show that the learning-based system is affected by the speed at which an individual performs sign language. The success rate of the kernels used in the SVM were compared and showed that the RBF kernel has a better performance. A comparison between

TABLE 6.12: Chi-square test results to determine significance of success rates between the two systems.

Sign Video	Chi-Square	P-Value
Away	27.0828	<0.0001
Bye	4.5904	0.0322
Cracker	24.0626	<0.0001
Curtains	15.8985	<0.0001
Dress	24.2667	<0.0001
Eat	59.6954	<0.0001
Left	3.2569	0.0711
Light	11.2903	0.0008
Love	33.7167	<0.0001
Right	0.0000	1.0000
Run	35.5072	<0.0001
Toast	32.1780	<0.0001
We	24.4674	<0.0001
Wide	0.2912	0.5895
Why	3.0186	0.0823

TABLE 6.13: McNemar's test results to determine significance of success rates between the two systems.

Sign Video	Label	P-Value
Away*Away	Pr > S	<0.0001
Bye*Bye	Pr > S	0.0499
Cracker*Cracker	Pr > S	<0.0001
Curtains*Curtains	Pr > S	<0.0001
Dress*Dress	Pr > S	<0.0001
Eat*Eat	Pr > S	<0.0001
Left*Left	Pr > S	0.0209
Light*Light	Pr > S	0.0016
Love*Love	Pr > S	<0.0001
Right*Right	Pr > S	0.7630
Run*Run	Pr > S	<0.0001
Toast*Toast	Pr > S	<0.0001
We*We	Pr > S	<0.0001
Wide*Wide	Pr > S	0.6949
Why*Why	Pr > S	0.1615

predicting every frame and predicting every fifth frame was also conducted and showed that these two methods produce comparable results in terms of success. It was, however, concluded that predicting every fifth frame is more advantageous since it has the added advantage of speed.

The comparison performed between the two systems has shown that although the

learning-based system generally performs better than the example-based system, both systems are suitable for upper body pose recognition and estimation.

Chapter 7

Conclusion and Directions for Future Research

Advances in technology are rapidly progressing towards eliminating the need to wear additional cumbersome equipment in human computer interaction. Recognising and estimating a human body posture using computer vision is one of them. A number of computer vision approaches have been proposed to recognize and estimate human body postures. These approaches are categorized into model-based, example-based and learning-based approaches.

In this thesis, several important contributions to the field of human body pose recognition and estimation were made. The first contribution was an example-based approach that employed a template-based matching technique. Image registration in this approach required a face detection and greyscaling method. A comparison of face detection algorithms has shown that the Viola and Jones algorithm was the most suitable one with a high face detection rate. A robust template-based matching technique is one that requires a *clean* edge detected image. This prompted a comparison of edge detection algorithms which revealed the Canny edge detection algorithm to be promising in this respect. This algorithm has good localisation and a good response to single edges. It also has a low error rate and works well on corners and curves. This system aims to find the best match in a database consisting of thousands of poses, using a good matching algorithm. The Chamfer Distance Transform, in comparison to other approximation methods, provides a closer approximation to the Euclidean distance and thereby a more accurate measurement for silhouette shape matching.

The second and main contribution of this research was a novel learning-based approach that applied feature extraction and SVMs. An attempt was made to find a suitable colour space for skin detection. Based on the research done, such a colour space does

not exist. This research, however, led to an important discovery on skin detection, which hopefully would benefit researchers in this field and similar fields alike. In order to determine the skin colour of an individual, the region around the individual's nose was used to determine the colour distribution. This distribution allowed for an instant and more accurate detection of skin-coloured pixels in an image, for an individual of any skin colour type, from the darkest skin colour to the lightest skin colour. With the skin colour diversity in South Africa, this skin detection has proved to be effective. Unfortunately, if a controlled environment is not used, false detections still occur from objects with an identical colour. These false detections were eliminated by applying an adaptive background subtraction method that continuously updates the background model. In order to aid robust features and reduce the number of unnecessary features, the opening morphological operation has shown to be effective. These features were labelled to a class and learned by an SVM. Features from a test image was later predicted and used to set keyframes on the 3D human body model to represent the predicted pose. In both approaches, the 3D human body model was extensively used to estimate the wrists, elbows and shoulders once the pose had been recognised.

Experimental analysis on both the example-based and learning-based systems was conducted. Evaluation on determining the best kernel for the learning-based system has shown that the RBF kernel with an average success rate of 86.44% was more suitable than the other kernels and therefore used for further evaluation on the system. A comparison between predicting every frame and predicting every fifth frame was also conducted and showed that these two methods produce comparable results in terms of success. It was concluded that predicting every fifth frame is more advantageous since it has the added advantage of speed.

These experiments were ultimately aimed at answering the research questions posed in this thesis. To answer these questions, both the example-based and learning-based systems are able to recognise and estimate upper body poses very well. On average, the learning-based system has a success rate of 87.51%, that is greater than the example-based system, that has an average success rate of 64.49%. From the comparison, it is concluded that, although the learning-based system generally performs better than the example-based system, both systems are suitable to recognise and estimate upper body poses in a sign language recognition and translation system.

7.1 Directions for Future Research

While different approaches to pose recognition and estimation were considered, there exist many areas for further study. One of these areas is to improve and extend both

systems to account for clothing. Clothing is an intricate aspect to handle in computer vision and it is believed from experimentation that it negatively affects both systems. It adds additional DOF to the already large dimensional search space in the example-based system and is falsely detected as features in the learning-based system when the clothes, especially the sleeves, are identical in colour to the individual's skin colour.

Another area is occlusion either by objects or other limbs. It is an area which this research has not focused on but requires some attention. It is a constant problem that many researchers in this field face [171][15][95], since 2D images do not contain the third dimensional factor. A number of recommendations are made with regard to this. There are recent developments in depth streams that use light and shadow to determine the distance of objects or limbs from the camera. Another recommendation would be to produce a distinct colour on the observed objects or limbs, using for example chroma-keying. This would allow the right and left side of the subject to be treated independently.

Experimentation in this research did not involve using different step sizes when selecting to predict the i'th frame. The number of frames available in a sign video depends on the duration of the sign and the speed at which the sign is performed. Therefore, an optimal strategy for selecting frames is required. A possible solution would be selecting the i'th frame based on the total number of frames available or by selecting frames based on how much has changed in the frame, for example, selecting a frame if the difference in frames exceeds a certain threshold.

In this research non-native SASL signers were used for experimentation. Future research could see experimentation using native SASL signers, which could highlight any similarities or differences between the two. The development of these systems did not involve any optimization. It is clear from experimentation that the learning-based system is much faster, however, there are a number of quick searching methods that could be incorporated in the example-based system. Thus, allowing for a comparison to be made in terms of speed performance. In addition, SVMs have demonstrated its effectiveness towards this problem, however, there are a number of other classifiers, for example Adaboost, that may also prove to be effective.

Finally, it would be interesting to see an evaluation of these systems on full body pose recognition and estimation.

