Principal Component Pyramids using Image Blurring for Nonlinearity Reduction in Hand Shape Recognition

By

Mohamed Farouk Kheir Eldin, M.Sc.

A dissertation submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy (Ph.D)

Dublin City University

Faculty of Engineering and Computing, School of Computing

Supervisors:

Dr. Alistair Sutherland

Prof. Amin Shoukry (Alexandria University, Egypt)

January 2015

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Mohamed Farouk.

(Candidate) ID No.: 57112223

Date: 19/01/2015

Acknowledgement

I would like to thank my family for their love and support all the time. I want to express my deepest appreciation to my supervisors Dr. Alistair Sutherland and Prof. Amin Shoukry for their guidance, support, keen interest and valuable advice which ultimately made my Ph.D thesis. It was a great opportunity and a good experience to make my study in Dublin City University. I will always remember these days I spent in Ireland. I also would like to thank all my colleagues at the Arab Academy for Science & Technology who supported and encouraged me.

Table of Contents

Chapter 1 Introduction……………………………………………………………. 1
1.1 Overview……………………………………………………………....... 1
1.2 The Proposed System………………………………………………………… 4
1.3 Thesis Contributions………………………………………………………. 7
1.4 Thesis Outline……………………………………………………………... 8
Chapter 2 Literature Review & Related Work 10
2.1 Introduction……………………………………………………………... 10
2.2 Glove-based Gesture Recognition Methods……………………………………. 13
2.3 Vision-based Gesture Recognition Methods………………………………..... 14
2.3.1 Model-based Methods……………………………………………..… 16
2.3.2 Appearance-based Methods…………………………………………… 17
2.3.3 Kinect-based Methods………………………………………………… 17
2.4 Related Work…………………………………………………………….… 18
2.4.1 Data Pyramids……………………………………………………….… 19
2.4.2 Principal Component Analysis……………………………………... 23
2.4.3 Perpendicular Distance Measure……………………….………….. 30
2.4.4 Multi-dimensional Grids……………………………………………… 31
2.4.5 Artificial Neural Networks…………………………………………… 35
2.4.6 Support Vector Machine……………………………………………… 39
2.4.7 Example-based Approaches………………………………………...… 48
2.4.8 Image Blurring…………………………………………………….... 43
2.4.9 Nonlinear Manifold Learning Techniques..…………………….….… 52
2.4.9.1 Nonlinear PCA…………………………………………… 52
2.4.9.2 Kernel PCA……………………………………………….. 53
2.4.9.3 Locally Linear Embedding…………………………..… 55
2.4.9.4 Isomap……………………………………………………. 56
2.4.10 Publications in which my work is cited……..………………………. 58
2.5 Conclusion…………………………………………………………………. 59
Chapter 3 Proposed Algorithms…………………………………………………... 61
3.1 Introduction……………………………………………………………... 61
3.2 Example-based Data Set……………………………………………………. 62
3.3 PCA Manifolds……………………………………………………………. 65
3.4 Nonlinearity Reduction using Image Blurring………………………………… 70
3.5 Proposed Multistage Hierarchy……………………………………………….. 75
3.6 A Multistage Hierarchical Algorithm for Hand Shape Recognition using Perpendicular Distances…………………………………………….……….. 78
3.6.1 Stage 1: Rotation Angle Estimation ….……………..………………. 79
3.6.2 Stage 2: Shape Classification………..…………………………….... 79
3.6.3 Stage 3: Translation Position Estimation ...………….……………….. 80
3.6.4 Fine Search…………………………………………………………. 82
3.6.5 Variations on the Algorithm using Sub-manifolds…..………………. 83

3.7 A Multistage Hierarchical Algorithm for Hand Shape Recognition using Supervised Multidimensional Grids………………………………..…………… 84
3.7.1 Stage 1: Rotation Angle Estimation ….……………………………. 85
3.7.2 Stage 2: Shape Classification……………………………………..…... 87
3.7.3 Stage 3: Final Classification…………………………………..……. 88
3.7.4 Handling Outliers and Empty Cells…………………..……………….. 88
3.8 A Multistage Hierarchical Algorithm for Hand Shape Recognition using Unsupervised Multidimensional Grids…………………………………………. 90
3.8.1 An Extension to Real Hands..............…………………..…..……… 93
3.9 A Multistage Hierarchical Algorithm for Hand Shape Recognition using Artificial Neural Networks…………………………………………………… 95
3.9.1 Stage 1: Rotation Angle Estimation……………………..…..……… 97
3.9.2 Stage 2: Shape Classification…………………………......……… 98
3.9.3 Stage 3: Translation Position Estimation………………..…………... 99
3.10 Conclusion………………………………………………………………... 101
Chapter 4 Experimental Results……………………………………………….. 103
4.1 Introduction………………………………………………………………... 103
4.2 First Algorithm Experimental Results……………………………………….. 105
4.3 Second Algorithm Experimental Results……………………………………….. 108
4.4 Third Algorithm Experimental Results…..……….…………………….…………… 112
4.5 Fourth Algorithm Experimental Results……………………………………... 117
4.6 Experimental Results for the Nearest Neighbour Classifier using Exhaustive Search……………………………………….…………...……………… 120
4.7 Experimental Results – Conclusion…………….…………...……………... 121
Chapter 5 Conclusion and Future Work…………….…………………………… 122
5.1 Introduction…………………………………….…………………………….. 122
5.2 Summary Of Contributions……………………….…………………………..… 123
5.3 Directions For Future Research………………………………………………..… 126
Publications Arising From The Research In This Thesis……………………….. 128
References…………………………………………………………………... 129

List of Figures

Figure 2.1, Hand gesture recognition process [15]... 12
Figure 2.2, Different approaches for hand gesture recognition................................. 12
Figure 2.3, Air-writing and sketching using data gloves [16]……………………….. 14
Figure 2.4, Flow diagram for vision-based gesture recognition methods…………… 15
Figure 2.5, Model-based hand shape tracking example [19]………………………... 16
Figure 2.6, Microsoft Kinect. 18
Figure 2.7, Searching for a target pattern over many scales using an image pyramid[21]……………………………………………………………… 19
Figure 2.8, A four level Gaussian pyramid [21]…………………………………….. 21
Figure 2.9, A four-level Laplacian pyramid [21]……………………………………. 21
Figure 2.10, An example for spatial pyramid image analysis considering three feature [22]………..…………………………………………………... 22
Figure 2.11, Applying PCA on 3D objects (a) Objects in 3D space (b) Computing the principal axes (c) Data projection onto the highest two components [23]……………………………………………………………...…… 23
Figure 2.12, Manifold examples [50]………………………………………………... 25
Figure 2.13, A SVM hand gesture recognition system using PCA for Gabor features dimensionality reduction [26]…………………………………. 27
Figure 2.14, Sample of various human body silhouettes [27]……………………….. 28
Figure 2.15, A 3D PCA space for five sequences characterizing five gestures [28]... 29
Figure 2.16, Perpendicular distance illustration from a point to an eigenspace [30]... 30
Figure 2.17, Grid clustering for 1000 patters in 3 clusters [31]……………………... 31
Figure 2.18, Grid clustering for data streaming using 2 eigenvectors [32]…………… 32
Figure 2.19, Illustration of fine grid construction [33]……………………………… 33
Figure 2.20, A three layer MLP architecture [36]…………………………………… 35
Figure 2.21, A data glove posture recognition system using ANN [37]…………….. 36
Figure 2.22, Fifteen postures using data gloves [37]…………………..…………… 37
Figure 2.23, Topological hand representation using a neural GAS algorithm [35]…. 38
Figure 2.24, A camera projected system [35]. ..…………………………………… 38
Figure 2.25, SVM decision boundary for the classification of two classes [27]……… 39
Figure 2.26, A gesture recognition system using multiclass SVMs [39]……………. 40
Figure 2.27, Key-points vectors quantization for SVM training [39]……………….. 41
Figure 2.28, Pointing gesture recognition using SVMs [40]. ……………………….. 42
Figure 2.29, A demonstration of an example-based hand shape recognition system using coloured glove in virtual reality [45]…………………………… 43
Figure 2.30, The retrieval of the nearest match using a database of tiny images for synthetic hand shapes [45]…………..……………………………... 44
Figure 2.31, 3D Hand pose estimation from cluttered images using an example-based classifier [46]…..……………………………………………... 46
Figure 2.32, Top Match examples for mapping observed images to 3D synthetic body poses [47]…...…………...……………………………………… 47
Figure 2.33, One-dimensional Gaussian function example. 48
Figure 2.34, Two-dimensional Gaussian function example. 49
Figure 2.35, Segmentation of liver part from abdominal slice [43]. 51

Figure 2.36, Gaussian blurring for liver region extraction [43]. 51
Figure 2.37, Bottleneck neural network topology [60]. 52
Figure 2.38, Locally Linear Embedding example [63]. 55
Figure 2.39, Isomap manifold learning Example [64] 56
Figure 2.32, Hierarchical pyramid image sampling [49]..…………………………... 58

Figure 3.1, Computer generated images for sign “A” in the Irish Sign Language at the horizontal and vertical position.. 63
Figure 3.2, Computer generated images for 20 shapes from the Irish Sign Language……………………………………………………………... 64
Figure 3.3, A 3D translation manifold for sign ‘A’... 66
Figure 3.4, A rotation manifold for sign ‘A’………………………………………… 67
Figure 3.5, Pseudo-code for constructing rotation sub-manifolds…………………... 67
Figure 3.6, A shape manifold, the associated shapes are shown in Fig. 3.7………… 68
Figure 3.7, Computer generated images for 20 shapes in the Irish Sign Language in the order in which they appear in Fig 3.6…….………………………… 69
Figure 3.8, Pseudo-code for constructing shape-sub manifolds……………………... 70
Figure 3.9, The effect of blurring on a computer generated image for sign ‘A’ in the Irish Sign Language…………………………………………………….. 71
Figure 3.10, The effect of image blurring on flattening a translation manifold for sign ‘A’ as shown in Fig. 3.9……………………………………………………... 72
Figure 3.11, The effect of blurring on separating the manifolds…………………… 73
Figure 3.12, The effect of blurring on flattening a rotation manifold……………….. 74
Figure 3.13, The frame-work for a PCA classifier…………………………………... 75
Figure 3.14, The proposed multistage hierarchy…………………………………….. 76
Figure 3.15, Multi-stage hierarchy using Perpendicular distance…………………… 78
Figure 3.16, Pseudo-code for Stage 1 in the 1st proposed classifier…………………. 80
Figure 3.17, Pseudo-code for Stage 2 and 3 in the 1st proposed classifier…………... 81
Figure 3.18, Variable separation at the third stage………………………………… 81
Figure 3.19, Multistage hierarchy using supervised MDGs…………………………. 84
Figure 3.20, MDG [2x2] for a rotation manifold at the 1st stage in the 2nd algorithm. 87
Figure 3.21, Pseudo-code for the 2nd proposed classifier……………………………. 89
Figure 3.22, Multistage hierarchy using un-supervised MDGs……………………... 90
Figure 3.23, The top level eigenspace……………………………………………….. 92
Figure 3.24, An eigenspace for a cell at the second level…………………………… 92
Figure 3.25, An eigenspace for a cell at the third level……………………………… 92
Figure 3.26, Pseudo-code for the 3rd proposed classifier……………………………. 93
Figure 3.27, Real hands for 20 shapes in the Irish Sign Language………………….. 94
Figure 3.28, Multistage hierarchy using ANNs……………………………………... 95
Figure 3.29, Pseudo code for the 4th proposed classifier…………………………….. 100

Figure 4.1, The effect of rotation, deformation and noise on a computer generated image……………………………………………………………………… 104
Figure 4.2, Accuracy for the 1st algorithm versus different number of eigenvectors at the 2nd stage using Rotation sub-manifolds and Shape sub-manifolds. 106
Figure 4.3, Accuracy for the 1st algorithm versus different blurring levels................. 106
Figure 4.4, Accuracy of the 1st Algorithm under a fine search strategy…………….. 107
Figure 4.5, Error analysis for the 1st Algorithm.. 108
Figure 4.6, Accuracy versus different number of eigenvectors for the 2nd algorithm.. 110
Figure 4.7, Error analysis for the 2nd algorithm... 111
Figure 4.8, Accuracy versus different number of eigenvectors for the 3rd algorithm.. 112
Figure 4.9, Error analysis for the 3rd algorithm.. 116
Figure 4.10, Accuracy versus different no. of eigenvectors at the 2nd stage for the 4th algorithm... 118
Figure 4.11, Training error rate versus different blurring levels................................. 118
Figure 4.12, Error analysis for the 4th algorithm... 119

List of Tables

Table 4.1, Different blurring levels and their filter size... 103

Table 4.2, Accuracy and performance of the 1st algorithm under different PCA spaces.. 107

Table 4.3, Maximum range of angles at Stage 1.A for the 2nd algorithm…………… 109

Table 4.4, Accuracy at stage 1.A for the 2nd algorithm……………….…………….... 109

Table 4.5, The Accuracy for the 2nd algorithm……………………………………… 111

Table 4.6, Accuracy using different grid structures for the 3rd algorithm…………… 113

Table 4.7, The number of grids in each level of the pyramid for three different grid structures………………………………………………………………… 113

Table 4.8, Accuracy measure under different thresholds for the second grid structure……………………………………………………………….. 114

Table 4.9, The number of grids in each level of the pyramid for the second grid structure under different thresholds…………………….………………… 114

Table 4.10, Accuracy for the 3rd algorithm..………………………………………… 115

Table 4.11, Neural networks architecture designs at the different stages................... 117

Table 4.12, Accuracy for the 4th algorithm.. 119

Table 4.13, Accuracy for the exhaustive search versus all proposed techniques based on a random sample of 10,000 test objects.................................. 120

Table 4.14, The speed up factor for the proposed algorithms against the exhaustive search.. 121

Principal Component Pyramids using Image Blurring for Nonlinearity Reduction in Hand Shape Recognition

By

Mohamed Farouk Kheir Eldin, M.Sc.

Abstract

The thesis presents four algorithms using a multistage hierarchical strategy for hand shape recognition. The proposed multistage hierarchy analyzes new patterns by projecting them into the different levels of a data pyramid, which consists of different principal component spaces. Image blurring is used to reduce the nonlinearity in manifolds generated by a set of example images. Flattening the space helps in classifying different hand shapes more accurately. Four algorithms using different pattern recognition techniques are proposed. The first algorithm is based on using perpendicular distance to measure the distance between new patterns and the nearest manifold. The second algorithm is based on using supervised multidimensional grids. The third algorithm uses unsupervised multidimensional grids to cluster the space into cells of similar objects. The fourth algorithm is based on training a set of simple architecture multi-layer neural networks at the different levels of the pyramid to map new patterns to the closest class. The proposed algorithms are categorized as example-based approaches where a large set of computer generated images are used to densely sample the space. Experimental results are presented to examine the accuracy and performance of the proposed algorithms. The effect of image blurring on reducing the nonlinearity in manifolds is examined. The results are compared with the exhaustive search scenario. The experimental results show that the proposed algorithms are applicable for real time applications with high accuracy measures. They can achieve frame rates of more than 10 frames per second and accuracies of up to 98% on test data.

Chapter 1

Introduction

1.1 Overview

Computer Vision (CV) is a field that helps in understanding digital images. CV extracts from the visual appearance of the images enough information for decision making. Artificial systems are used to reach this goal by extracting the information from images through the steps of preprocessing, analysis and classification. The image data is usually taken from video sequences using single or multiple cameras. The images can also be gathered from medical scanners as multi-dimensional data. [1,2,3]

CV has many applications. Medical computer vision concentrates on extracting information from images for patient diagnosis. It can be used for measuring the size and shape of organs. Also, image enhancement can be applied to images to reduce the effect of noise [51]. CV has some applications in the field of industry. It can be used as a quality control aid tool to automatically classify the category of manufactured products. It can also be used in safety support by monitoring the actions made by workers to control machines and avoid accidents [52]. Autonomous vehicle control is one of the recent applications in this area. CV-based systems can support drivers or pilots in various situations. It can be used for navigation, detecting obstacles and autonomous landing of aircraft [53].

Human Computer Interaction (HCI) is one of the areas where CV plays a great role. CV methods can be used to implement a new way of interaction. Hand shape recognition for gestures provides a natural interaction between humans and computers. Hand shapes hold the information about the meaning of the gestures. As the hand is a deformable object, it is a challenging task to build a recognition system that can classify the same hand shape under different orientations or different viewpoints of the camera. Some of these applications are human-robot interaction, smart environments and multimedia analysis. Gesture recognition can help in achieving these applications. By performing a gesture, a robot can indicate where an object is by pointing to it. Smart meeting rooms can use gesture recognition for controlling presentations. Activity recognition can be used to analyze and categorize the huge amount of available multimedia data for automatic retrieval systems. [4,5]

Gesture recognition approaches can be divided into glove-based and vision-based. The first approach uses electronic gloves to gather the information via a set of sensors about the hand shape and its position. Data gloves give good information as they directly measure the pose of the hand and fingers. However, they are expensive and are cumbersome to use. On the other hand, vision-based methods use only a camera to capture the hand shape in a natural way of interaction. The input video is decomposed into a sequence of frames that pass through a pre-processing stage to remove the unnecessary data. For example, hand and arm can be segmented from other background objects. There are two categories of vision-based systems, model-based and appearance-based. [6]

Model-based methods depend on the 3D kinematics of a hand model, which takes into account all possible different degrees of freedom. A model-based approach provides a rich description of the hand shapes. However, it is computationally expensive, as a large number of instances of the model have to be generated to compare with the target image. Appearance-based methods depend on extracting the features of the images from the input video frames. Generally these methods have the advantage of real-time performance. [7]

There are different techniques used to build classifiers in the category of appearance-based methods for gesture recognition. Some of them are Principal Component Analysis (PCA), Multidimensional grids, Artificial Neural Networks and Support vector machines. PCA can be used as a feature extraction and dimensionality reduction method. The data are projected into the eigenspace defined by the principal axes calculated from the covariance matrix of the training data. These principal axes represent a set of eigenvectors that have the highest eigenvalues. PCA can be viewed as a manifold-visualisation method that reveals the low-dimensional structures, which are embedded in the high dimensional feature space. PCA usually works well if the embedded manifold is linear. A disadvantage of PCA is that it is highly sensitive to position, orientation, scaling, and lighting effects, which can lead to non-linear manifolds. Some of these parameters can be solved through a pre-processing stage which sometimes is time consuming. [8,9,23]

Example-based approaches, which are a type of appearance-based method, use a large database of hand images to cover the possible variations in hand shapes. Many hand shape recognition algorithms need to apply an alignment as a pre-processing step to rotate or scale the hand shape in a new image in order to classify it. Example-based approaches can avoid this preprocessing step by building a large database for training purposes. Computer-generated images can be used to build this database, as it is possible to control the position and orientation of the hand at regular intervals to densely sample the space, which is not easy if we use natural images. These approaches are limited to the poses or motions used in the training set and an extension to a wider variation may cause data overlapping and make the classification process more difficult. Depending on the size of the example images, the performance speed can be improved by accelerating database queries through applying hashing as an indexing method to approximate the k-nearest neighbour problem. However, the classification accuracy usually is sensitive to the choice of the hashing functions and the size of the hash tables as well. [10]

1.2 The Proposed System

This section gives an introduction to the proposed algorithms. The thesis discusses the design of four algorithms using hierarchical pyramids. Image blurring is applied to reduce the nonlinearity in manifolds. PCA is used for building a set of eigenspaces for a set of example images. The thesis discusses the acceleration of the search process, robustness to rigid transformations using PCA and the effect of nonlinearity reduction using image blurring on increasing the accuracy measure.

This thesis discusses accelerating the classification process by using data pyramids consisting of PCA spaces based on example images for 20 shapes from the Irish Sign Language. PCA is used for dimensionality reduction and feature extraction. As a large set of manifolds are generated to cover the wide variations in the data, it is time consuming to exhaustively search them in order to classify new patterns. The rendered images are generated using a computer graphics software package to form a large database of 111320 objects. These objects represent the 20 shapes at different rotations and translations in order to densely sample the space and increase the robustness of the system. While PCA eigenspaces are sensitive to translation and rotation, the proposed algorithms are invariant to these rigid transformations.

This thesis discusses the contribution of using data pyramids as a multistage hierarchical algorithm to organize the different eigenspaces that are generated by applying PCA on the set of example images. PCA is an unsupervised learning technique that builds eigenspaces based on the feature space of the original data without taking into consideration the different classes of the data, which represent the different shapes and poses for the hands. The data pyramid organizes the eigenspaces in a supervised manner, where different eigenspaces are generated with example images for the same shape or at the same rotation.

The proposed algorithms explore the effect of image blurring on the manifolds. Image blurring has the effect of removing small changes between objects. It is possible at a certain blurring level to estimate the pose parameters for new incoming patterns based on labeling by the nearest neighbour object from the set of example images. The multistage hierarchy uses image blurring to help in estimating the pose parameters for the new patterns by focusing the search process within a small number of eigenspaces from one level to the next. A final decision on the pose parameters can be taken at the bottom level of the pyramid.

Based on our set of example images, the thesis discusses the effect of reducing the nonlinearity in the manifolds by using image blurring. Nonlinearity reduction in manifolds plays a great role in designing the different classifiers in this thesis to get high accuracy measures as will be discussed in Chapter (3). Four algorithms are discussed through the thesis using different techniques. The first algorithm depends on a perpendicular distance measure [30] in searching for the closest manifold. The perpendicular distance measure assumes that the manifolds are linear and so blurring the example images helps in increasing the accuracy of the measure. The second and the third algorithms use Multidimensional Grids (MDGs) [31] for grouping the training data into cells containing objects that have high similarity. The cells are hyper-cubes with linear sides and reducing the nonlinearity in manifolds with image blurring has the effect of better separation between the objects within the grid cells. The fourth algorithm is based on using Artificial Neural Networks. As the pyramid organizes the manifolds into a multistage hierarchy, this gives the algorithm the advantage of using simpler Multi-Layer Perceptrons (MLPs) architectures. Each MLP is trained with a subspace, that represents certain parameters, instead of training a big architecture with the original global space. Nonlinearity reduction has the effect of speeding up the training process for the neural networks using the Feed Forward Back Propagation algorithm. It also helps to design simpler MLP architectures because less nonlinear decision boundaries can be found to solve the classification problem as will be discussed in Chapter (3).

1.3 Thesis Contributions

- Four algorithms are presented which can classify hand images at different in-plane rotations and translations. The algorithms do not require pre-alignment of the images. Other variations, such as scale and lighting will be treated in future work.

- A new way of organizing data pyramids is introduced in this thesis by analysing the example images using different PCA spaces throughout the hierarchy. Each level of the pyramid represents a different parameter of the hand image. This differs from most previous research, where typically data pyramids have been used to analyse images at different scales or resolutions.

- Based on our set of example images, we demonstrate that image blurring can be used for nonlinearity reduction in manifolds. This can help in increasing the accuracy level of the classification process and speeds up training stage by building more linear decision boundaries.

- The proposed PCA pyramids provide an alternative to the indexing method to accelerate the search process to approximate the k nearest neighbour problem. Indexing methods usually are sensitive to the choice of the hash functions.

- The proposed PCA pyramids provide a method to design simple architectures for the classification process. The simplicity of the design can be expressed in terms of the size of the MDGs or the number of neurons and layers in the MLP architecture.

1.4 Thesis Outline

The thesis contains five chapters that are organized as follows:

- Chapter 1 introduces the area of Computer Vision. Human Computer Interaction is one of the areas, to which CV contributes. In particular, gesture recognition systems provide a natural form of interaction between human and computer. An introduction to the thesis is presented showing how PCA pyramids are used in the application of hand shape recognition.

- Chapter 2 reviews the research area of gesture recognition including the different methodologies that have been applied. The related work section refers to some algorithms that use the same techniques, on which our thesis is based.

- Chapter 3 discusses four algorithms that use PCA pyramids in conjunction with different classifiers. The first algorithm is based on a perpendicular distance measure to find the nearest manifold through the different levels of the hierarchy. The second and the third algorithm use multidimensional grids to divide the space into hyper-cubic cells in a supervised and un-supervised manner respectively. The fourth algorithm trains Multilayer Perceptrons on the different PCA subspaces using a Feed Forward Back Propagation algorithm to map the new patterns to closest class.

- Chapter 4 shows the experimental results for each algorithm based on three test sets. The first test-set is generated from the original images by rotating each image by a random angle filling the intermediate angles between the training images. The second test-set is generated by adding a deformation to each image of the previous set by randomly rotating the hand at the wrist. To examine the robustness of the algorithm to noise, the third test-set is generated by adding white Gaussian noise to the second test set. The experimental results show the accuracy of each aspect and the speed up over the exhaustive search method.

- Chapter 5 concludes the thesis and mentions the future work.

Chapter 2

Literature Review & Related Work

2.1 Introduction

Gestures are a useful way of communication between people to express what they want to say in everyday life. Hand shape recognition for gestures provides a natural interaction between humans and computers. As a new type of Human Computer Interaction (HCI), hand gesture recognition has attracted the interest of many researchers in recent years to bring ease and naturalness to cumbersome interface devices. HCI keeps moving closer towards natural user interfaces [11].

The key problem in gesture interaction is how to make hand gestures understood by computers. An important aspect of gestures is the hand shape. Hand shapes hold important information about the meaning of a gesture. Hand shape recognition is a very challenging task, where the orientation of the hand and the viewpoint of the camera can make the same hand shape look different [12]. This is the topic of this thesis.

Gesture recognition has been used in many areas such as Human Computer Interaction, Virtual Reality, Robotic Control and Game Control [12,13]. Some of the applications in these areas are:

- Automatic sign language translators that facilitate communication for deaf people. These systems analyze the signer gestures and synthesize the sound for the corresponding word or letter for hearing people to understand.
- Distant control of robots or vehicles in dangerous environments for humans, e.g. underwater, in space, etc.
- Manipulation of endoscopic tools through surgical simulation or tele-surgical procedures.
- Intelligent homes. Example of common functions that can be applied using gesture recognition are, Vol +, Vol -, Play, Stop and Power off.

Automatic sign language recognition systems, in particular, reduce the barrier of communication for deaf people in everyday life so that hearing people can understand their expressions. In a sign language, gestures are used instead of sound to convey meaning by simultaneously combining hand shapes, orientations and movement of the hand. Gestures can be viewed as a series of postures, which represent static hand shapes within a certain period of time. Holding the hand with a specific pose is a posture, e.g a victory sign, pointing, or thumbs up. A gesture can be defined as a dynamic movement, such as waving good-bye. Sign language usually provides signs for whole words. It can also provide signs for letters of the alphabet to perform words that don’t have a corresponding sign in that sign language [14]. In automatic sign language recognition, signs can be considered as a continuous sequence of postures with different hand shapes and positions within a small interval of time under a certain gesture grammar as shown in Figure 2.1. The grammar describes the meaning of different sequence of postures. A feed-back can be applied to the system giving a prediction for the next posture as an error correction technique [15]. The details of the internal structure are given in the next sections. Gesture recognition approaches can be divided into either glove-based methods or vision-based methods as shown in Figure 2.2. And vision-based methods can be further divided into model-based, appearance-based and Kinect-based.

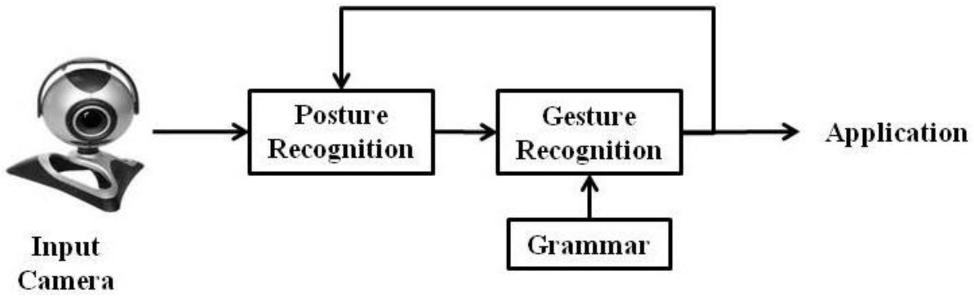


Figure 2.1, Hand gesture recognition process.

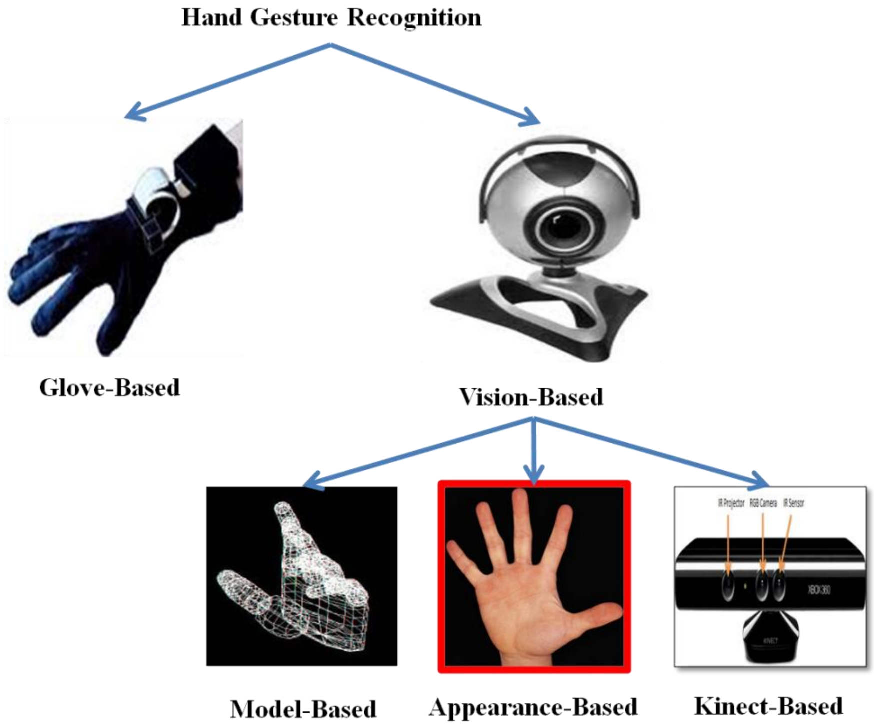


Figure 2.2, Different approaches for hand gesture recognition.

2.2 Glove-based Gesture Recognition Methods

Data gloves are electronic devices that have a set of sensors to digitize the hand and finger movements. It is possible to build a classifier to recognize the hand shape and its position according to the electronic data gathered from them. The set of sensors can give enough information to make it easy to get the hand configuration. Electronic gloves can determine various posture parameters, including the hand position, angle and the location of the fingertips. The major problem is that they are expensive and are cumbersome for the users. Wearing inconvenient devices, results in a less natural interaction.

In [14], a wireless data glove is introduced as an electronic device that is used for sign language translation. The gestures are recognized and automatically translated to speech. The type of the electronic glove which is used by the system has flex sensors along each finger. These sensors change in resistance depending on the amount of stretch happens by the signer fingers. The sensor data are stored for different gestures for training and recognition. The system has a drawback of losing the ability of measuring and recognizing the shape or motion of the arm and elbows, so that only static gestures can be recognized by the system. The problem can be solved by attaching more sensors at the wrist, elbow and shoulder but this will increase the cumbersomeness for the users.

In [16], data gloves and a nearest neighbour classifier are used to build a real-time HCI system for gesture recognition. The system recognizes five gestures, such as clicking, rotating, dragging, pointing and pause. These gestures are helpful in Virtual Reality and Augmented Reality where gesture recognition is considered as the main input device. The data glove used in this experiment is the DG5 VHand 2.0, which is a wireless data glove based on Bluetooth technology. It has a working range of up to 10 meters to give more freedom to the user. Figure 2.3 shows how the system can be used in air-writing and sketching. The system can also be used in 3D gaming and 3D animation.

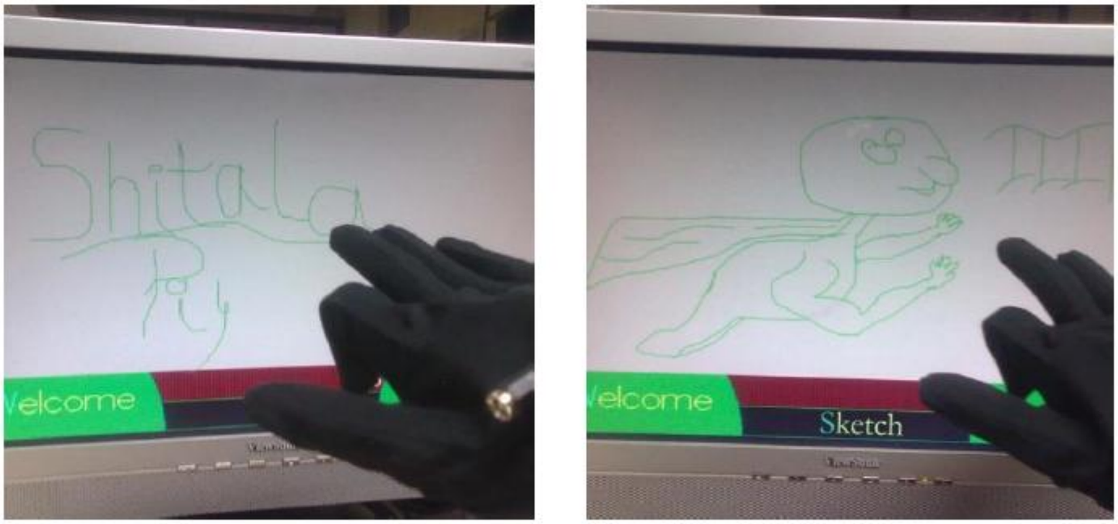


Figure 2.3, Air-writing and sketching using data gloves [16].