7.2 Concluding Remarks

A final suggestion towards the SASL project is that integration of the SASL system should not only be one-way, where each subsystem works independently and sends the results to the parent system, but rather two-way, where each subsystem also uses information returned by other subsystems to improve its performance. The research conducted in this thesis has been an enormously educational experience for this researcher and it is hoped that future researchers in this field and the SASL project would benefit from this work and be equally rewarded.

Appendix A

Binomial Probability Distribution

Analysing a complete sign video is similar to a binomial experiment. To consider it as a binomial experiment, it should satisfy the following properties:

1. A single binomial experiment consists of n Bernoulli trials where $n > 1$
2. Each Bernoulli trial is independent of the other.
3. The probability of success and failure is p and q respectively, and remains the same for each trial.
4. The binomial random variable, X, is the total number of successes in an experiment.

Thus, from these properties, the binomial probability distribution of X can be computed as follows [153]:

$$Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \text{ where } k = 0, 1, 2, ..., n \text{ and } q = 1 - p \tag{A.1}$$

It follows that the mean, μ_n, and variance, σ_n^2, of the binomial experiment are equal to the sum of the mean, μ, and variance, σ^2, of each Bernoulli trial and is computed as follows [153]:

$$\mu_n = \sum_{k=1}^{n} \mu = np \tag{A.2}$$

and

$$\sigma_n = \sum_{k=1}^{n} \sigma = np(1-p) \tag{A.3}$$

A.1 Example-based system distribution

Each index is a representative of a sign following the order in table 6.1, n is the number of images per sign and x is the number of successfully matched poses. The results pertaining to the binomial experiments on this system is listed in tables A.1 - A.6

TABLE A.1: Binomial Probability Distribution for Person A in the Example-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	20	15	0.000079	0.999921	4.50618E-58	0.0016	0.0016	0.0397
Bye-Bye	25	25	0.000079	0.999921	2.74981E-103	0.0020	0.0020	0.0444
Cracker	22	18	0.000079	0.999921	1.04795E-70	0.0017	0.0017	0.0417
Curtains	22	15	0.000079	0.999921	8.08424E-17	0.0017	0.0017	0.0417
Dress	10	10	0.000079	0.999921	9.45630E-42	0.0008	0.0008	0.0281
Eat	12	5	0.000079	0.999921	2.43378E-18	0.0009	0.0009	0.0308
Left	22	19	0.000079	0.999921	1.74285E-75	0.0017	0.0017	0.0417
Light	14	14	0.000079	0.999921	3.68137E-58	0.0011	0.0011	0.0333
Love	14	10	0.000079	0.999921	9.46197E-39	0.0011	0.0011	0.0333
Right	15	15	0.000079	0.999921	2.90791E-62	0.0012	0.0012	0.0344
Run	15	7	0.000079	0.999921	1.23369E-25	0.0012	0.0012	0.0344
Toast	23	8	0.000079	0.999921	7.41993E-28	0.0018	0.0018	0.0426
We	18	16	0.000079	0.999921	3.51365E-64	0.0014	0.0014	0.0377
Wide	16	11	0.000079	0.999921	3.26106E-42	0.0013	0.0013	0.0355
Why	10	10	0.000079	0.999921	9.45630E-42	0.0008	0.0008	0.0281
Total	258	198	0.000079	0.999921	8.32762E-17	0.0203	0.0203	0.5474
Average			0.000079	0.999921	5.55174E-18	0.0014	0.0014	0.0365

The results show that the average probability of an outcome is very small. Furthermore, according to the empirical rule, on average, approximately 68% of the distribution of a successfully matched pose lies between -0.0353 and 0.0381, approximately 95% lie between -0.072 and 0.0748 and 99.7% lie between -0.1087 and 0.1115.

A.2 Learning-based system distribution

Similarly to the example-based system, the index is a representative of a sign following the same order in table 6.1, n is the number of images per sign and x is the number of successfully matched poses. The results pertaining to the binomial experiments on this system is listed in tables A.7 - A.12.

The results show a similar trend to the example-based system, where the average probability of an outcome is very small. Furthermore, according to the empirical rule, on

TABLE A.2: Binomial Probability Distribution for Person B in the Example-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	17	15	0.000079	0.999921	3.95397E-60	0.0013	0.0013	0.0366
Bye-Bye	16	11	0.000079	0.999921	3.26106E-42	0.0013	0.0013	0.0355
Cracker	16	11	0.000079	0.999921	3.26106E-42	0.0013	0.0013	0.0355
Curtains	15	11	0.000079	0.999921	1.01918E-42	0.0012	0.0012	0.0344
Dress	18	12	0.000079	0.999921	1.09465E-45	0.0014	0.0014	0.0377
Eat	14	5	0.000079	0.999921	6.15083E-18	0.0011	0.0011	0.0333
Left	12	8	0.000079	0.999921	7.49909E-31	0.0009	0.0009	0.0308
Light	19	16	0.000079	0.999921	2.22509E-63	0.0015	0.0015	0.0387
Love	13	9	0.000079	0.999921	8.55621E-35	0.0010	0.0010	0.0320
Right	13	8	0.000079	0.999921	1.94957E-30	0.0010	0.0010	0.0320
Run	19	5	0.000079	0.999921	3.57073E-17	0.0015	0.0015	0.0387
Toast	18	6	0.000079	0.999921	4.50384E-21	0.0014	0.0014	0.0377
We	16	14	0.000079	0.999921	4.41676E-56	0.0013	0.0013	0.0355
Wide	11	7	0.000079	0.999921	6.32914E-27	0.0009	0.0009	0.0295
Why	12	11	0.000079	0.999921	8.96254E-45	0.0009	0.0009	0.0308
Total	229	149	0.000079	0.999921	4.18627E-17	0.0180	0.0180	0.5187
Average			0.000079	0.999921	2.79085E-18	0.0012	0.0012	0.0346

TABLE A.3: Binomial Probability Distribution for Person C in the Example-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	16	13	0.000079	0.999921	2.60913E-51	0.0013	0.0013	0.0355
Bye-Bye	21	20	0.000079	0.999921	1.87767E-81	0.0017	0.0017	0.0407
Cracker	19	11	0.000079	0.999921	5.64110E-41	0.0015	0.0015	0.0387
Curtains	27	19	0.000079	0.999921	2.51125E-72	0.0021	0.0021	0.0462
Dress	19	12	0.000079	0.999921	2.97090E-45	0.0015	0.0015	0.0387
Eat	19	7	0.000079	0.999921	9.65630E-25	0.0015	0.0015	0.0387
Left	23	23	0.000079	0.999921	4.40716E-95	0.0018	0.0018	0.0426
Light	24	19	0.000079	0.999921	4.80931E-74	0.0019	0.0019	0.0435
Love	18	10	0.000079	0.999921	4.13458E-37	0.0014	0.0014	0.0377
Right	20	20	0.000079	0.999921	8.94216E-83	0.0016	0.0016	0.0397
Run	18	6	0.000079	0.999921	4.50384E-21	0.0014	0.0014	0.0377
Toast	25	11	0.000079	0.999921	3.32481E-39	0.0020	0.0020	0.0444
We	17	9	0.000079	0.999921	2.90795E-33	0.0013	0.0013	0.0366
Wide	18	17	0.000079	0.999921	3.26554E-69	0.0014	0.0014	0.0377
Why	16	9	0.000079	0.999921	1.36858E-33	0.0013	0.0013	0.0355
Total	300	206	0.000079	0.999921	4.50481E-21	0.0237	0.0237	0.5939
Average			0.000079	0.999921	3.00320E-22	0.0016	0.0016	0.0396