2.3 Vision-based Gesture Recognition Methods

Computer vision (CV) techniques can generate intelligent and useful descriptions of visual appearance by performing processing on the signals received from video cameras. CV provides a more natural interaction and is less cumbersome for the user than data gloves. The challenge of computer vision is to recognize the visual representation in a flexible and robust way. Several challenges have to be achieved including accuracy, processing speed and generality.

The hand is a deformable object and recognizing the full degrees of freedom for a hand motion is a challenging problem, where data glove methods have an advantage. Vision-based methods use only a camera to capture the hand shape in a natural way of interaction. The input video is decomposed into a sequence of frames that pass through a pre-processing stage to remove the unnecessary data. For example, the hand and arm can be segmented from other background objects [17,18]. Figure 2.4 shows the sequence of steps that a vision-based system can pass through. There are two categories of vision-based systems, model-based or appearance-based, as will be discussed in the following sections.

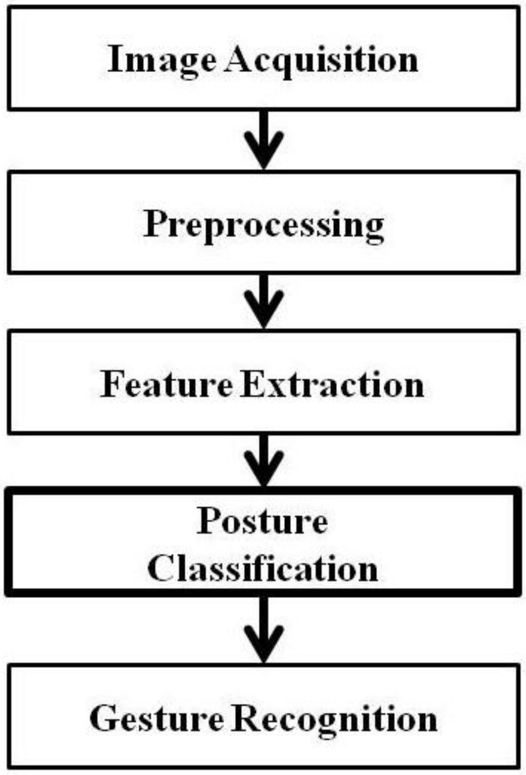


Figure 2.4, Flow diagram for vision-based gesture recognition methods.

2.3.1 Model-based Methods

Model-based methods depend on the 3D kinematics of the hand considering the possible different degrees of freedom (DOF). A model-based approach provides a rich description for the hand which can help in achieving a full DOF hand pose estimation. This approach can help in estimating all kinematic parameters of the hand skeleton to be able to achieve a full hand reconstruction. For the purpose of matching, a projection of the 3D model on the image plane is commonly used. However, it is a computationally expensive process due to the need to match with a very large set of 3D models covering all the characteristic hand shapes using high level features extraction methods like fingertips and line features. 3D features from stereo-cameras can also be used by these systems to obtain the 3D depth view parameters in the images for matching. Model-based approaches can effectively be used for hand tracking using multiple hypotheses by finding multiple pose estimation for each frame. The system can continue tracking in the case that the best estimate fails. Figure 2.5 shows an example for tracking results using a model-based approach in the presence of a considerable amount of hand shape occlusions [19,20].

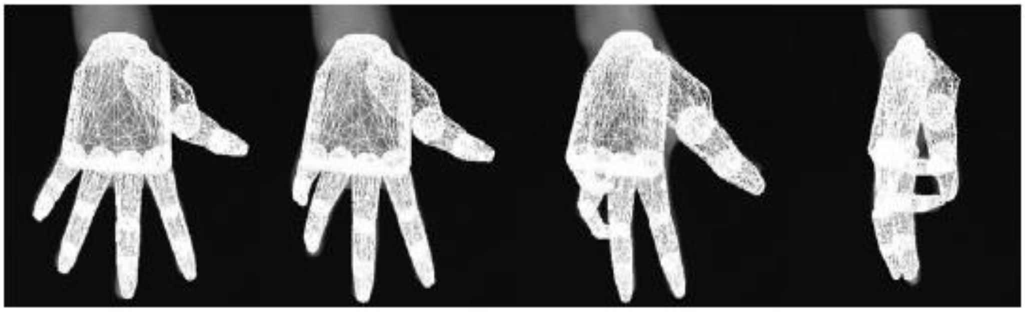


Figure 2.5, Model-based hand shape tracking example [19].

2.3.2 Appearance-based Methods

Appearance-based methods depend on recognizing objects from their visual information by extracting image features to model their visual appearance. Classification is achieved by comparing these features with the extracted image features from the input video. It is a challenging task that can be applied to many problems in computer vision, especially to hand gesture recognition as the hand is a non-rigid object. These methods can give partial hand pose estimation parameters. This information can be used in 2D applications with a low DOF task like pointing or navigation. Generally these methods are used because of their simplicity and real-time performance. However, the learning algorithms in this category are always difficult and it is usually not easy to collect large training datasets. There are different techniques used to build classifiers in this category. Some of these techniques are Principal Component Analysis, Artificial Neural Networks, and Support Vector Machines. [17,18]

2.3.3 Kinect-based Methods

Kinect is a novel device that can help in building gesture recognition techniques. Kinect is a RGB camera with additional IR projector and IR sensor as a depth sensing input device as shown Figure 2.6. This device can be used to develop gesture recognition algorithms as an extension to the vision-based techniques. Kinect is offered by Microsoft to enable users to control and interact with computers through a natural user interface using gestures without the need for a game interface device. A version for Windows was released in 2012 with a software development kit to allow developers to write Kinecting applications in a number of high level programming languages.

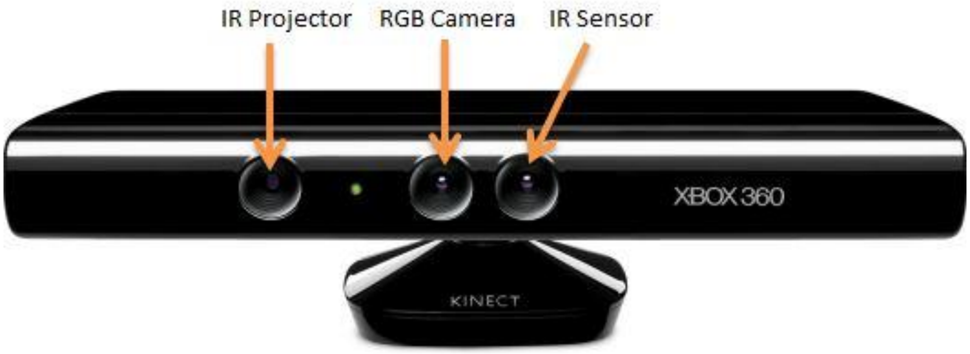


Figure 2.6, Microsoft Kinect.

The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video frames in 3D under any lighting conditions. The Kinect software is capable of automatically calibrating the sensor based on game play and the player's physical environment under the presence of obstacles. The software technology enables advanced gesture recognition and facial recognition. Kinect is capable of simultaneously tracking two active players for motion analysis with a feature extraction of 20 joints per player. [66]

2.4 Related Work

The algorithms proposed in this thesis fall into the category of appearance-based methods, using a set of example images. They use computer-generated images in order to ease the training process by creating large set of labelled images, which is hard for a human to do. In this section a review of the related techniques that are used to build the systems described in this thesis is presented. These techniques are listed in the following sub-sections according to the flow of the proposed ideas and the order in which they appear in the next chapter.

2.4.1 Data Pyramids

A Data Pyramid is a data structure that represents image information in different ways at each level of the pyramid. Pyramid-based methods have been applied to image analysis, data compression and image manipulation. In [21], a data pyramid structure is introduced to solve the task of matching patterns, which may appear at any scale in an image. This image pyramid stores different versions of the images at different scales. The images are decreased in resolution in regular steps through the different levels. The target pattern can be found in the level that represents the best match by applying a convolution process between the target image and the sequence of images that are stored in the levels of the pyramid as shown in Figure 2.7.

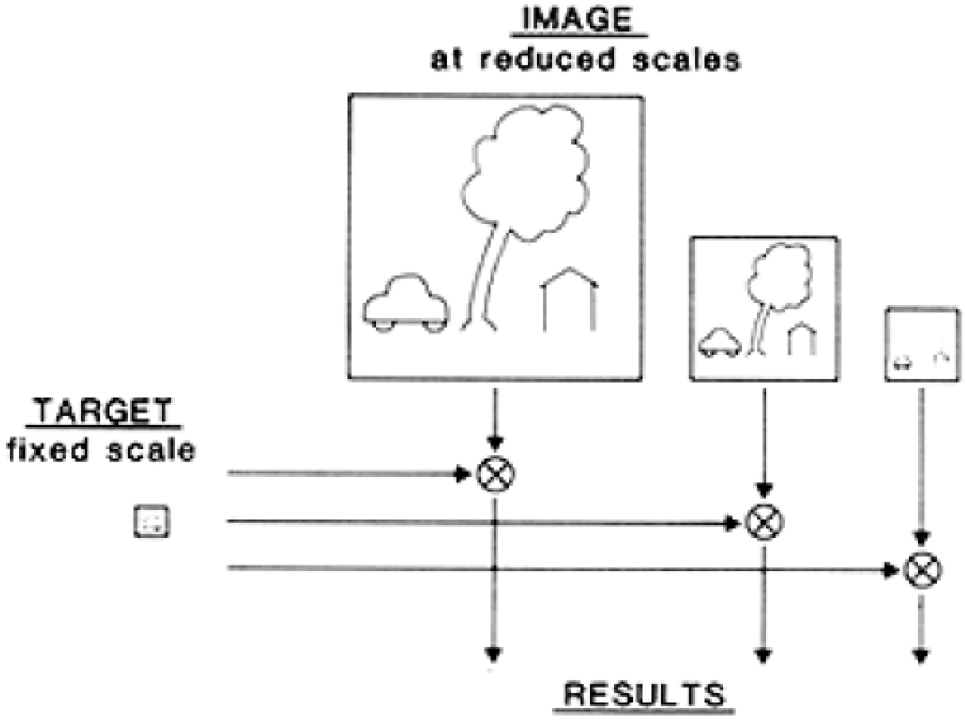


Figure 2.7, Searching for a target pattern over many scales using an image pyramid [21].

This pyramid structure can be used as a coarse-to-fine search technique to provide a fast, coarse estimate of the possible location of the target pattern. The search begins at the top level of the pyramid by convolving a reduced resolution pattern with a reduced resolution version of the stored images. The search proceeds through the pyramid levels, at each level increasing resolution and position refinement.

The same pyramid hierarchy can be used for image compression by providing a compact code. A Gaussian pyramid can be constructed by building a hierarchy of images reduced in sample density and resolution in regular steps by a factor of two. This can be achieved by convolving the images with a Gaussian function as a low-pass filter. Figure 2.8 shows a four level Gaussian filter where the band limit is reduced by one octave at each new level. A Band-pass pyramid, which is referred to as a Laplacian pyramid, can be constructed by subtracting each two consecutive Gaussian pyramid levels. This process is reversible and can be used to reconstruct the original level. Figure 2.8 shows a four-level Laplacian pyramid. This pyramid can be applied in image compression using a compact code as the pixel values in these image samples tend to be near zero. A further compression can be gained through a vector quantization process. As shown in Figure 2.9, the Laplacian pyramid has the advantage of enhancing the image features by extracting the edges of the individual objects in the image.

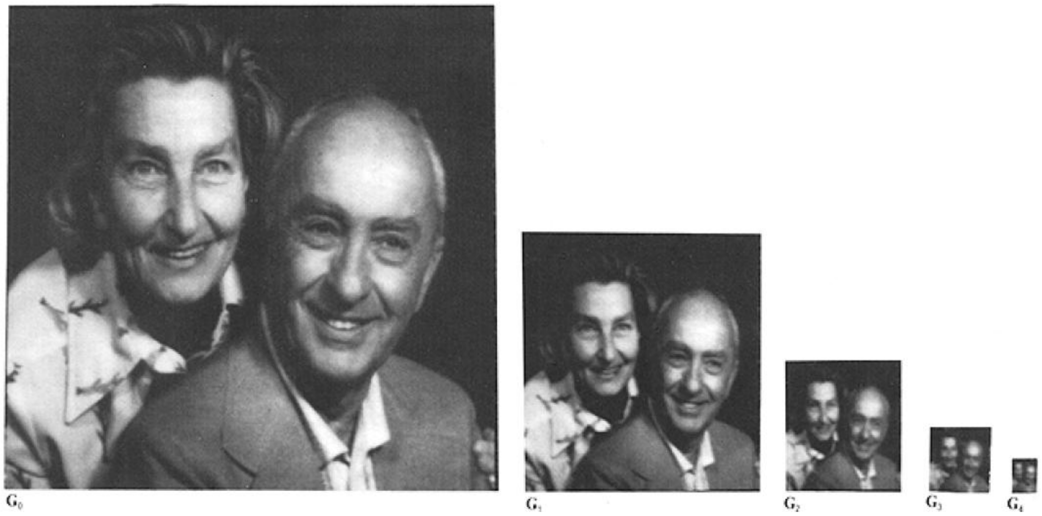


Figure 2.8, A four level Gaussian Pyramid [21].

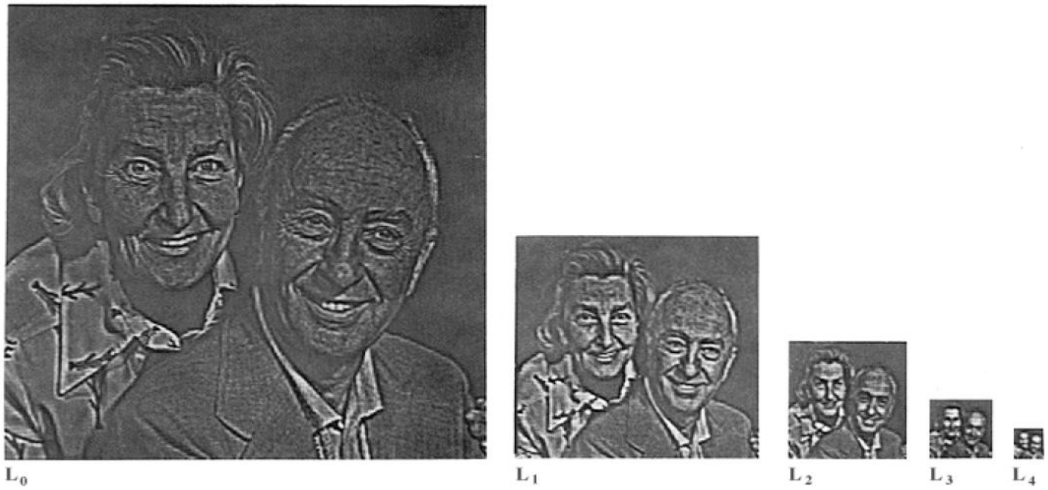


Figure 2.9, A four-level Laplacian Pyramid [21].

The pyramid method was introduced as a data structure for image analysis in multi-scale resolution. This pyramid hierarchy is applicable for other image processing tasks. In [22], a method for recognizing scene categories is presented using spatial pyramids for feature extraction and matching. The spatial pyramid method is more useful than other transformation representations when the spatial location of a pattern is critical. The technique works by partitioning the image into sub-regions through the pyramid. From one level to another the image is increasingly subdivided. Histograms are computed for the local features inside each sub-region. Matching is calculated based on taking a weighted sum of the number of matches that occur between a new image and the training set at each level of the spatial pyramid where matches found at a higher resolution are weighted more.

Figure 2.10 shows an example of a three-level spatial pyramid. The image analysis is based on three feature types. They are indicated by circles, diamonds, and crosses. Three histograms are calculated for each image cell in each level for the matching process. A histogram is calculated for each channel. At the top level the whole image is considered as one cell where the image is subdivided through the different levels to get more resolutions.

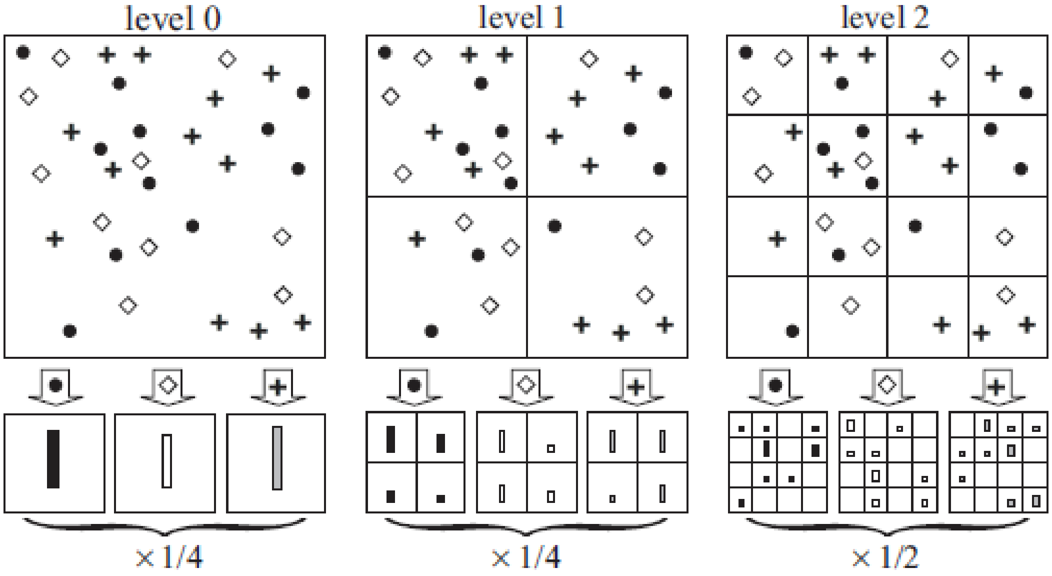


Figure 2.10, An example for spatial pyramid image analysis considering three feature [22].

2.4.2 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique which can be used as a feature extraction and dimensionality reduction method. PCA has been used in many applications to reduce a complex dataset to a lower dimensional space to discover the hidden and simplified structure that often underlies it. PCA reduces the dimensionality of data while preserving as much information as possible. This is achieved through maximizing the variance and minimizing the mean squared reconstruction error for the projected data [23]. Images can be considered as objects in a high dimensional space spanned by the values of the individual pixels. PCA is used to map these images to a low-dimensional linear space that represents the images as shown in Figure 2.11. PCA assumes that the set of model images lies in a low-dimensional subspace spanned by the eigenvectors of the covariance matrix of a set of images.

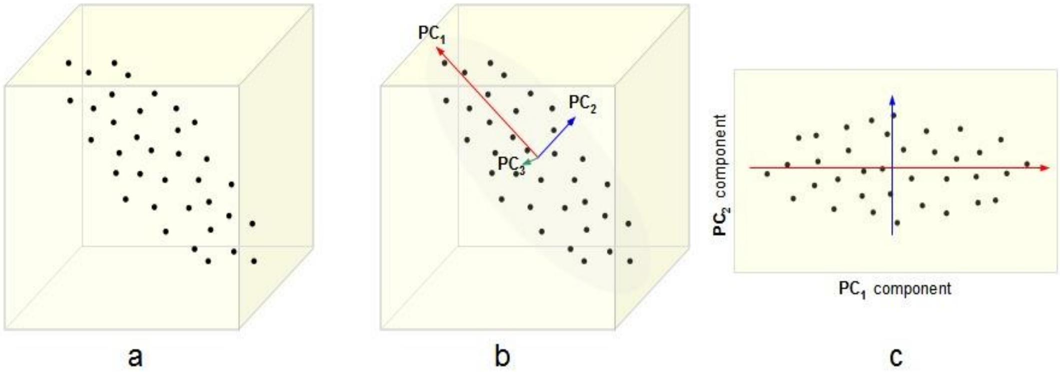


Figure 2.11, Applying PCA on 3D objects (a) Objects in 3D space (b) Computing the principal axes (c) Data projection onto the highest two components [23].

PCA is a well-known method for feature extraction by calculating the eigenvectors of the covariance matrix of the original input data. In our case, the input data consists of a set of images of hand shapes. PCA linearly transforms a high-dimensional input vector into a low-dimensional one, whose components are uncorrelated [55]. Given a set of *n* images defined by Equation 2.1, where $\boldsymbol{x_i}$ is the i^{th} image reshaped as a one-dimensional vector of length *N*, where *N* is the number of pixels in each image.

$$X = \{x_1, x_2, x_3 .... x_n\}^T \qquad (2.1)$$

The covariance matrix is an *N*×*N* matrix whose elements can be computed by Equation 2.2, where a_k and b_k are the i^{th} and j^{th} pixel values of the k^{th} image.

$$\sigma_{ij}^2 = \sum_{k=1}^{n} (a_k - m_a)(b_k - m_b) \qquad (2.2)$$

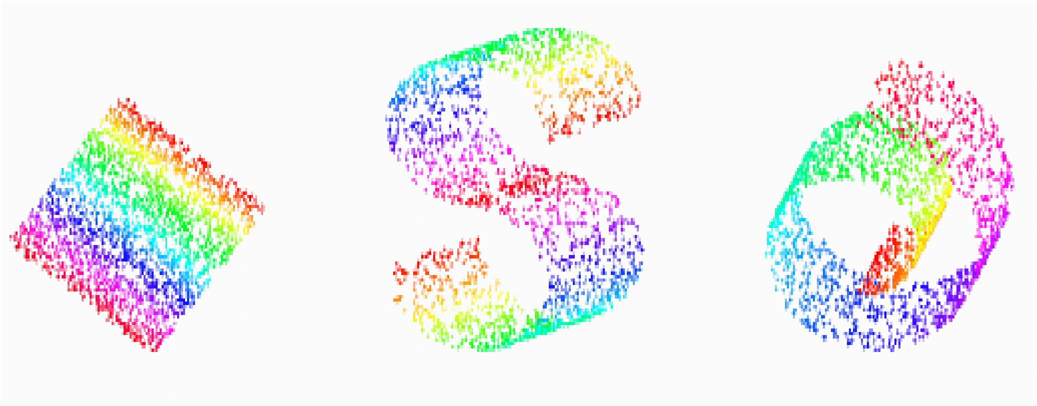
The eigenvalues and eigenvectors of the covariance matrix *C* can be found by solving the eigenproblem in Equation 2.3, where u_k is the k^{th} eigenvector and λ_k is the k^{th} eigenvalue. The eigenvalue is a measure of the variation in the data along its eigenvector.

$$Cu_k = \lambda_k u_k \qquad (2.3)$$

Finding the eigenvectors of C we get *N* eigenvectors, each of length *N*. A subset ***W*** is retained from the set of eigenvectors representing the eigenvectors with the greatest variation. The new feature vector *y* for image *x* is computed as follows:

$$y = W^T x \qquad (2.4)$$

PCA is a linear method that can be used as a manifold visualisation technique by finding a low-dimensional structure that is embedded in a higher-dimensional space. Manifolds offer a powerful framework for dimension reduction. PCA usually is effective in constructing manifolds if the embedded data is linear. For example PCA is useful for the data in Figure 2.12 (a). However, it is not so good for nonlinear data in (b) and (c). Some of the techniques that can work with these nonlinear data are Locally Linear Embedding and IsoMap [24].



(a) (b) (c)

Figure 2.12, Manifold examples [50].

A manifold can be visualized in a low-dimensional space by projecting the original images into an eigenspace using a number of eigenvectors that have the highest eigenvalues. Some approaches use PCA as a pre-processing step to reduce the dimensionality of the data to reduce the computational complexity for a classification task. Some other approaches use PCA to extract features and then apply a learning algorithm for the classification process as will be discussed with examples later in this section [23].

A disadvantage of PCA is that it is highly sensitive to position, orientation, scaling, and lighting effects. For example, PCA cannot transform two identical postures with different hand sizes and orientation to the same point in the low-dimensional space. Some techniques therefore normalize each image to centre the hand, rotate, and scale it to fit the posture image. Another disadvantage is that the training set must contain all the variations for robustness and PCA requires training by more than one person for accurate results and user independence [25].

In [26], a dynamic gesture recognition system is proposed based on Gabor features. PCA is used to reduce the dimensionality of Gabor filtered images where the Support Vector Machine method is used to carry out the recognition task. Gabor features have several advantages, such as localizability and orientation selectivity. However, Gabor features are inefficient to use due to high dimensionality. Gabor filters are used for pose estimation to apply a hand orientation correction as a next step. Different Gabor filters at 3 different scales and 8 different orientations are used for the training stage. New images are convolved with 24 filters and the filter with the highest response in gray scale can give the closest pose of the incoming hand posture. The hand only is segmented for the classification process after the correction stage. The algorithm is robust to different illumination effects using an adaptive skin-color model switching method. Figure 2.13 show the flow diagram for the system.

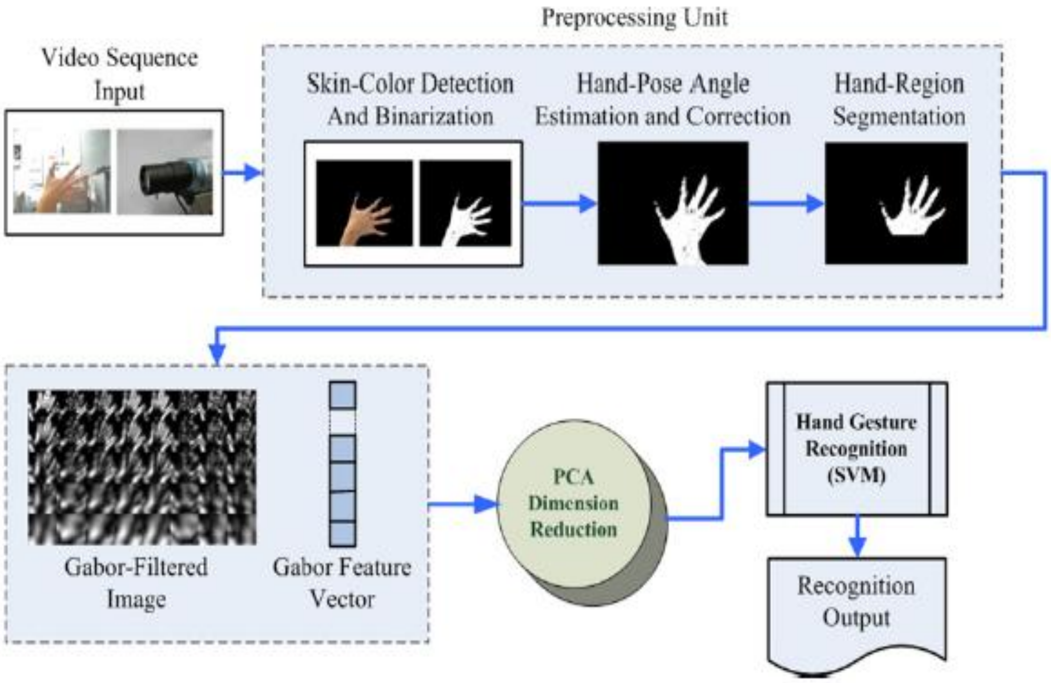


Figure 2.13, A SVM hand gesture recognition system using PCA for Gabor features dimensionality reduction [26].

In [27], the PCA technique is used to extract the features from human shape silhouettes to analyze and classify human body posture. Support Vector Machines are used for the classification task to classify either human standing posture or human non-standing posture. The algorithm is built based on testing various combinations of two eigenpostures to get the highest accuracy considering the highest 6 eigenvectors. Figure 2.14 shows a sample of images for the human body silhouettes used in training and testing the system.

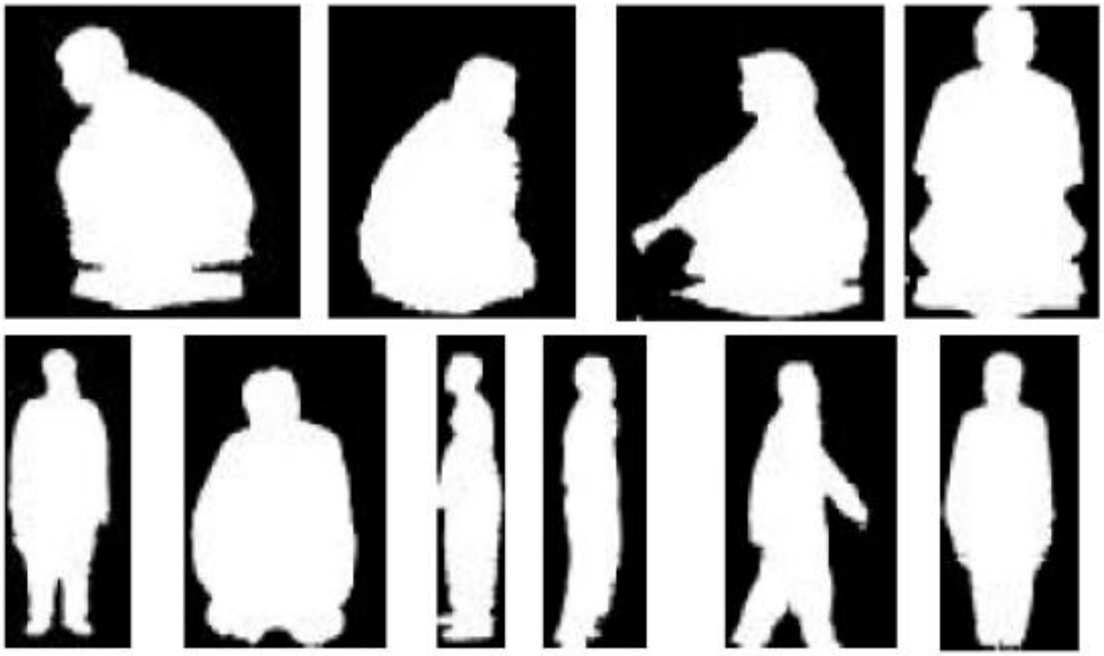


Figure 2.14, Sample of various human body silhouettes [27].

In [28], a human-machine interface based on visual appearance of hand movements is presented. A statistical approach based on Hidden Markov Models (HMMs) is applied after using PCA for dimensionality reduction and features extraction from the input sequence. Figure 2.15 shows five gestures projected into a PCA space constructed from the highest three eigenvectors. For the recognition task, five HMMs, one for each hand movement, are trained using ten repetitions of each gesture. The probability for each model is computed for a new gesture and the largest probability corresponds to the recognized gesture.

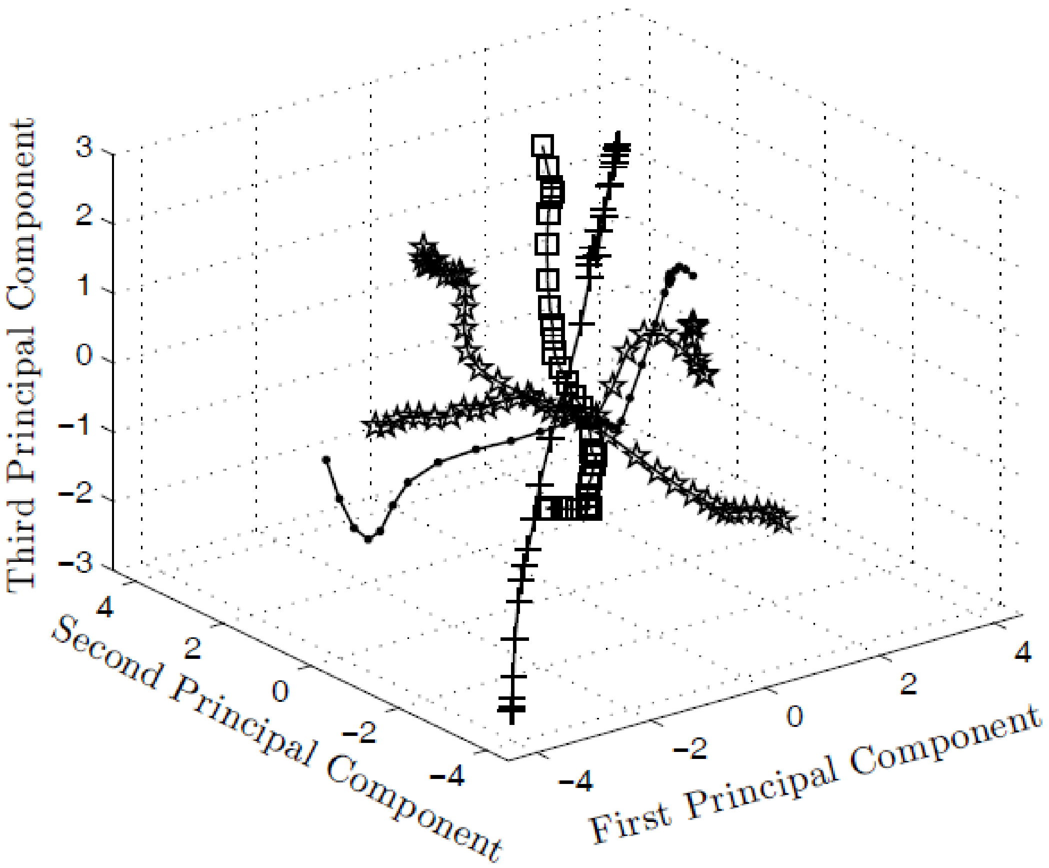


Figure 2.15, A 3D PCA space for five sequences characterizing five gestures [28].