TABLE A.4: Binomial Probability Distribution for Person D in the Example-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	16	10	0.000079	0.999921	7.56806E-38	0.0013	0.0013	0.0355
Bye-Bye	21	19	0.000079	0.999921	2.37685E-76	0.0017	0.0017	0.0407
Cracker	18	5	0.000079	0.999921	2.63133E-17	0.0014	0.0014	0.0377
Curtains	23	15	0.000079	0.999921	1.42465E-56	0.0018	0.0018	0.0426
Dress	18	10	0.000079	0.999921	4.13458E-37	0.0014	0.0014	0.0377
Eat	18	4	0.000079	0.999921	1.18960E-13	0.0014	0.0014	0.0377
Left	20	20	0.000079	0.999921	8.94216E-83	0.0016	0.0016	0.0397
Light	18	13	0.000079	0.999921	3.99117E-50	0.0014	0.0014	0.0377
Love	15	4	0.000079	0.999921	5.30815E-14	0.0012	0.0012	0.0344
Right	18	18	0.000079	0.999921	1.43317E-74	0.0014	0.0014	0.0377
Run	18	6	0.000079	0.999921	4.50384E-21	0.0014	0.0014	0.0377
Toast	18	5	0.000079	0.999921	2.63133E-17	0.0014	0.0014	0.0377
We	16	7	0.000079	0.999921	2.19301E-25	0.0013	0.0013	0.0355
Wide	18	16	0.000079	0.999921	3.51365E-64	0.0014	0.0014	0.0377
Why	14	8	0.000079	0.999921	4.54854E-30	0.0011	0.0011	0.0333
Total	269	160	0.000079	0.999921	1.72094E-13	0.0212	0.0212	0.5633
Average			0.000079	0.999921	1.14730E-14	0.0014	0.0014	0.0376

TABLE A.5: Binomial Probability Distribution for Person E in the Example-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	24	15	0.000079	0.999921	3.79869E-56	0.0019	0.0019	0.0435
Bye-Bye	22	18	0.000079	0.999921	1.04795E-70	0.0017	0.0017	0.0417
Cracker	13	3	0.000079	0.999921	1.40815E-10	0.0010	0.0010	0.0320
Curtains	19	8	0.000079	0.999921	1.14424E-28	0.0015	0.0015	0.0387
Dress	18	11	0.000079	0.999921	4.39624E-30	0.0014	0.0014	0.0377
Eat	18	6	0.000079	0.999921	4.50384E-21	0.0014	0.0014	0.0377
Left	12	9	0.000079	0.999921	2.63294E-35	0.0009	0.0009	0.0308
Light	13	8	0.000079	0.999921	1.94957E-30	0.0010	0.0010	0.0320
Love	13	6	0.000079	0.999921	4.16530E-22	0.0010	0.0010	0.0320
Right	10	10	0.000079	0.999921	9.45630E-42	0.0008	0.0008	0.0281
Run	13	3	0.000079	0.999921	1.40815E-10	0.0010	0.0010	0.0320
Toast	19	6	0.000079	0.999921	6.58188E-21	0.0015	0.0015	0.0387
We	13	7	0.000079	0.999921	3.29050E-26	0.0010	0.0010	0.0320
Wide	17	14	0.000079	0.999921	2.50258E-55	0.0013	0.0013	0.0366
Why	10	6	0.000079	0.999921	5.09892E-23	0.0008	0.0008	0.0281
Total	234	130	0.000079	0.999921	2.81629E-10	0.0182	0.0182	0.5216
Average			0.000079	0.999921	1.87753E-11	0.0012	0.0012	0.0348

TABLE A.6: Binomial Probability Distribution for Person F in the Example-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	30	18	0.000079	0.999921	1.23811E-66	0.0024	0.0024	0.0487
Bye-Bye	25	20	0.000079	0.999921	4.74859E-78	0.0020	0.0020	0.0444
Cracker	19	10	0.000079	0.999921	8.72768E-37	0.0015	0.0015	0.0387
Curtains	21	19	0.000079	0.999921	2.37685E-76	0.0017	0.0017	0.0407
Dress	21	15	0.000079	0.999921	1.57700E-57	0.0017	0.0017	0.0407
Eat	18	7	0.000079	0.999921	6.09933E-25	0.0014	0.0014	0.0377
Left	14	9	0.000079	0.999921	2.39550E-34	0.0011	0.0011	0.0333
Light	21	13	0.000079	0.999921	9.47618E-49	0.0017	0.0017	0.0407
Love	11	4	0.000079	0.999921	1.28380E-14	0.0009	0.0009	0.0295
Right	13	10	0.000079	0.999921	2.70369E-39	0.0010	0.0010	0.0320
Run	17	6	0.000079	0.999921	3.00286E-21	0.0013	0.0013	0.0366
Toast	18	7	0.000079	0.999921	6.09933E-25	0.0014	0.0014	0.0377
We	15	10	0.000079	0.999921	2.83831E-38	0.0012	0.0012	0.0344
Wide	14	11	0.000079	0.999921	2.71809E-43	0.0011	0.0011	0.0333
Why	13	9	0.000079	0.999921	8.55621E-35	0.0010	0.0010	0.0320
Total	270	168	0.000079	0.999921	1.28380E-14	0.0214	0.0214	0.5604
Average			0.000079	0.999921	8.55869E-16	0.0014	0.0014	0.0374

TABLE A.7: Binomial Probability Distribution for Person A in the Learning-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	20	19	0.005952	0.994048	1.04121E-41	0.1190	0.1183	0.3440
Bye-Bye	25	23	0.005952	0.994048	1.94885E-49	0.1488	0.1479	0.3846
Cracker	22	16	0.005952	0.994048	1.78724E-31	0.1310	0.1302	0.3608
Curtains	22	18	0.005952	0.99408	6.28335E-37	0.1310	0.1302	0.3608
Dress	10	9	0.005952	0.994048	9.32400E-20	0.0595	0.0592	0.2432
Eat	12	8	0.005952	0.994048	7.61511E-16	0.0714	0.0710	0.2665
Left	22	21	0.005952	0.994048	4.05801E-46	0.1310	0.1302	0.3608
Light	14	13	0.005952	0.994048	1.63868E-28	0.0833	0.0828	0.2878
Love	14	13	0.005952	0.994048	1.63868E-28	0.0833	0.0828	0.2878
Right	15	14	0.005952	0.994048	1.04507E-30	0.0893	0.0888	0.2979
Run	15	12	0.005952	0.994048	8.84012E-25	0.0893	0.0888	0.2979
Toast	23	15	0.005952	0.994048	1.94951E-28	0.1369	0.1361	0.3689
We	18	17	0.005952	0.994048	2.64484E-37	0.1071	0.1065	0.3264
Wide	16	14	0.005952	0.994048	8.31042E-30	0.0952	0.0947	0.3077
Why	10	8	0.005952	0.994048	7.00665E-17	0.0595	0.0592	0.2432
Total	258	220	0.005952	0.994048	8.31670E-16	1.5356	1.5267	4.7383
Average			0.005952	0.994048	5.54447E-17	0.1024	0.1018	0.3159