2.4.3 Perpendicular Distance Measure

A PCA eigenspace is constructed using a number of orthogonal eigenvectors that have the highest eigenvalues to maintain the maximum energy of the dataset. Perpendicular distance measures the residual between the original vector and its PCA representation. Perpendicular distance is a distance metric that is used to compute the distance between a projected object and an eigenspace. During the classification process, each class may be represented by a different eigenspace. An object can be classified by computing its perpendicular distance to each eigenspace. The class with the lowest perpendicular distance is the most likely to which the object belongs. Figure 2.16 illustrates diagrammatically the perpendicular distance (D_p) of a point p to an eigenspace E where it is mathematically calculated using the Equation 2.5 [29,30].

$$D_p{}^2 = D_e{}^2 - \sum_{i=1}^{M} [(\underline{p} - \underline{o}) \bullet \underline{u}_i]^2 \qquad (2.5)$$

Where D_e = Euclidean distance between **p** and the origin **o**.

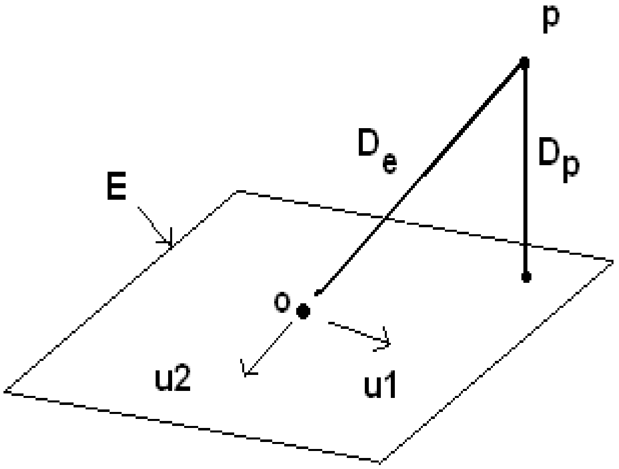


Figure 2.16, Perpendicular distance illustration from a point to an eigenspace [30].

2.4.4 Multi-dimensional Grids

A multi-dimensional grid is a data structure that organizes the space around objects that are represented in multi-dimensional space. Clustering is one of the fundamental techniques in pattern recognition. Clustering can discover the relevant groups of data within the dataset for further analysis. Grid clustering applies block partitioning to the data through a neighbour search. Grid clustering is different from other clustering techniques as it does not organize the patterns but the feature space that represents them. The grid structure partitions the objects into buckets of data within hyper rectangular cells. The similarity is high for the objects within each cell and low for data in other cells. A fuzzy clustering technique can solve the problem of the objects on the boundaries. Figure 2.17 shows a block partitioning for a data set of 1000 objects for 3 classes. [31]

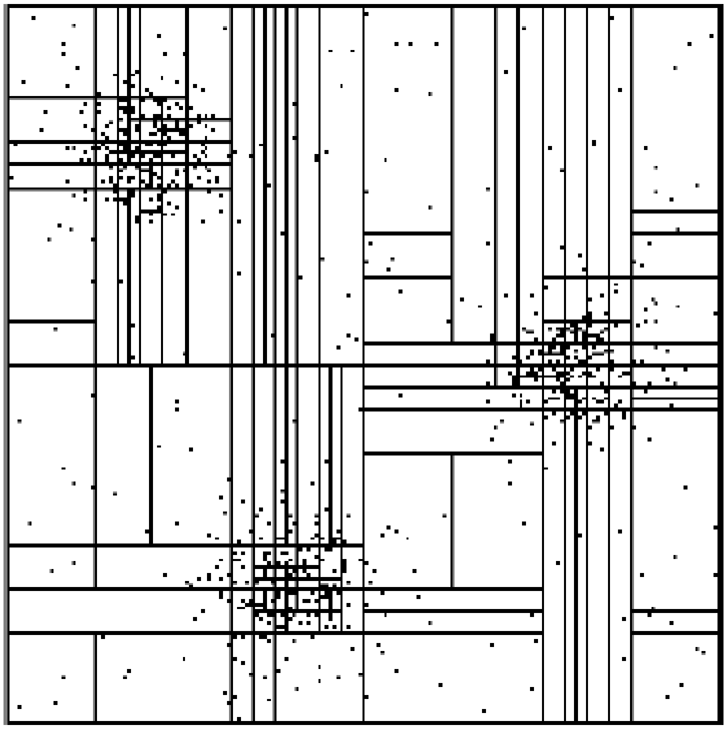


Figure 2.17, Grid clustering for 1000 patters in 3 clusters [31].

In [32] grid clustering is used for clustering data streams. Grid clustering is fast and applicable for data streams as it depends on the number of cells and not on the data itself. It can map infinite data streams to finite grid cells that summarize the information of the data streams as shown in Figure 2.18.

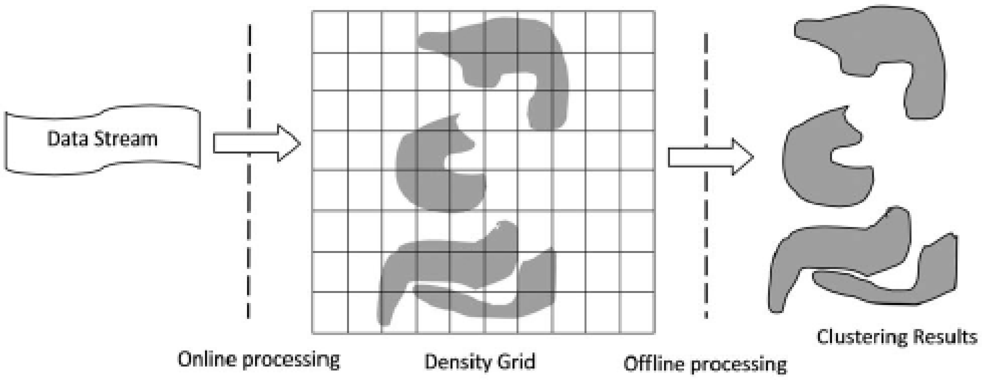


Figure 2.18, Grid clustering for data streaming [32].

Contemporary with our research but independently, in 2012, multidimensional grid structures are used in [33] for data classification in lower dimensional subspaces for small size data samples. The algorithm uses PCA and Multiple Discriminant Analysis (MDA) for dimensionality reduction. The projected feature space is divided equally along each eigenvector to build the grid structure. The algorithm generates 2D grids in subspaces constructed from the highest two eigenvectors. The first grid is generated based on a combination of PCA and MDA spaces using a grid size [5x5]. Another grid is constructed if a cell holds multiple classes. These fine grids are [10x10] in size. The fine grids use PCA in the second level and MDA in the third level.

The algorithm is tested with the IRIS benchmark data, which is described by 4 attributes holding three classes. The data set holds 150 examples, 100 of them is used in training and building the hierarchy rules. Figure 2.19 illustrates the 2D grids in the three levels.

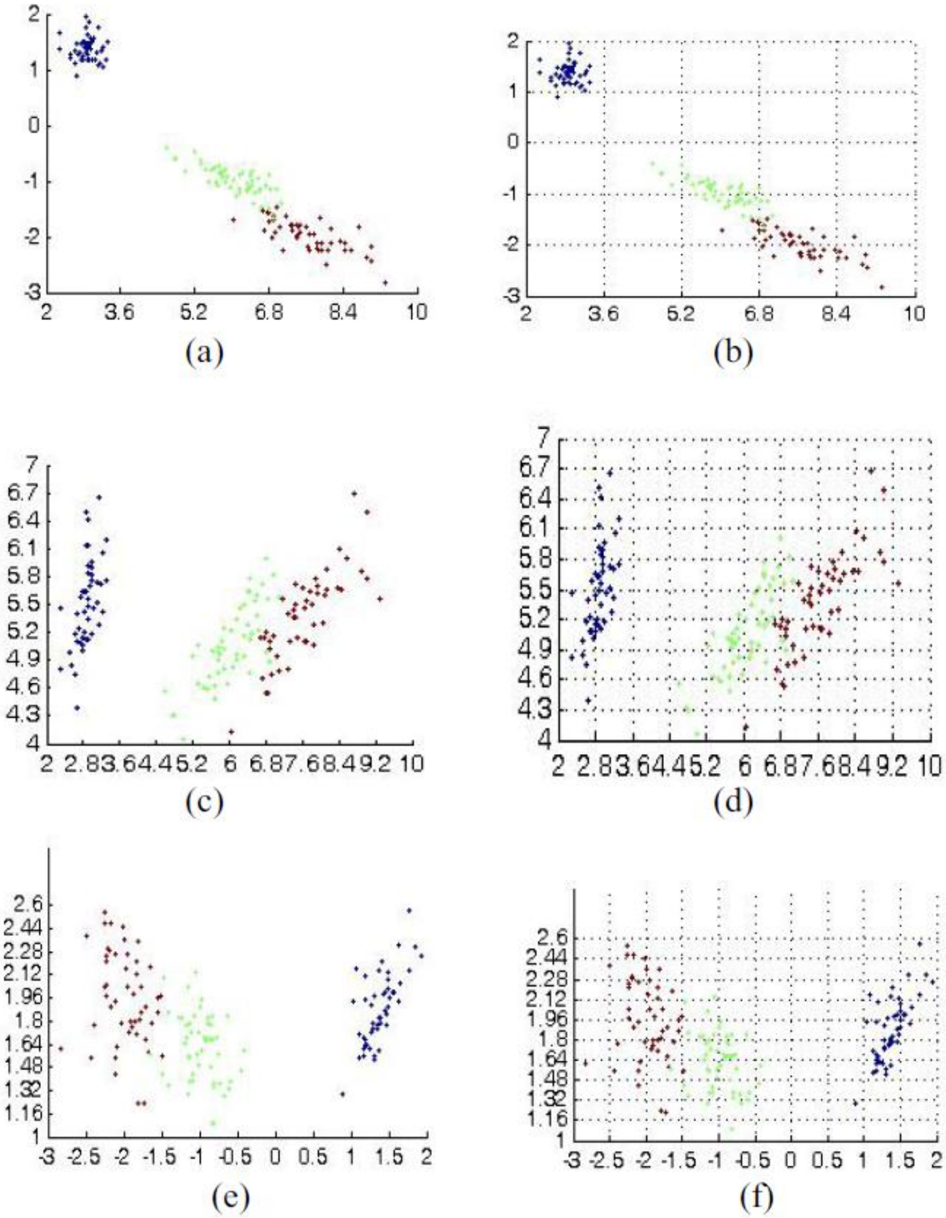


Figure 2.19, Illustration of fine grid construction using 2 eigenvectors [33].

MDA is a dimensionality reduction technique that is similar to PCA. However, MDA uses the class information to maximize the differences between the groups within the data. The separability is measured based on the mean values of all classes. This descriptive feature analysis can be achieved by maximizing the criterion function in Equation 2.6 [33].

$$J(v) = \frac{\left| W^t S_B W \right|}{\left| W^t S_W W \right|} \qquad (2.6)$$

Where *W* is the weight vector of a linear feature extractor. S_B measures the separability of class centers. S_W measures the within-class variance. S_B and S_W are given by:

$$S_B = \sum_{j=1}^{C} N_j .(m_j - m)(m_j - m)^T \qquad (2.7)$$

$$S_W = \sum_{j=1}^{C} \sum_{i=1}^{N_j} (x_i^j - m_j)(x_i^j - m_j)^T \qquad (2.8)$$

Where *x* is a set of feature vectors of training samples, *C* is the number of classes. N_j is the number of the samples of the j^{th} class. m_j is the mean vector of the j^{th} class, and *m* is grand mean of all examples.

MDA is better than PCA in some cases as it uses class information during building the lower dimensional space. However, PCA is better than MDA when the data is not uniformly distributed, as MDA uses the mean value of the classes assuming that the classes are spherical in shape [33].

2.4.5 Artificial Neural Networks

Artificial Neural Networks (ANN) have been used in artificial intelligence to simulate some properties of biological neural networks. A neural network consists of multiple processing units linked to each other with weights. By training the network on a set of example images, a new incoming object can be classified. Once learning is completed, the network weight set and the mapping function from input to output are determined. One of the disadvantages of using ANNs is to know the best network configuration prior to the training stage, where different configurations can give different results. Another disadvantage is that for new images the network has to be trained again [34].

Many different algorithms for supervised and unsupervised learning strategies have been proposed. The use of the Feed Forward Back Propagation (FFBP) training algorithm with a multilayer perceptron (MLP), which has one or more hidden layers, has been used by many algorithms for hand gesture recognition. Figure 2.20 shows an MLP network with one hidden layer. An MLP network with nonlinear activation functions can classify data which is nonlinearly separated using a hyper-plane as a decision boundary.

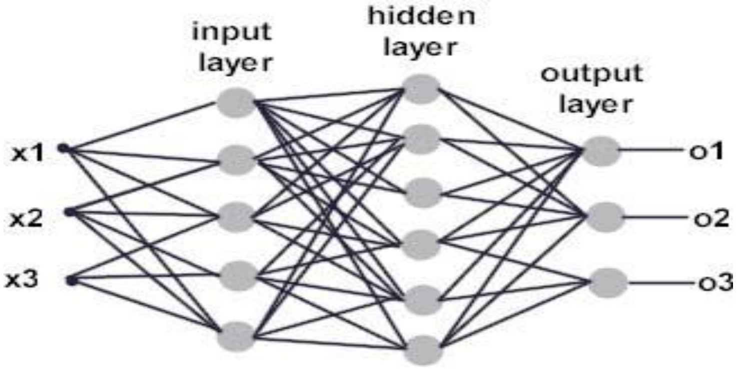


Figure 2.20, A three layer MLP architecture [36].

In [36], an intelligent human posture recognition system is proposed based on training ANNs on the silhouettes of five human body postures in order to understand human body actions and behaviour. The system is trained and evaluated using two different types of ANNs. The first is a supervised learning algorithm using MLP with the FFBP training algorithm. The second is an unsupervised learning algorithm using self organizing maps (SOM). The experimental results of the proposed system show, that the design with an MLP has a higher accuracy over the SOM under the specified training set.

In [37] ANNs are used for both preprocessing and the classification process for hand posture recognition. The system is able to recognize 15 static hand gestures, shown in Figure 2.22, using a data glove. The proposed system uses a FFBP supervised ANN for the preprocessing and SOM for the classification process as shown in Figure 2.21. ANNs are implemented using field programmable arrays (FPGA).

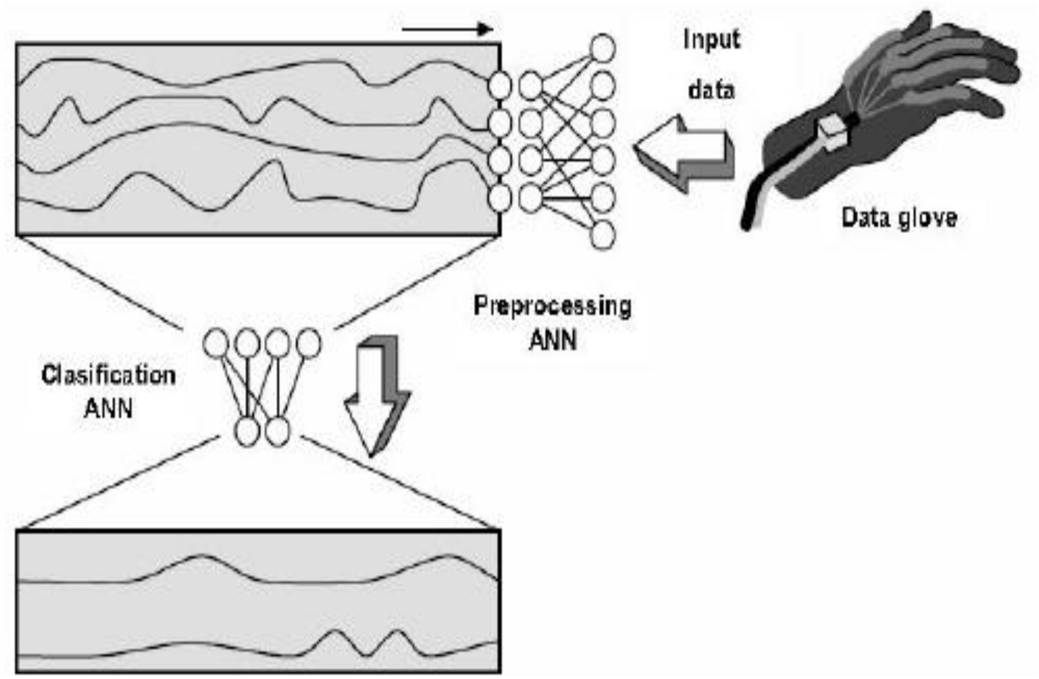


Figure 2.21, A data glove posture recognition system using ANN [37].

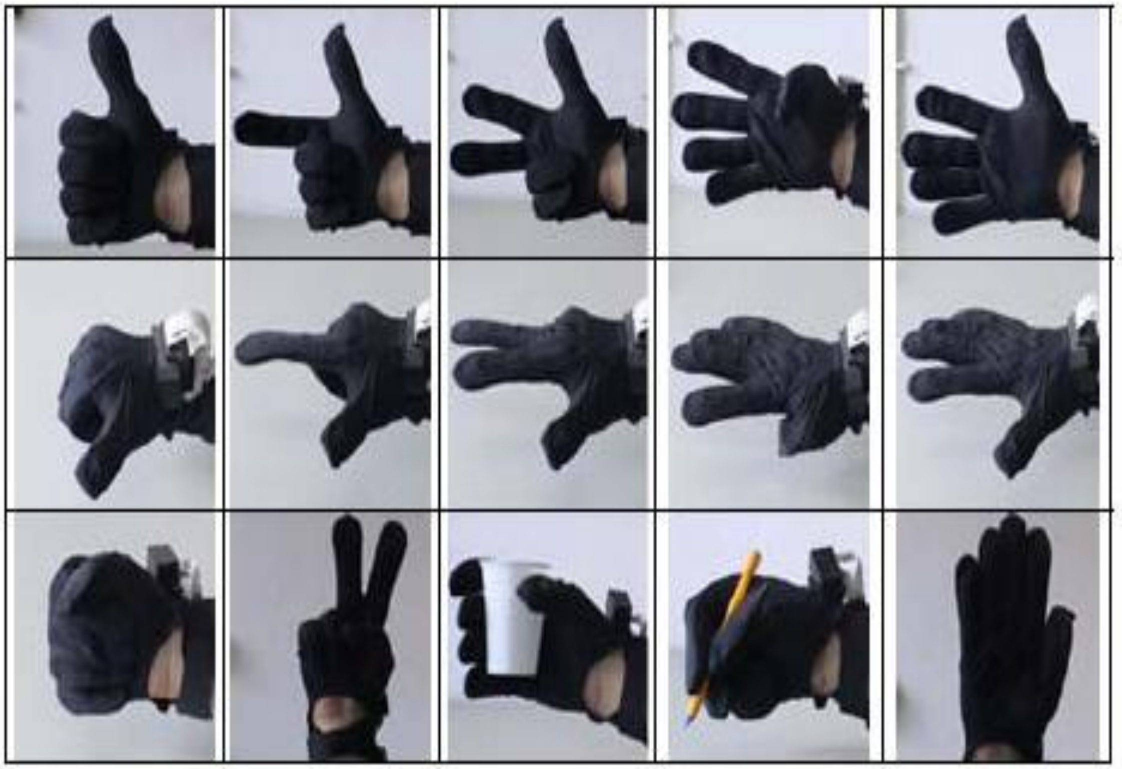


Figure 2.22, Fifteen postures using data gloves [37].

In [35], neural networks are used for real-time hand posture analysis based on a topological representation of the hand shapes to facilitate fingertip localization. A neural GAS algorithm is applied on the segmented hand area to obtain the topological representation. This representation is defined by nodes and connections in order to optimize clustering the input data. The neural Gas algorithm is an alternative to k-means clustering. It is a vector quantization technique that maps the input from topologically arbitrary data to a codebook representation. In addition to quantization, the algorithm provides a connection graph described by the network itself where the connectivity between the neural units reflects a priori unknown topological structure of the input data [54].

Figure 2.23 shows randomly selected nodes from a segmented hand as a start solution and the topological representation after applying the technique. This hand posture analysis algorithm is applied for a camera projector system shown in Figure 2.24.

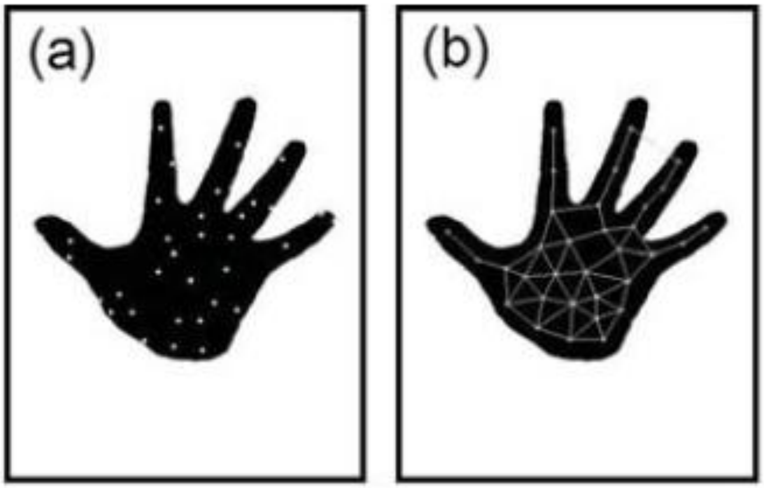


Figure 2.23, Topological hand representation using a neural GAS algorithm [35].

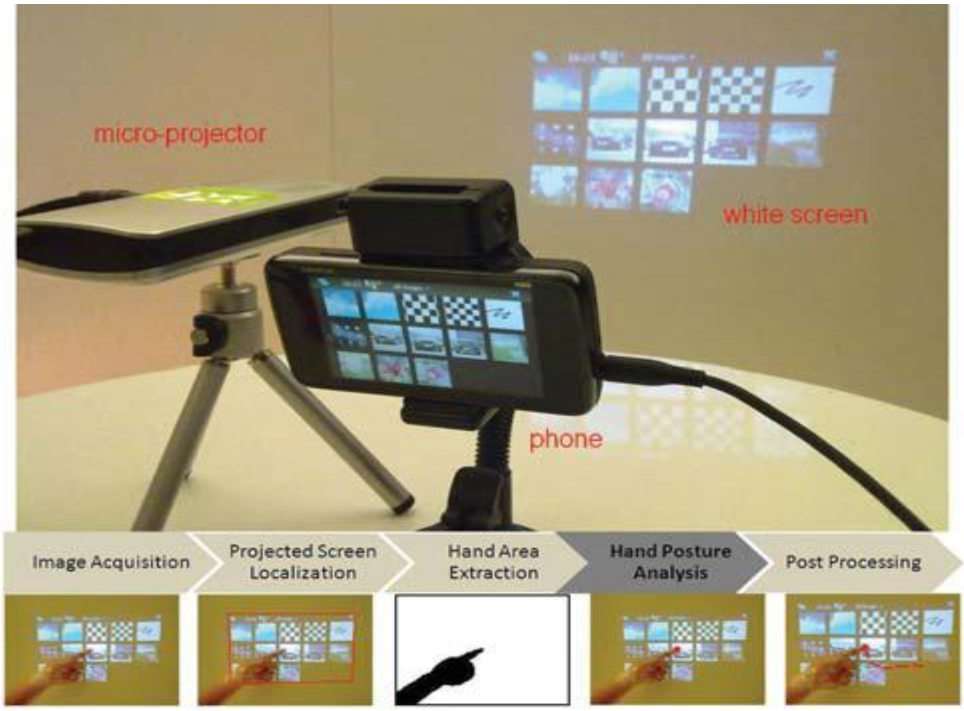


Figure 2.24, A camera projected system [35].

2.4.6 Support Vector Machine

A Support Vector Machine (SVM) is a learning machine for binary classification problems. Support vectors are the training examples, which are close to the decision boundary that separates two classes of data. These objects are in a critical place, where removing them would change the location of the separating hyper plane in a multidimensional space. The algorithm tries to build a hyper plane that represents the decision boundary between the two classes by maximizing the possible margin that separates the two classes preserving the minimum number of errors for the classification process in the training data [38,41,27]. The decision boundary from the SVM classifier is a soft margin hyper-plane in high dimensional space that separates most of the points for two classes. Figure 2.25 shows a decision boundary in 2D PCA space separating two classes using a SVM Matlab solver. In the 2D PCA space the decision boundary is nonlinear but in the high-dimensional space using a kernel it is linear.

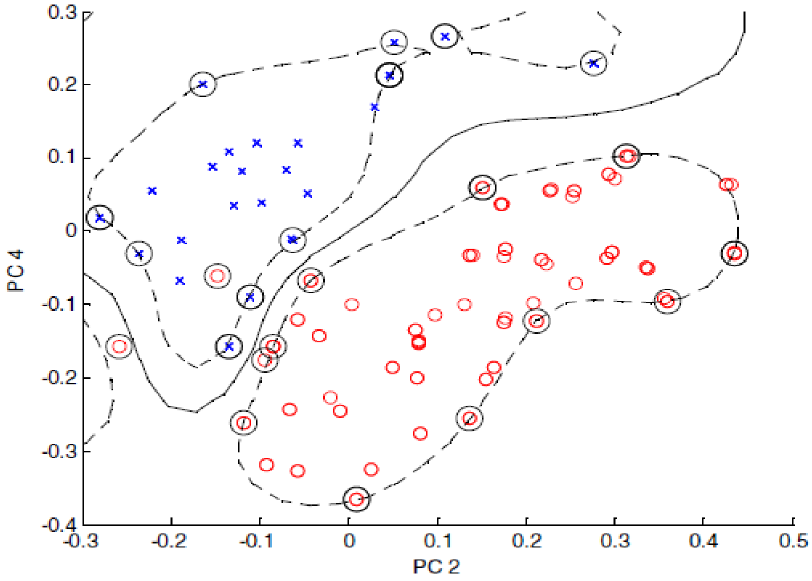


Figure 2.25, SVM decision boundary for the classification of two classes [27].

A SVM can be extended to a multiclass classification. It can be achieved through training using the one-versus-all rule. A SVM is built to separate each class from the rest of the data. A new test object can be classified according to the SVM that gives the highest response. [48]

The system in [39], presents a human computer interaction via hand gestures in real time performance. A multiclass SVM is used for the recognition task for hand postures. The system has a grammar to generate gesture commands to control an application. Figure 2.26 shows a flow diagram for the system. For classifying a new image, the skin region is detected and the face is subtracted from the product image. The Scale-invariant Feature Transform (SIFT) is used for feature extraction. SIFT is invariant to scale, orientation and partially invariant to illumination changes. These features are a set of key-point descriptors extracted from the image for matching. However, SIFT features are inefficient to use due to high dimensionality. Another problem is that the numbers of key-points are different from one shape to another where a SVM classifier requires a fixed size feature space.

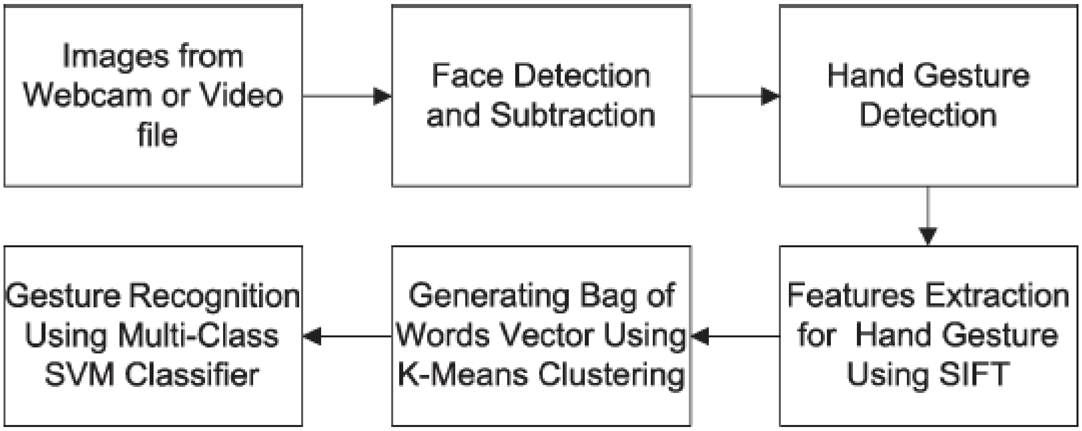

Images from Webcam or Video file
Face Detection and Subtraction
Hand Gesture Detection
Features Extraction for Hand Gesture Using SIFT
Generating Bag of Words Vector Using K-Means Clustering
Gesture Recognition Using Multi-Class SVM Classifier

Figure 2.26, A gesture recognition system using multiclass SVMs [39].

In order to solve this problem, a Vector Quantization (VQ) using the k-means algorithm is applied on the training examples as shown in Figure 2.27. This VQ maps the key-points of every test image into a unified dimensional histogram by encoding every key-point using the index of the cluster it belongs to. The size of the observed space is based on the number of clusters used in VQ. A multiclass SVM can be trained using the bag-of-words from the VQ for the training set with its class or label. SVMs are used to find the optimal hyper-plane that divides these vectors into the associative groups.

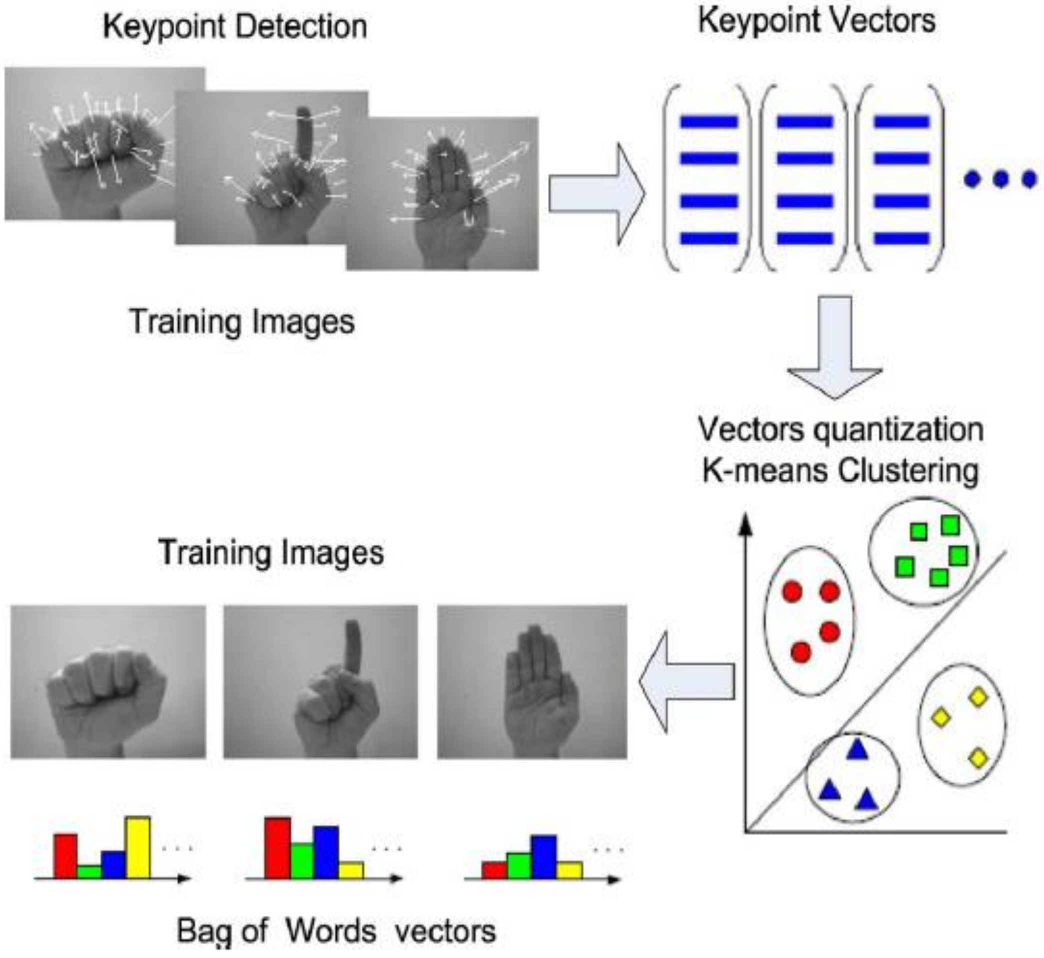


Figure 2.27, Key-points vectors quantization for SVM training [39].

In [40], multiclass SVMs are used for recognizing the pointing gesture on a 2D screen as shown in Figure 2.28. The SVMs are trained based on extracting the silhouette features of the video frames. The output from the SVM with the highest vote gives a location for a cell on the screen. Different SVM kernels are used during the training stage to adopt the best results. A radial basis function had the best results over the linear and quadratic polynomial kernels. The system uses one camera to capture the user position in front of the screen. Two positions for the camera settings are tested. The first is a side position to the right hand side of the user. And the second, which has a higher accuracy rate, is a frontal position to the top of the screen. The system has a restriction as the user and camera positions are fixed.