TABLE A.8: Binomial Probability Distribution for Person B in the Learning-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	17	16	0.005952	0.994048	4.19649E-35	0.1012	0.1006	0.3172
Bye-Bye	16	15	0.005952	0.994048	6.63538E-33	0.0952	0.0947	0.3077
Cracker	16	13	0.005952	0.994048	6.47628E-27	0.0952	0.0947	0.3077
Curtains	15	14	0.005952	0.994048	1.04507E-30	0.0893	0.0888	0.2979
Dress	18	17	0.005952	0.994048	2.64484E-37	0.1071	0.1065	0.3264
Eat	14	13	0.005952	0.994048	1.63868E-28	0.0833	0.0828	0.2878
Left	12	11	0.005952	0.994048	3.96428E-24	0.0714	0.0710	0.2665
Light	19	18	0.005952	0.994048	1.66177E-39	0.1131	0.1124	0.3353
Love	13	12	0.005952	0.994048	2.55633E-26	0.0774	0.0769	0.2773
Right	13	11	0.005952	0.994048	2.56132E-23	0.0774	0.0769	0.2773
Run	19	14	0.005952	0.994048	7.90872E-28	0.1131	0.1124	0.3353
Toast	18	12	0.005952	0.994048	3.54224E-23	0.1071	0.1065	0.3264
We	16	15	0.005952	0.994048	6.63538E-33	0.0952	0.0947	0.3077
Wide	11	5	0.005952	0.994048	3.32976E-09	0.0655	0.0651	0.2551
Why	12	10	0.005952	0.994048	3.64102E-21	0.0714	0.0710	0.2665
Total	229	196	0.005952	0.994048	3.32976E-09	1.3629	1.3550	4.4921
Average			0.005952	0.994048	2.21984E-10	0.0909	0.0903	0.2995

TABLE A.9: Binomial Probability Distribution for Person C in the Learning-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	16	14	0.005952	0.994048	8.31042E-30	0.0952	0.0947	0.3077
Bye-Bye	21	20	0.005952	0.994048	6.50757E-44	0.1250	0.1243	0.3525
Cracker	19	16	0.005952	0.994048	2.36338E-33	0.1131	0.1124	0.3353
Curtains	27	25	0.005952	0.994048	8.07879E-54	0.1607	0.1598	0.3997
Dress	19	18	0.005952	0.994048	1.66177E-39	0.1131	0.1124	0.3353
Eat	19	15	0.005952	0.994048	1.57866E-30	0.1131	0.1124	0.3353
Left	23	22	0.005952	0.994048	2.52527E-48	0.1369	0.1361	0.3689
Light	24	21	0.005952	0.994048	3.68870E-44	0.1429	0.1420	0.3768
Love	18	15	0.005952	0.994048	3.34356E-31	0.1071	0.1065	0.3264
Right	20	19	0.005952	0.994048	1.04121E-41	0.1190	0.1183	0.3440
Run	18	10	0.005952	0.994048	2.32839E-18	0.1071	0.1065	0.3264
Toast	25	17	0.005952	0.994048	1.52366E-32	0.1488	0.1479	0.3846
We	17	16	0.005952	0.994048	4.19649E-35	0.1012	0.1006	0.3172
Wide	18	15	0.005952	0.994048	3.34356E-31	0.1071	0.1065	0.3264
Why	16	12	0.005952	0.994048	3.51483E-24	0.0952	0.0947	0.3077
Total	300	255	0.005952	0.994048	2.32839E-18	1.7855	1.7751	5.1442
Average			0.005952	0.994048	1.55226E-19	0.1190	0.1183	0.3429

TABLE A.10: Binomial Probability Distribution for Person D in the Learning-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	16	15	0.005952	0.994048	6.63538E-33	0.0952	0.0947	0.3077
Bye-Bye	21	19	0.005952	0.994048	1.08671E-40	0.1250	0.1243	0.3525
Cracker	18	17	0.005952	0.994048	2.64484E-37	0.1071	0.1065	0.3264
Curtains	23	21	0.005952	0.994048	4.63871E-45	0.1369	0.1361	0.3689
Dress	18	17	0.005952	0.994048	2.64484E-37	0.1071	0.1065	0.3264
Eat	18	17	0.005952	0.994048	2.64484E-37	0.1071	0.1065	0.3264
Left	20	20	0.005952	0.994048	3.11755E-45	0.1190	0.1183	0.3440
Light	18	17	0.005952	0.994048	2.64484E-37	0.1071	0.1065	0.3264
Love	15	14	0.005952	0.994048	1.04507E-30	0.0893	0.0888	0.2979
Right	18	18	0.005952	0.994048	8.79897E-41	0.1071	0.1065	0.3264
Run	18	14	0.005952	0.994048	2.09380E-28	0.1071	0.1065	0.3264
Toast	18	14	0.005952	0.994048	2.09380E-28	0.1071	0.1065	0.3264
We	16	16	0.005952	0.994048	2.48342E-36	0.0952	0.0947	0.3077
Wide	18	16	0.005952	0.994048	3.75418E-34	0.1071	0.1065	0.3264
Why	14	13	0.005952	0.994048	1.63868E-28	0.0833	0.0828	0.2878
Total	269	248	0.005952	0.994048	5.83680E-28	1.6007	1.5917	4.8777
Average			0.005952	0.994048	3.89120E-29	0.1067	0.1061	0.3252

TABLE A.11: Binomial Probability Distribution for Person E in the Learning-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	24	23	0.005952	0.994048	1.56849E-50	0.1429	0.1420	0.3768
Bye-Bye	22	22	0.005952	0.994048	1.10457E-49	0.1310	0.1302	0.3608
Cracker	13	11	0.005952	0.994048	2.56132E-23	0.0774	0.0769	0.2773
Curtains	19	15	0.005952	0.994048	1.57866E-30	0.1131	0.1124	0.3353
Dress	18	17	0.005952	0.994048	2.64484E-37	0.1071	0.1065	0.3264
Eat	18	17	0.005952	0.994048	2.64484E-37	0.1071	0.1065	0.3264
Left	12	9	0.005952	0.994048	2.02674E-18	0.0714	0.0710	0.2665
Light	13	13	0.005952	0.994048	1.17755E-29	0.0774	0.0769	0.2773
Love	13	12	0.005952	0.994048	2.55633E-26	0.0774	0.0769	0.2773
Right	10	10	0.005952	0.994048	5.58350E-23	0.0595	0.0592	0.2432
Run	13	10	0.005952	0.994048	1.56831E-20	0.0774	0.0769	0.2773
Toast	19	14	0.005952	0.994048	7.90872E-28	0.1131	0.1124	0.3353
We	13	13	0.005952	0.994048	1.17755E-29	0.0774	0.0769	0.2773
Wide	17	13	0.005952	0.994048	2.73590E-26	0.1012	0.1006	0.3172
Why	10	10	0.005952	0.994048	5.58350E-23	0.0595	0.0592	0.2432
Total	234	209	0.005952	0.994048	2.04256E-18	1.3929	1.3845	4.5176
Average			0.005952	0.994048	1.36171E-19	0.0929	0.0923	0.3012