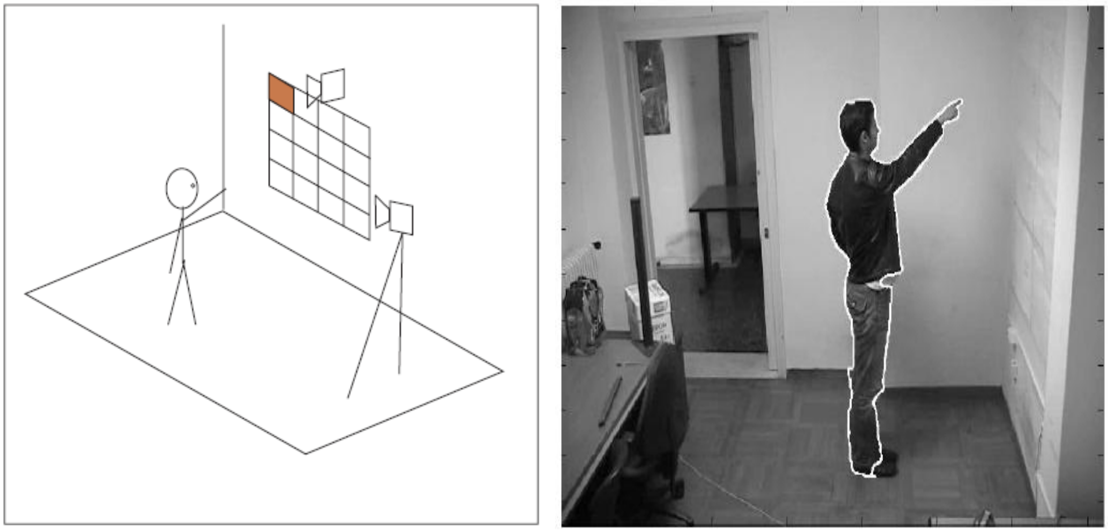


Figure 2.28, Pointing gesture recognition using SVMs [40].

2.4.7 Example-Based Approaches

Example-based approaches use a large database of hand images to represent the wide variations in the appearance of hand-shapes. Computer generated images can be used to provide training examples with known pose that densely sample the space of all possible orientations. These methods provide a powerful mechanism for directly estimating the pose by mapping the image to pose space by comparing a new image with a database of rendered samples. A limitation of current example-based approaches is the restriction to the poses or motions used in training. Extension to a wider variation of movements may cause difficulties due to data overlapping. Depending on the number of the examples, system improvements can be achieved by accelerating the search in the database [44].

In [45], example images are used to map hand shapes to the closest rendered images for the application of virtual reality. The system uses a specially designed coloured glove, as shown in Figure 2.29, to facilitate the segmentation and recognition of different hand poses.

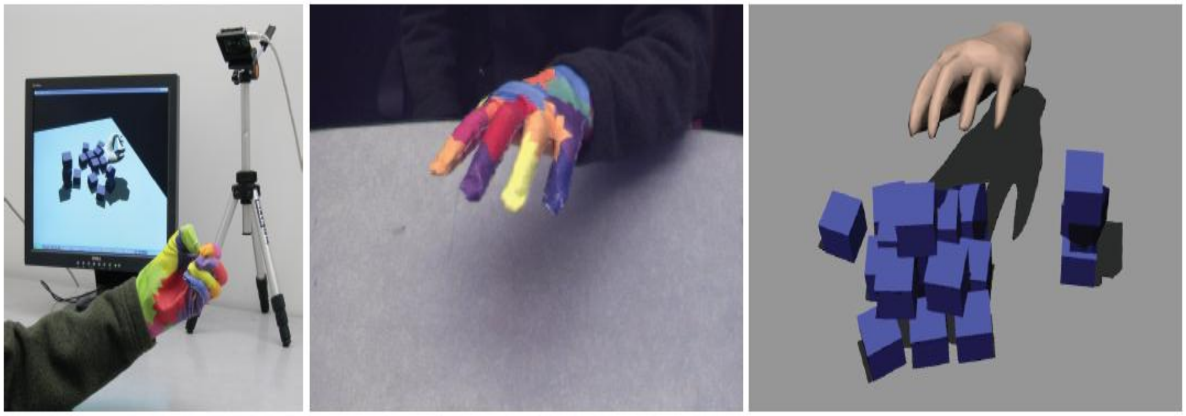


Figure 2.29, A demonstration of an example-based hand shape recognition system using coloured glove in virtual reality [45].

18000 finger configurations using a Cyber glove II are chosen as the sample for rendering the example images using a fixed hand model from the camera. The example images as well as the captured coloured glove image are rasterized and converted to a tiny image of size 40x40. Nearest neighbour search is applied on these tiny images to get the closest hand pose as shown in Figure 2.30. To accelerate the process of nearest neighbour search, a data compression is used to compress the database entries to a short binary code where Hamming distance is used as a distance measure.

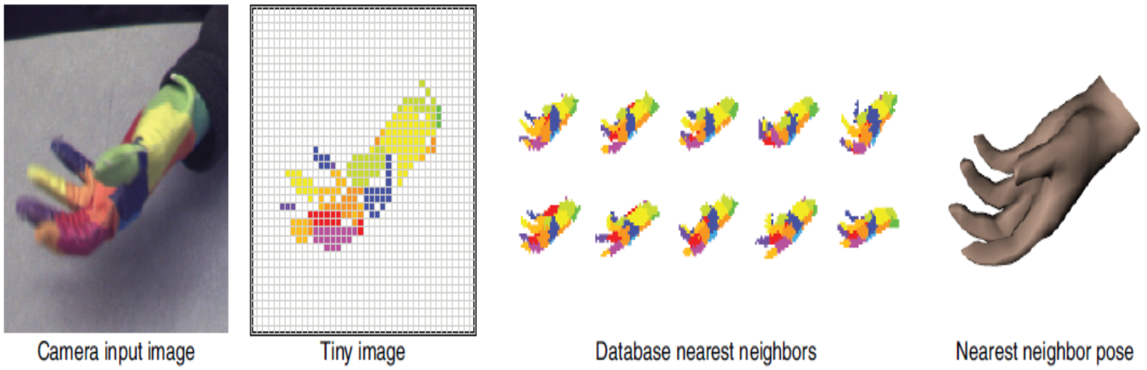


Figure 2.30, The retrieval of the nearest match using a database of tiny images for synthetic hand shapes [45].

In [46], a 3D hand pose estimation for cluttered images is achieved using a large database of 107,328 synthetic images for 26 shapes uniformly sampled at different viewpoints. The problem is considered as an indexing problem to retrieve the closest match, which the incoming image is labeled with. Database candidate matches are eliminated quickly using indexing. A database index is a data structure that improves the speed of the search process for information retrieval to prune the search space to a range query. R-tree, is one of the data structures that can handle *k*-dimensional data for a database indexing. R-tree is a height balanced tree with index records in its leaf nodes as pointers to the data objects. The search algorithm descends the tree from the root for the searching purpose. An update for the tree is essential to examine only the relevant data close to the search area [57].

Image-to-model mapping is obtained using binary edge images based on a chamfer distance measure. Chamfer distance provides an efficient way to measure shape similarity between two images after applying edge detection as a feature extraction method to preserve the shape information discarding the gray scale information [58]. Finger line-matching is appended as a second stage to improve the accuracy of matching. The feature vector for each edge image is represented by a set of points corresponding to edge pixels locations. The *X*-to-*Y* chamfer distance *C*(*X*, *Y*) is defined by Equation 2.9, where $\|a-b\|$ denotes the Euclidean distance between two pixel locations *a* and *b*. An example of the matching by the proposed system is shown in Figure 2.31. [46].

$$C(X,Y) = \frac{1}{|X|}\sum_{x \in X} \min_{y \in Y} \|x - y\| \qquad (2.9)$$

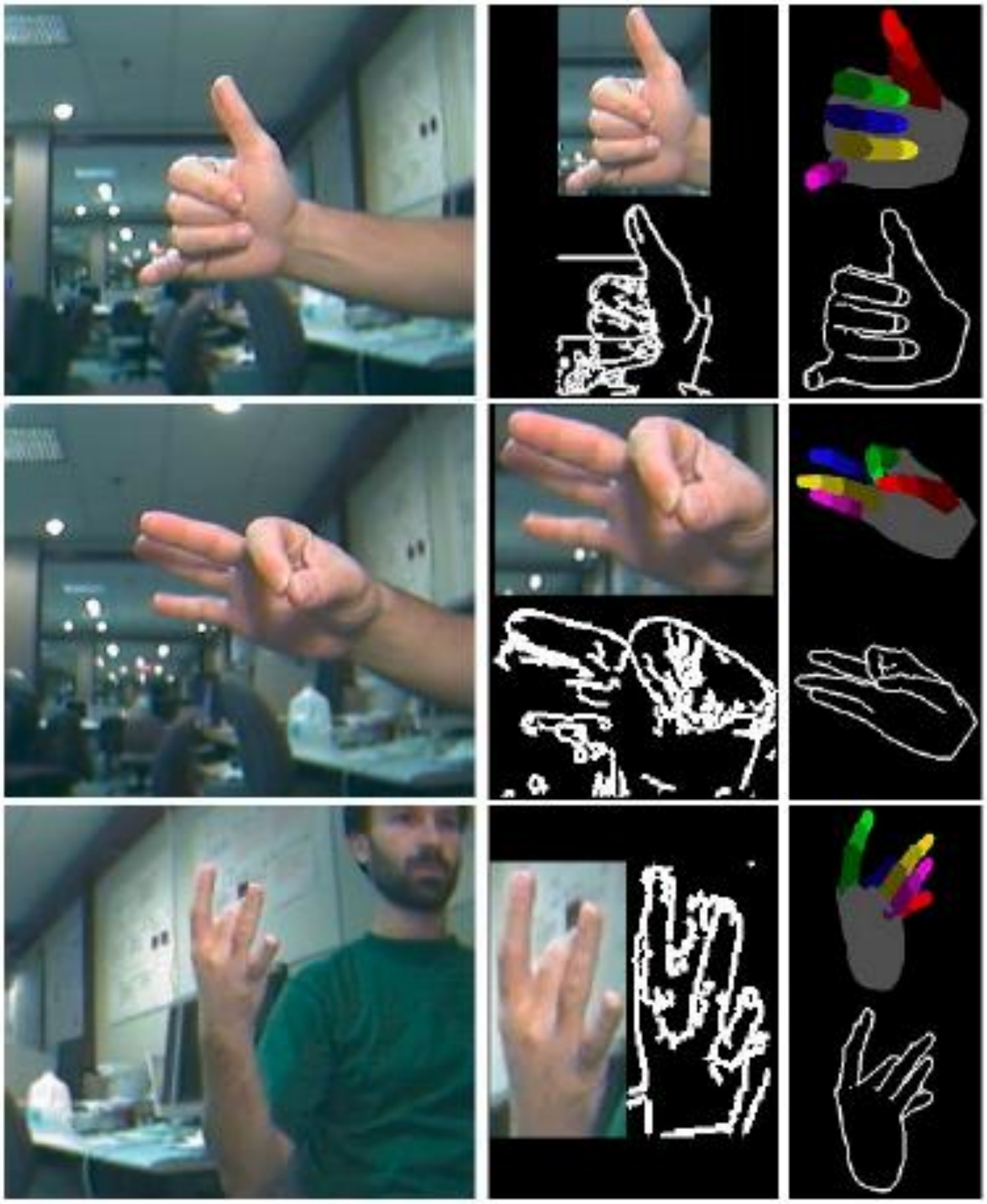


Figure 2.31, 3D Hand pose estimation from cluttered images using an example based classifier [46].

In [47], an example-based classifier is presented for 3D human body pose estimation as shown in Figure 2.32. The system assumes that the body is segmented from background, scaled and centred in the incoming image. The set of synthetic images contains 150,000 images generated by a computer animation package called ‘Poser’ using a model for the human body. Multi-scale edge direction histograms are used to represent the images. Hashing is used to solve the nearest neighbour search problem in a sub-linear time due to the large size of the database. However, the system performance and accuracy critically depend on the choice of the hashing functions. The training example images are hashed where collision is high for similar objects and small for dissimilar ones. For new images, an approximation to the k-nearest neighbour is achieved using hashing tables to retrieve the group of images that share the same index.

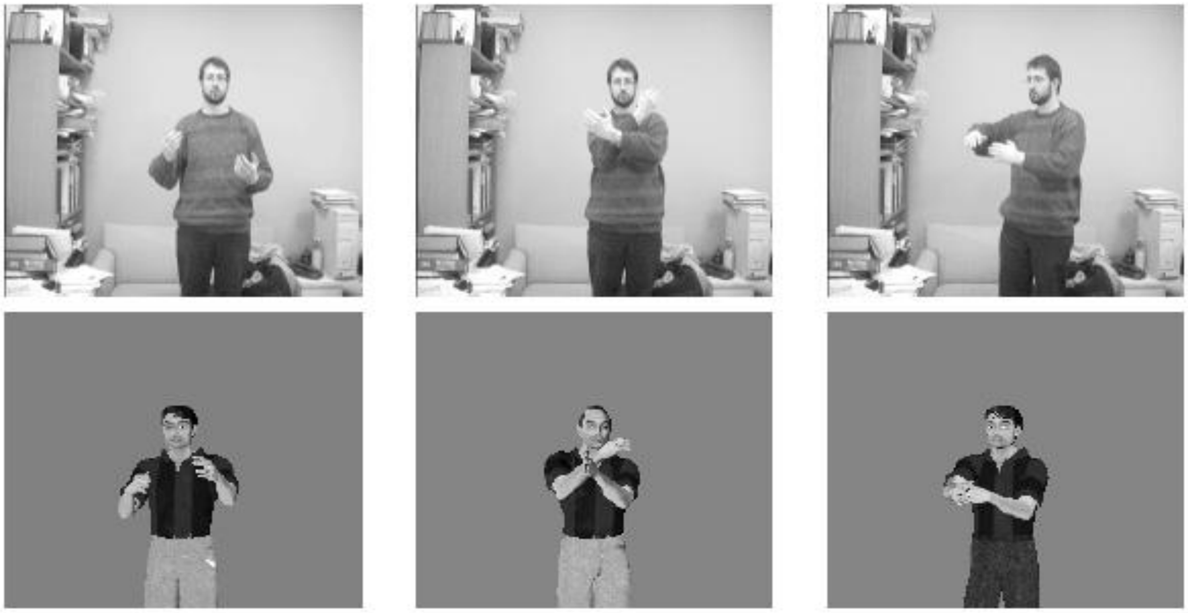


Figure 2.32, Top Match examples for mapping observed images to 3D synthetic body poses [47].

2.4.8 Image Blurring

The convolution of a kernel described by a Gaussian function with the pixels of an image is commonly called a Gaussian blur. Image blurring can be applied by convolving the image with a square array containing a two-dimensional Gaussian function [56]. The convolution with the Gaussian kernel is usually used as a low-pass filter, to filter noise from images that is inherent in the physical process of acquisition. It reduces the high frequencies from images and smoothes the edges and reduces small details.

Equation 2.10 is an equation for a Gaussian function in one dimension, where a is the height of the curve's peak, x_o is the position of the centre of the peak (also called the 'mean') and σ is the standard deviation, which can be regarded as a measure of the width of the bell function as shown in Figure 2.33.

$$f(x,y) = a\exp(-\frac{(x-x_o)^2}{2\sigma^2}) \qquad (2.10)$$

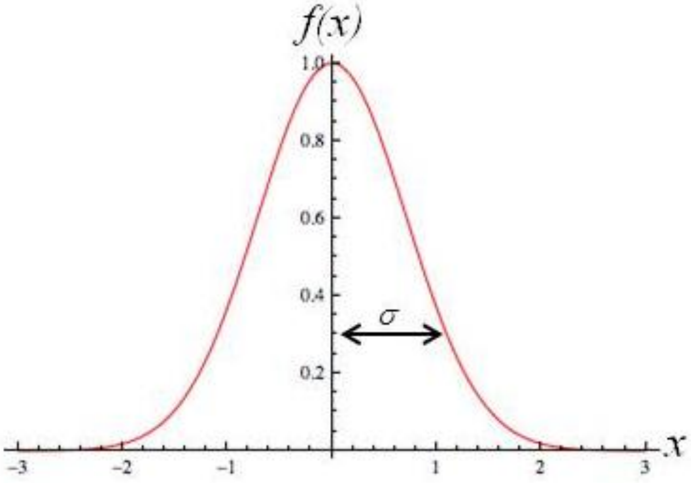


Figure 2.33, One-dimensional Gaussian function example.

The equation of a Gaussian function in two dimensions can be expressed as in Equation 2.11, where x_o and y_o are the coordinates of the centre or mean, and σ_x and σ_y are the standard deviations of the Gaussian distribution in the x and y directions respectively.

$$f(x,y) = a\exp(-(\frac{(x-x_o)^2}{2\sigma_x^2}) + (\frac{(y-y_o)^2}{2\sigma_y^2})) \qquad (2.11)$$

A two-dimensional Gaussian function with equal standard deviations in each dimension produces a surface whose contours are concentric circles about the mean as shown in Figure 2.34.