TABLE A.12: Binomial Probability Distribution for Person F in the Learning-Based System

Signs	**n**	**x**	**p**	**q**	**Pr: x = n**	**Mean**	**Variance**	**Standard Deviation**
Away	30	30	0.005952	0.994048	1.74068E-67	0.1786	0.1775	0.4213
Bye-Bye	25	24	0.005952	0.994048	9.72528E-53	0.1488	0.1479	0.3846
Cracker	19	18	0.005952	0.994048	1.66177E-39	0.1131	0.1124	0.3353
Curtains	21	20	0.005952	0.994048	6.50757E-44	0.1250	0.1243	0.3525
Dress	21	20	0.005952	0.994048	6.50757E-44	0.1250	0.1243	0.3525
Eat	18	17	0.005952	0.994048	2.64484E-37	0.1071	0.1065	0.3264
Left	14	13	0.005952	0.994048	1.63868E-28	0.0833	0.0828	0.2878
Light	21	19	0.005952	0.994048	1.08671E-40	0.1250	0.1243	0.3525
Love	11	11	0.005952	0.994048	3.32351E-25	0.0655	0.0651	0.2551
Right	13	9	0.005952	0.994048	6.54738E-18	0.0774	0.0769	0.2773
Run	17	15	0.005952	0.994048	5.60624E-32	0.1012	0.1006	0.3172
Toast	18	15	0.005952	0.994048	3.34356E-31	0.1071	0.1065	0.3264
We	15	13	0.005952	0.994048	1.22163E-27	0.0893	0.0888	0.2979
Wide	14	10	0.005952	0.994048	5.45615E-20	0.0833	0.0828	0.2878
Why	13	9	0.005952	0.994048	6.54738E-18	0.0774	0.0769	0.2773
Total	270	243	0.00595238	0.994048	1.31493E-17	1.6071	1.5976	4.8519
Average			0.00595238	0.994048	8.76621E-19	0.1071	0.1065	0.3235

average, approximately 68% of the distribution of a successfully matched pose lie between -0.2148 and 0.4212, approximately 95% lie between -0.5328 and 0.7392 and 99.7% lie between -0.8508 and 1.0572.

Bibliography

[1] I. Achmed and J. Connan, "A panoramic video system," in *Proceedings of the Southern Africa Telecommunication Networks and Applications Conference*. SATNAC, 2009, pp. 279–284.

[2] I. Achmed and J. Connan, "Upper body pose estimation towards the translation of South African sign language," in *Proceedings of the Southern Africa Telecommunication Networks and Applications Conference*. SATNAC, 2010, pp. 427–432.

[3] A. Agarwal and B. Triggs, "3D human pose from silhouettes by relevance vector regression," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2004, pp. 882–888.

[4] A. Agarwal and B. Triggs, "Monocular human motion capture with a mixture of regressors," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 3. IEEE Computer Society, 2005, p. 72.

[5] A. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, 2005.

[6] A. Agarwal and B. Triggs, "A local basis representation for estimating human pose from cluttered images," *in Proceedings of the Asian Conference on Computer Vision*, pp. 50–59, 2006.

[7] U. Ahlvers, R. Rajagopalan, and U. Zölzer, "Model-free face detection and head tracking with morphological hole mapping," in *Proceedings of the European Signal Processing Conference*, 2005.

[8] O. Aran, I. Ari, L. Akarun, B. Sankur, A. Benoit, A. Caplier, P. Campr, A. H. Carrillo, and F. Fanard, "Signtutor: An interactive system for sign language tutoring," *IEEE Multimedia*, vol. 16, pp. 81–93, 2009.

[9] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios, "Boostmap: A method for efficient approximate similarity rankings," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2004, pp. 268–275.

[10] V. Athitsos and S. Sclaroff, "Inferring body pose without tracking body parts," in *International Conference on Computer Vision and Pattern Recognition*, 2000.

[11] V. Athitsos and S. Sclaroff, "Estimating 3D hand pose from a cluttered image," in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2003.

[12] D. Bailey, "An efficient Euclidean distance transform," in *Proceedings of the Combinatorial Image Analysis: 10th International Workshop, IWCIA 2004*. Springer-Verlag New York Inc, 2004, p. 394.

[13] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 509–522, 2002.

[14] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 849–865, 1988.

[15] B. Boulay, "Human posture recognition for behaviour understanding," PhD Dissertation, University of Nice Sophia Antipolis, 2007.

[16] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.

[17] J. Brand and J. Mason, "A comparative assessment of three approaches to pixel-level human skin-detection," in *Proceedings of the 15th International Conference on Pattern Recognition*, 2000, pp. 1056–1059.

[18] M. Brand, "Shadow puppetry," in *Proceedings of the 7th IEEE International Conference on Computer Vision*, vol. 2. IEEE, 1999, pp. 1237–1244.

[19] J. Canny, "Computational theory of edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[20] H. Cao, N. Ohnishi, Y. Takeuchi, T. Matsumoto, and H. Kudo, "Fast human pose retrieval using approximate chamfer distance," *Transactions of the Institute of Electrical Engineers of Japan*, vol. 126, no. 12, pp. 1490–1496, 2006.

[21] H. cao, "Example-based methods for estimating 3d human pose from silhouette image using approximate chamfer distance and kernel subspace," PhD Dissertation, Nagoya University, March 2007.

[22] C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, 2001, [Online] Software Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[23] D. Chen, P. Chou, C. B. Fookes, and S. Sridharan, "Multi-view human pose estimation using modified five-point skeleton model," *International Conference on Signal Processing and Communication Systems*, December 2008.

[24] J. Chen, P. Tan, and T. Goh, "Refinement to the chamfer matching for a 'center-on' fit," *Image and Vision Computing*, pp. 360–365, 2003.

[25] Q. Chen, E. Zheng, and Y. Liu, "Pose estimation based on human detection and segmentation," *Science in China Series F: Information Sciences*, vol. 52, no. 2, pp. 244–251, 2009.

[26] K. Cho, J. Jang, and K. Hong, "Adaptive skin-color filter," *Pattern Recognition*, vol. 34, no. 5, pp. 1067–1073, 2001.

[27] N. Cristianini, "Support vector and kernel machines," *Tutorial at ICML*, 2001.

[28] O. Cuisenaire, "Distance transformations: Fast algorithms and applications to medical image processing," PhD Dissertation, Communications and Remote Sensing Laboratory, Catholic University of Louvain, Belgium, October 1999.

[29] F. Dadgostar and A. Sarrafzadeh, "A fast real-time skin detector for video sequences," *Image Analysis and Recognition*, pp. 804–811, 2005.

[30] F. Dadgostar and A. Sarrafzadeh, "An adaptive real-time skin detector based on Hue thresholding: A comparison on two motion tracking methods," *Pattern Recognition Letters*, vol. 27, no. 12, pp. 1342–1352, 2006.

[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 886–893.

[32] J. Dattorro, *Convex optimization & Euclidean distance geometry*. Meboo Publishing USA, 2005.

[33] DeafSA, "Deaf federation of south africa," January 2010, [Online] Available at http://www.deafsa.co.za/index-2.html.

[34] S. Deane, "A Comparison of background subtraction techniques," IRP Report, 2007.

[35] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision Pattern Recognition*, vol. 2, 2000, pp. 126–133.

[36] A. Elgammal and C. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2004.

[37] A. Elgammal, C. Muang, and D. Hu, "Skin detection-A short tutorial," *Encyclopedia of Biometrics, Verlag*, 2009.

[38] P. Felzenszwalb and D. Huttenlocher, "Efficient matching of pictorial structures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2000, pp. 66–73.

[39] M. Filhol, "Zebedee: A lexical description model for sign language synthesis," *LIMSI Report*, 2009.

[40] R. Fisher, S. Perkins, A. Walker, and E. Wolfart, *Hypermedia image processing reference*. Wiley, 1996.

[41] M. Fleck, D. Forsyth, and C. Bregler, "Finding naked people," in *Proceedings of the 4th European Conference on Computer Vision*, vol. 2. Springer-Verlag, 1996, pp. 593–602.

[42] J. Friedman, "Another approach to polychotomous classification," Department of Statistics, Stanford University, Technical Report, 1996, [Online] Available at http://www.stat.stanford.edu/reports/friedman/poly.ps.z.

[43] A. Ghafoor, R. Iqbal, and S. Khan, "Image matching using distance transform," in *Proceedings of the 13th Scandinavian Conference on Image Analysis*. Springer-Verlag, 2003, pp. 654–660.

[44] M. Ghaziasgar, "The use of mobile phones as service-delivery devices in a sign language machine translation system," Master's Thesis, University of the Western Cape, Computer Science, June 2010.

[45] M. Glaser and W. Tucker, "Telecommunications bridging between Deaf and hearing users in South Africa," *in Proceedings of the Conference and Workshop on Assistive Technologies for People with Vision and Hearing Impairments*, 2004.

[46] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–6.

[47] R. Gordon Jr, "Ethnologue: Languages of the world," *[Online] Available at http://www.ethnologue.com*, 2005.

[48] K. Grauman, G. Shakhnarovich, and T. Darrell, "Inferring 3D structure with a statistical image-based shape model," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 641–647.

[49] K. Grochow, S. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 522–531, 2004.

[50] A. Gupta, A. Mittal, and L. Davis, "Constraint integration for efficient multiview pose estimation with self-occlusions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 493–506, 2007.

[51] S. Gupta, "Support vector machines based modelling of concrete strength," *International Journal of Computer Systems Science and Engineering*, vol. 3, no. 1, 2008.

[52] R. Hassanpour, A. Shahbahrami, and S. Wong, "Adaptive Gaussian mixture model for skin color segmentation," in *Proceedings of the World Academy of Science, Engineering and Technology*, vol. 31. Citeseer, 2008, pp. 1–6.

[53] K. Hayashi, L. Heng, and V. Srivastava, "Pose estimation from occluded images," Machine Learning Project, August 2006.

[54] L. He, "Generation of human body models," Master's Thesis, The University of Auckland, April 2005.

[55] S. Howard, *Finger talk-South African sign language dictionary*. South Africa: Mondi, 2008.

[56] N. Howe, M. Leventon, and W. Freeman, "Bayesian reconstruction of 3D human motion from single-camera video," in *Neural Information Processing Systems*, vol. 1. Citeseer, 1999.

[57] C. Hsu, C. Chang, and C. Lin, "A practical guide to support vector classification," National Taiwan University, Technical Report, 2003.

[58] C. Hsu and C. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[59] R. Hsu, M. Abdel-Mottaleb, and A. Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696–706, 2002.

[60] Z. Hu, G. Wang, X. Lin, and H. Yan, "Recovery of upper body poses in static images based on joints detection," *Pattern Recognition Letters*, vol. 30, no. 5, pp. 503–512, 2009.

[61] G. Hua, M. Yang, and Y. Wu, "Learning to estimate human pose with data driven belief propagation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE Computer Society, 2005, pp. 747–754.

[62] H. Huang, Z. Liang, and P. Pardalos, "Some properties for the Euclidean distance matrix and positive semidefinite matrix completion problems," *Journal of Global Optimization*, vol. 25, no. 1, pp. 3–21, 2003.

[63] M. Huenerfauth, "Generating american sign language classifier predicates for English-to-ASL machine translation," PhD Dissertation, University of Pennsylvania Philadelphia, PA, USA, 2006.

[64] E. Hunter, "Visual estimation of articulated motion using the expectation-constrained maximization algorithm," PhD Dissertation, University of California, San Diego, 1999.

[65] S. Ioffe and D. Forsyth, "Probabilistic methods for finding people," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 45–68, 2001.

[66] T. Jaeggli, E. Koller-Meier, and L. Van Gool, "Learning generative models for monocular body pose estimation," in *Proceedings of the 8th Asian conference on Computer vision*, vol. 1. Springer-Verlag, 2007, pp. 608–617.

[67] M. Jones and J. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, 2002.

[68] S. Ju, M. Black, and Y. Yacoob, "Cardboard people: A parameterized model of articulated image motion," in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*. IEEE, 1996, pp. 38–44.

[69] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern Recognition*, vol. 40, no. 3, pp. 1106–1122, 2007.

[70] T. Karrels, "Fourier descriptors: Properties and utility in leaf classification," Digital Image Processing Project, December 2006.

[71] S. Keerthi and C. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Computation*, vol. 15, no. 7, pp. 1667–1689, 2003.

[72] W. Kelly, A. Donnellan, and D. Molloy, "A review of skin detection techniques for objectionable images," *IT&T 2007 General Chairs Letter*, p. 40, 2007.

[73] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, vol. 14. Citeseer, 1995, pp. 1137–1145.

[74] B. Laxton, "Monocular human pose estimation," Research Examination for the Master of Science degree at University of California, San Diego, 2007.

[75] M. Lee and I. Cohen, "Human upper body pose estimation in static images," *in Proceedings of the European Conference on Computer Vision*, pp. 126–138, 2004.

[76] M. Lee and I. Cohen, "A model-based approach for estimating human 3D poses in static images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 905–916, 2006.

[77] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE Computer Society, 2005, pp. 878–885.

[78] J. Li, "A wavelet approach to edge detection," Master's Thesis, Sam Houston State University, August 2003.

[79] L. Li, W. Huang, I. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proceedings of the Eleventh ACM international conference on Multimedia*. ACM, 2003, p. 10.

[80] D. Lim, "Design of a vision system for an autonomous underwater vehicle," Master's Thesis, Electronic and Computer Engineering, University of Western Australia, November 2004.

[81] H. Lin and C. Lin, "A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods," Department of Computer Science and Information Engineering, National Taiwan University, Technical Report, March 2003, [Online] Available at http://www.csie.ntu.edu.tw/cjlin/papers/tanh.pdf.

[82] Y. Lin, H. Tseng, and C. Fuh, "Pornography detection using support vector machine," in *Proceedings of the 16th IPPR Conference on Computer Vision, Graphics and Image Processing*, 2003, pp. 123–130.

[83] H. Liu, Y. Wang, and X. Lu, "A method to choose kernel function and its parameters for support vector machines," in *Proceedings of the 2005 International Conference on Machine Learning and Cybernetics*, vol. 7. IEEE, 2005, pp. 4277–4280.

[84] L. Louw, "Automated face detection and recognition for a login system," Master's Thesis, University of Stellenbosch, March 2007.

[85] N. Lu, J. Wang, Q. Wu, and L. Yang, "An improved motion detection method for real-time surveillance," *IAENG International Journal of Computer Science*, vol. 35, no. 1, pp. 119–128, 2008.

[86] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *International Journal of Computer Vision*, vol. 39, no. 1, pp. 57–71, 2000.

[87] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proceedings of the 6th European Conference on Computer Vision*, vol. 2. Springer-Verlag, 2000, pp. 3–19.

[88] G. Madzarov, D. Gjorgjevikj, and I. Chorbev, "A multi-class SVM classifier utilizing binary decision tree," *Informatica: An International Journal of Computing and Informatics*, vol. 33, no. 2, pp. 225–233, 2009.

[89] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *International Journal of Image Processing*, vol. 3, no. 1, p. 1, 2009.

[90] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, pp. 187–217, 1980.

[91] A. Maruch, "Talking with the hearing-impaired," [Online] Available at http://rule6.info/hearing.html, February 2010.

[92] O. Mazanỳ, "Articulated 3D human model and its animation for testing and learning algorithms of multi-camera systems," Master's Thesis, Center for Machine Perception, Czech Technical University, January 2007.

[93] A. McIvor, "Background subtraction techniques," in *Proceedings of the Image and Vision Computing New Zealand*. Citeseer, 2000, pp. 147–153.

[94] A. Micilotta, E. Ong, and R. Bowden, "Real-time upper body detection and 3D pose estimation in monoscopic images," *European Conference on Computer Vision–ECCV 2006*, pp. 139–150, 2006.

[95] A. Micilotta, "Detection and tracking of humans for visual interaction," PhD Dissertation, University of Surrey, September 2005.

[96] A. Micilotta, E. Ong, and R. Bowden, "Real-time upper body 3D pose estimation from a single uncalibrated camera," *in Proceedings of Eurographics, Short Presentations*, pp. 41–44, August 2005.

[97] I. Mikić, "Human body model acquisition and tracking using multi-camera voxel data," PhD Dissertation, University of California, San Diego, 2003.

[98] I. Mikić, M. Trivedi, E. Hunter, and P. Cosman, "Human body model acquisition and tracking using voxel data," *International Journal of Computer Vision*, vol. 53, no. 3, pp. 199–223, 2003.

[99] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," *European Conference on Computer Vision-ECCV 2004*, pp. 69–82, 2004.

[100] V. Mittal, "Edge detection technique using fuzzy logic," Master's Thesis, Thapar University, Patiala, May 2008.

[101] T. B. Moeslund and E. Granum, "A survey of computer vision based human motion capture," *Computer Vision and Image Understanding*, vol. 81, pp. 231–268, March 2001.

[102] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 349–361, 2002.

[103] M. Mojarrad, A. Rahmani, and M. Mohebi, "Detection and pose estimation of people in images," *World Academy of Science, Engineering and Technology*, pp. 985–990, 2009.

[104] A. Moore, "The case for approximate distance transforms," *University of Otago, Dunedin, Presented at SIRC*, 2002.

[105] G. Mori and J. Malik, "Estimating human body configurations using shape context matching," in *Proceedings of the 7th European Conference on Computer Vision-Part III*, vol. 3. Springer-Verlag, 2002, pp. 666–680.

[106] G. Mori and J. Malik, "Recovering 3D human body configurations using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1052–1062, 2006.

[107] G. Mori, X. Ren, A. Efros, and J. Malik, "Recovering human body configurations: Combining segmentation and recognition," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 326–333, 2004.

[108] V. Nabiyev and A. Günay, "Towards a biometric purpose image filter according to skin detection," in *The Second International Conference on Problems of Cybernetics and Informatics*, 2008, pp. 1–4.

[109] E. Nadernejad, "Edge detection techniques: Evaluation and comparisons ," *Applied Mathematical Sciences*, vol. 2, no. 31, pp. 1507–1520, 2008.

[110] N. Naidoo, "South african sign language recognition using feature vectors and hidden markov models," Master's Thesis, University of the Western Cape, Computer Science, December 2009.

[111] K. Nallaperumal, S. Ravi, C. N. K. Babu, R. K. Selvakumar, A. L. Fred, S. Christopher, and S. S. Vinsley, "Skin detection using color pixel classification with application to face detection: A comparative study," in *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications*, vol. 3. IEEE Computer Society, 2007, pp. 436–441.

[112] R. Navaratnam, A. Thayananthan, P. Torr, and R. Cipolla, "Hierarchical part-based human body pose estimation," in *Proceedings of the British Machine Vision Conference*, vol. 1. Citeseer, September 2005, pp. 479–488.

[113] W. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[114] C. Northway, "Real-time traffic sign detection using hierarchical distance matching," Thesis, School of Information Technology and Electrical Engineering, The University of Queensland, October 2002.

[115] R. Okada and S. Soatto, "Relevant feature selection for human pose estimation and localization in cluttered images," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, vol. 2. Springer-Verlag, 2008, pp. 434–445.

[116] B. Örten, "Moving Object Identification and Event Recogition in Video Surveillance Systems," PhD Dissertation, Middle East Technical University, 2005.

[117] V. Parameswaran and R. Chellappa, "View independent human body pose estimation from a single perspective image," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2004, pp. 16–22.

[118] P. Peer and F. Solina, "An automatic human face detection method," in *Proceedings of the Computer Vision Winter Workshop*. Citeseer, 1999, pp. 122–130.

[119] M. Piccardi, "Background subtraction techniques: a review," in *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4. Ieee, 2005, pp. 3099–3104.

[120] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," *Advances in Neural Information Processing Systems*, vol. 12, no. 3, pp. 547–553, 2000.

[121] R. Poppe, "Evaluating example-based pose estimation: Experiments on the humaneva sets," in *Computer Vision and Pattern Recognition workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM2)*. Citeseer, 2007.

[122] R. Poppe and M. Poel, "Example-based pose estimation in monocular images using compact fourier descriptors," Centre for Telematics and Information Technology, University of Twente, Technical Report TR-CTIT-05-49, October 2005.

[123] C. Rajah, "Chereme-based recognition of isolated, dynamic gestures from south african sign language with hidden markov models," Master's Thesis, University of the Western Cape, Computer Science, January 2006.

[124] J. Relethford, "Human skin color diversity is highest in sub-Saharan African populations." *Human biology; An International Record of Research*, vol. 72, no. 5, pp. 773–780, October 2000.

[125] L. Ren, G. Shakhnarovich, J. Hodgins, H. Pfister, and P. Viola, "Learning silhouette features for control of human motion," in *ACM SIGGRAPH 2004 Sketches*. ACM, 2004, p. 129.

[126] X. Ren, A. Berg, and J. Malik, "Recovering human body configurations using pairwise constraints between parts," in *Tenth IEEE International Conference on Computer Vision*, vol. 1. IEEE, 2005, pp. 824–831.

[127] T. Roberts, S. McKenna, and I. Ricketts, "Human pose estimation using learnt probabilistic region similarities and partial configurations," *European Conference on Computer Vision-ECCV 2004*, pp. 291–303, 2004.

[128] R. Ronfard, C. Schmid, and B. Triggs, "Learning to parse pictures of people," in *Proceedings of the 7th European Conference on Computer Vision-Part IV*, vol. 4. Springer-Verlag, 2002, pp. 700–714.

[129] R. Rosales, "Specialized mappings architecture with applications to vision-based estimation of articulated body pose," PhD Dissertation, Boston University, January 2002.

[130] R. Rosales and S. Sclaroff, "Specialized mappings and the estimation of human body pose from a single image," in *Proceedings of the Workshop on Human Motion*. IEEE Computer Society, 2000, p. 19.

[131] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[132] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 2002.

[133] D. Rybach, "Appearance-based features for automatic continuous sign language recognition," Master's Thesis, RWTH Aachen University, Aachen, Germany, Human Language Technology and Pattern Recognition Group, June 2006.

[134] J. Schneider, "Sign language expression: The importance of gestures and facial expressions," [Online] Available at http://www.essortment.com/lifestyle/signlanguageex_shrn.htm, January 2010.

[135] V. Segers, "The efficacy of the eigenvector approach to south african sign language identification," Master's Thesis, University of the Western Cape, Computer Science, December 2009.

[136] S. Sen-ching and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Proceedings of the SPIE*, vol. 5308, 2004, p. 881.

[137] N. Seo, "A comparison of multi-class support vector machine methods for face recognition," Department of Electrical and Computer Engineering, The University of Maryland, Project, December 2007.

[138] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 750–757.

[139] S. Shelly and J. Schneck, *The complete idiot's guide to learning sign language*. Alpha Books, 1998.

[140] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2002.

[141] F. Shih and Y. Wu, "Fast Euclidean distance transformation in two scans using a 3× 3 neighborhood," *Computer Vision and Image Understanding*, vol. 93, no. 2, p. 205, 2004.

[142] M. Shin, K. Chang, and L. Tsap, "Does colorspace transformation make any difference on skin detection?" in *Proceedings of the 6th IEEE Workshop on Applications of Computer Vision*. IEEE Computer Society, 2002, pp. 275–279.

[143] L. Sigal, M. Isard, B. Sigelman, and M. Black, "Attractive people: Assembling loose-limbed models using non-parametric belief propagation," *Advances in Neural Information Processing System*, vol. 16, 2004.

[144] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Discriminative density propagation for 3D human motion estimation," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE Computer Society, 2005, pp. 390–397.

[145] C. Sminchisescu and A. Telea, "Human pose estimation from silhouettes. A consistent approach using distance level sets," in *WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, vol. 1. Citeseer, 2002.

[146] C. Sminchisescu and B. Triggs, "Estimating articulated human motion with covariance scaled sampling," *The International Journal of Robotics Research*, vol. 22, no. 6, p. 371, 2003.

[147] P. Srinivasan and J. Shi, "Bottom-up recognition and parsing of the human body," in *Proceedings of the 6th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer-Verlag, 2007, pp. 153–168.

[148] Start-American-Sign-Language, "Deaf culture," [Online] Available at http://www.start-american-sign-language.com/deaf-culture.html, February 2010.

[149] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, June 1999, pp. 246–252.

[150] B. Stenger, "Model-based hand tracking using a hierarchical bayesian filter," PhD Dissertation, University of Cambridge, March 2004.

[151] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla, "Filtering using a tree-based estimator," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 1063–1070.

[152] W. Stokoe Jr, "Sign language structure: An outline of the visual communication systems of the American deaf," *Journal of Deaf Studies and Deaf Education*, vol. 10, no. 1, p. 3, 2005.

[153] N. Strawn, "Bernoulli trials and the binomial distribution," Course Notes, 2008.

[154] J. Sullivan and S. Carlsson, "Recognizing and tracking human action," *European Conference on Computer VisionECCV 2002*, pp. 629–644, 2002.

[155] T. Tangkuampien and D. Suter, "Real-time human pose inference using kernel principal component pre-image approximations," in *Proceedings of the British Machine Vision Conference*, vol. 2. Citeseer, 2006, pp. 599–608.

[156] C. Taylor, "Reconstruction of articulated objects from point correspondences in a single uncalibrated image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2000, pp. 677–684.

[157] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla, "Pose estimation and tracking using multivariate regression," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1302–1310, 2008.

[158] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE Comput. Soc, June 2003, pp. 127–133.

[159] S. Theodoridis and K. Koutroumbas, *Pattern recognition*, 2nd ed. Academic Press-Elsevier, 2003.

[160] M. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, vol. 1, no. 3. Citeseer, 2003.

[161] A. Tzotsos, "A support vector machine approach for object based image analysis," *in Proceedings of the 1st International Conference on Object-based Image Analysis*, 2006.

[162] M. Van der Schyff, "Bandwidth efficient virtual classroom," Master's Thesis, University of Johannesburg, May 2005.

[163] D. Van Wyk, "Virtual human modelling and animation for sign language visualisation," Master's Thesis, University of the Western Cape, Computer Science, December 2008.

[164] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proceedings of the Graphicon*, vol. 85, no. 1. Citeseer, 2003, pp. 85–92.

[165] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, vol. 1. Citeseer, December 2001, pp. 511–518.

[166] J. Whitehill, "Automatic real-time facial expression recognition for signed language translation," Master's Thesis, University of the Western Cape, Computer Science, May 2006.

[167] J. Yang and A. Waibel, "A real-time face tracker," in *Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision*. IEEE Computer Society, 1996, pp. 142–147.

[168] L. Yi, "KernTune: Self-tuning linux kernel performance using support vector machines," Master's Thesis, The University of the Western Cape, November 2006.

[169] B. Zarit, B. Super, and F. Quek, "Comparison of five color models in skin pixel classification," in *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*. IEEE Computer Society, 1999, pp. 58–63.

[170] J. Zhang, J. Luo, R. Collins, and Y. Liu, "Body localization in still images using hierarchical models and hybrid search," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2006, pp. 1536–1543.

[171] Y. Zhu, "Model-based human pose estimation with spatio-temporal inferencing," PhD Dissertation, The Ohio State University, 2009.