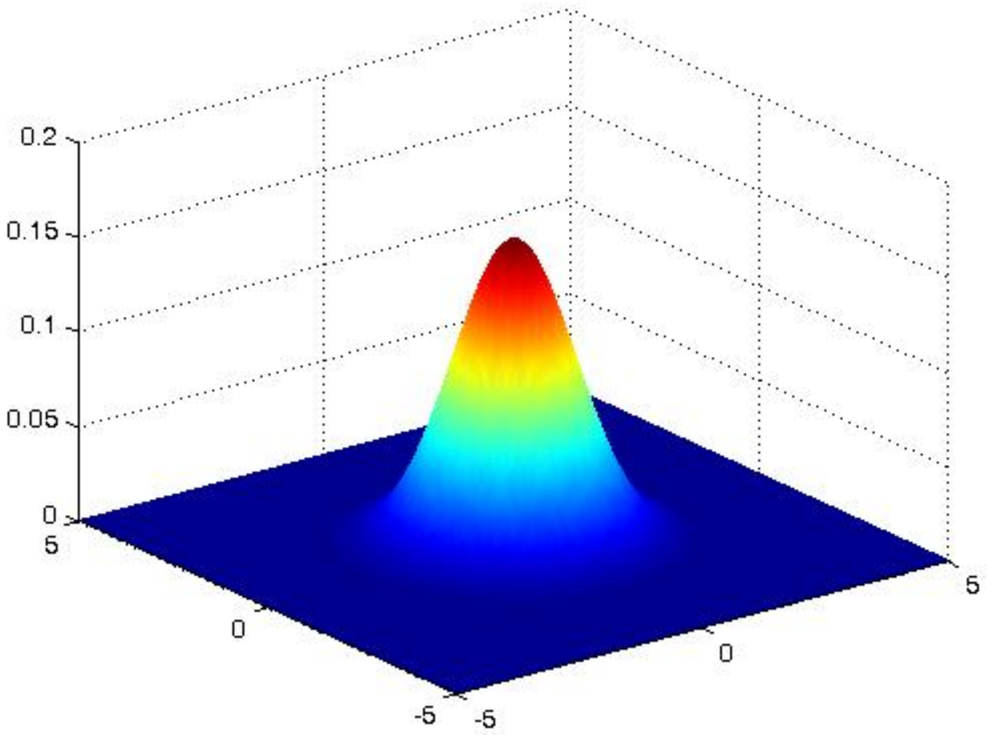


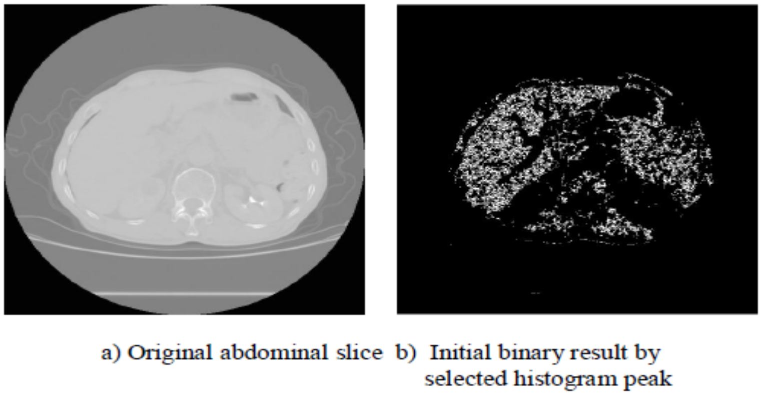
Figure 2.34, Two-dimensional Gaussian function example.

In image processing, values from a 2D Gaussian distribution are used to build a convolution matrix to be convolved with the original image. Each pixel's value is updated by the weighted average of neighborhood pixels. The original pixel's value receives the heaviest weight as it is convolved with the peak of the Gaussian function and the neighboring pixels receive smaller weights as they are far from the mean of the Gaussian distribution. A convolution function between an image and a Gaussian function can be defined by Equation 2.12 [67].

$$f * g[n] = \sum_{m=-\infty}^{\infty} f[m].g[n-m] \qquad (2.12)$$

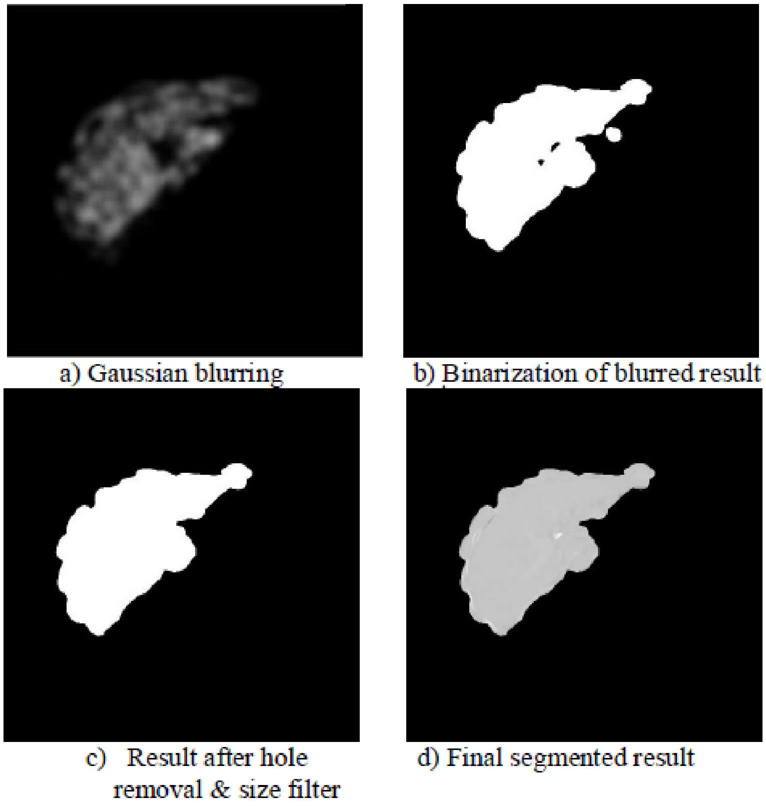
In [42] a recognition / reconstruction architecture using an ANN is presented. This system can recognize face images affected by various forms of blurring. The system uses a FFBP MLP network to classify and reconstruct the inputs. Networks are trained on original as well as Gaussian-blurred images to achieve a high robustness to different blurring effects.

In [43] a Gaussian blur filter is used to help in the automatic segmentation of liver from CT images. The algorithm follows 4 steps. First the liver part is segmented from the original scan image by analyzing its histogram, and then the liver part image is converted into a binary image as shown in Figure 2.34. Second, the binary image for the segmented liver part is blurred by a Gaussian kernel in order to connect isolated pixel clusters. The third step converts the blurred image into a binary image and any holes are removed. At the end, this result image can be used as a mask to segment the liver part from the original CT image as shown in Figure 2.36.



a) Original abdominal slice b) Initial binary result by selected histogram peak

Figure 2.35, Segmentation of liver part from abdominal slice [43].



a) Gaussian blurring

b) Binarization of blurred result

c) Result after hole removal & size filter

d) Final segmented result

Figure 2.36, Gaussian blurring for liver region extraction [43].

2.4.9 Nonlinear Manifold Learning Techniques

As linear PCA is not effective when the embedded data are nonlinear, in this section a review about nonlinear manifold learning techniques is presented including nonlinear versions from PCA, Locally Linear Embedding and Isomap.

2.4.9.1 Nonlinear Principal Component Analysis

Studying nonlinear manifolds using a manifold learning technique is an important issue in pattern recognition. A nonlinear phenomenon is characterised by a curve in the original data space. Nonlinear Principal Component Analysis (NLPCA) is a nonlinear generalisation of standard Principal Component Analysis, which can extract the principal components as curves rather than straight lines. NLPCA helps in visualising the nonlinear data as an aspect of data analysis by mapping the data from the original space to a component space using an artificial neural network [59].

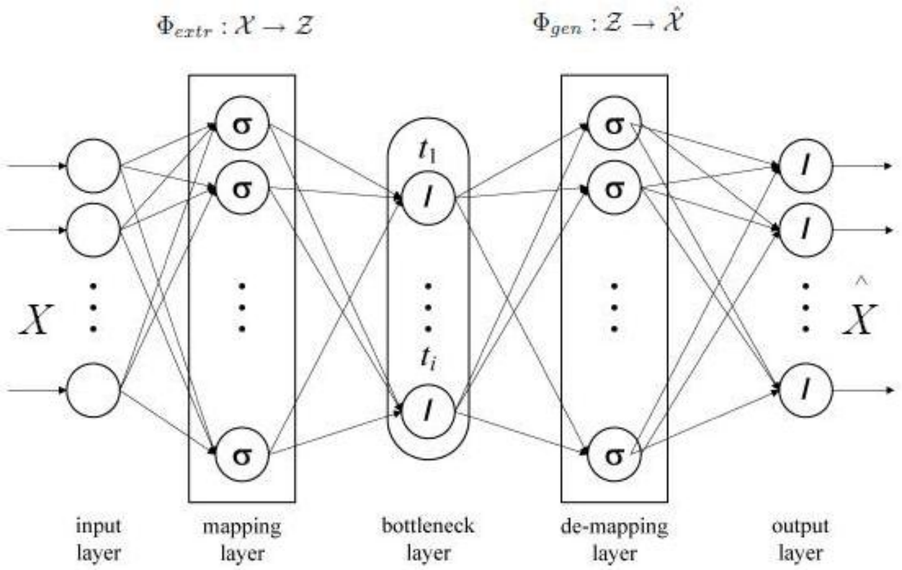


Figure 2.37, Bottleneck neural network topology [60].

NLPCA is based on learning a multi-layer perceptron with an auto-associative topology known as a bottleneck network to perform the mapping to the eigenspace. The network consists of five layers as shown in Figure 2.37. In the middle, the bottleneck layer consists of fewer units than the input or output layer where the data are compressed or projected into a lower dimensional space *Z*. The network can be considered to consist of two parts where a hidden layer in each part enables the network to perform the nonlinear mapping. The first part performs the extraction function $\Phi extr : X \rightarrow Z$ and the second part performs the generation or reconstruction function $\Phi gen : Z \rightarrow \hat{X}$. The learning strategy is based on enforcing the output $\hat{X}$ to equal the input *X* with high accuracy by minimizing the squared reconstruction error *E* in Equation 2.13 [59].

$$E = \frac{1}{2}\left\| \hat{X} - X \right\|^2 \qquad (2.13)$$

2.4.9.2 Kernel Principal Component Analysis

Kernel Principal Component Analysis (KPCA) is a nonlinear PCA that is computed by applying linear PCA after using a kernel function to map the original inputs into a high dimensional feature space. The linear PCA in the high-dimensional feature space corresponds to a nonlinear PCA in the original input space. The covariance matrix can be computed by Equation 2.14, where x_j is the j^{th} original input vector which is mapped into the high dimensional space using $\Phi(x_j)$ [55,61].

$$C = \frac{1}{N}\sum_{j=1}^{N}\Phi(x_j)\Phi(x_j)^T \qquad (2.14)$$

KPCA solves the eigenvalue problem in Equation 2.15 where λ_i is one of the eigenvalues and u_i is the corresponding eigenvector.

$$Cu_i = \lambda_i u_i \qquad (2.15)$$

The set of eigenvectors can be computed by Equation 2.16 where $\alpha_i = (\alpha_{i1},........,\alpha_{iN})$ are found by solving the eigenproblem in Equation 2.17.

$$u_i = \sum_{j=1}^{N}\alpha_{ij}\Phi(x_j) \qquad (2.16)$$

$$K\alpha_i = \lambda_i \alpha_i \qquad (2.17)$$

K is the Gram matrix defined by the dot product in the high dimensional space as described in Equation 2.18.

$$K(x_i, x_j) = \Phi(x_i).\Phi(x_j) \qquad (2.18)$$

The principal components for x_i are calculated by

$$S_j(i) = u_i^T\Phi(x_j) \qquad (2.19)$$

2.4 Locally Linear Embedding

Locally linear embedding (LLE) is another approach which addresses the problem of nonlinearity in manifolds. LLE is a manifold learning technique that maps the high dimensional data points to a global coordinate system preserving the relationships between neighbouring points. Each data point and its neighbours are assumed to lie on a locally linear patch of the manifold. The high-dimensional coordinates of each neighbourhood can be mapped linearly to global coordinates on the manifold. The nonlinear structure of the data can be identified through three steps [62], as shown in Figure 2.38.

1. Identify the neighbours of each data point. This can be done by finding the *k*-nearest neighbours, or by choosing all points within some fixed radius.
2. Compute a local geometry for each locally linear patch and finding the weights that best linearly reconstruct each point from its neighbours.
3. Build the low-dimensional embedding space which is best reconstructed by the weights determined in the previous step.

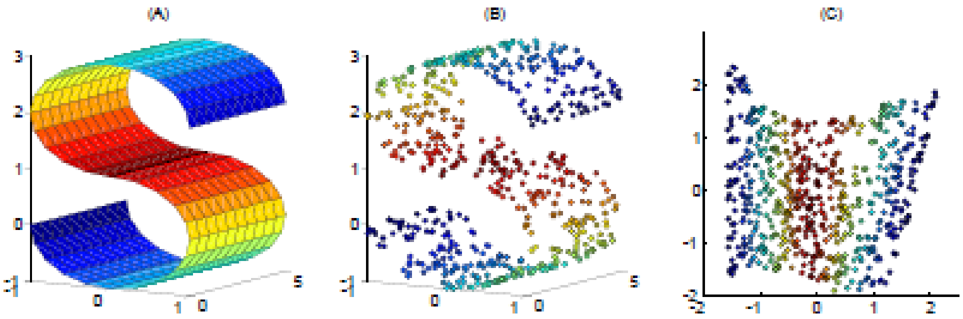


Figure 2.38, Locally Linear Embedding example [63].

2.5 Isomap

Isomap is a manifold learning technique that is based on computing the geodesic distances to represent the shortest paths between the different points along the curved surface of the manifold as if the surface is flat. The shortest paths can be approximately measured by moving through the curved surface by taking a sequence of short steps between neighbouring sample points. As in LLE, this can be performed by identifying the *k* nearest neighbours or by choosing neighbouring points within some fixed radius. A graph is constructed to represent the neighbourhood relations between the points. Dijkstra's algorithm [65] can be used to the find the shortest path distances in this graph to approximate the geodesic distances between all pairs of points on the manifold. Figure 2.39 shows a visualization for a data embedding using Isomap [62].

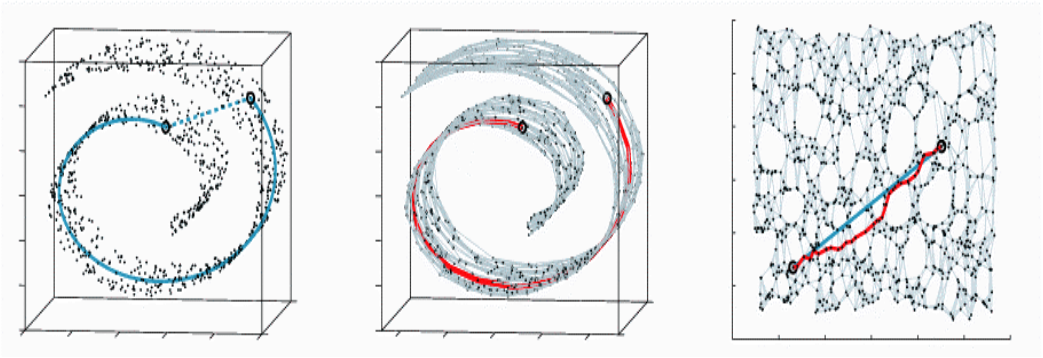


Figure 2.39, Isomap manifold learning Example [64].

Isomap applies MDS (Multidimensional scaling) as a dimensionality reduction technique in the geodesic space of the nonlinear data manifold to find a low-dimensional mapping that preserves these pair-wise distances. Isomap can be considered as a nonlinear generalization of classical MDS. Originally, MDS tries to minimize the error in Equation 2.20. MDS attempts to find *t* data points *Y* in a lower dimension space, such that $D^{(y)}$ is similar to the original distance matrix $D^{(x)}$ [62].

$$\min \sum_{i=1}^{t} \sum_{j=1}^{t} \left(d_{ij}^{(x)} - d_{ij}^{(y)} \right)^2 \qquad (2.20)$$

2.4.10 Publications in which my work is cited

In [49], our work is cited in this paper as one of the recent publications in the area of hand shape recognition. This paper uses the method of hierarchical pyramids in a different way from that presented in this thesis. It actually introduces a hierarchical pyramid sampling (HPS) method for sampling the hand images in order to reduce the effect of variations in the images of the same class and reduce the overlapping with images of different classes. At each sampling point, the proposed HPS gets the average window of pixels using a square and diamond as shown in Figure 2.34. The idea is tested under different variations in the test data including translation, rotation, and scale, giving a high accuracy measure up to 99% for the recognition task.

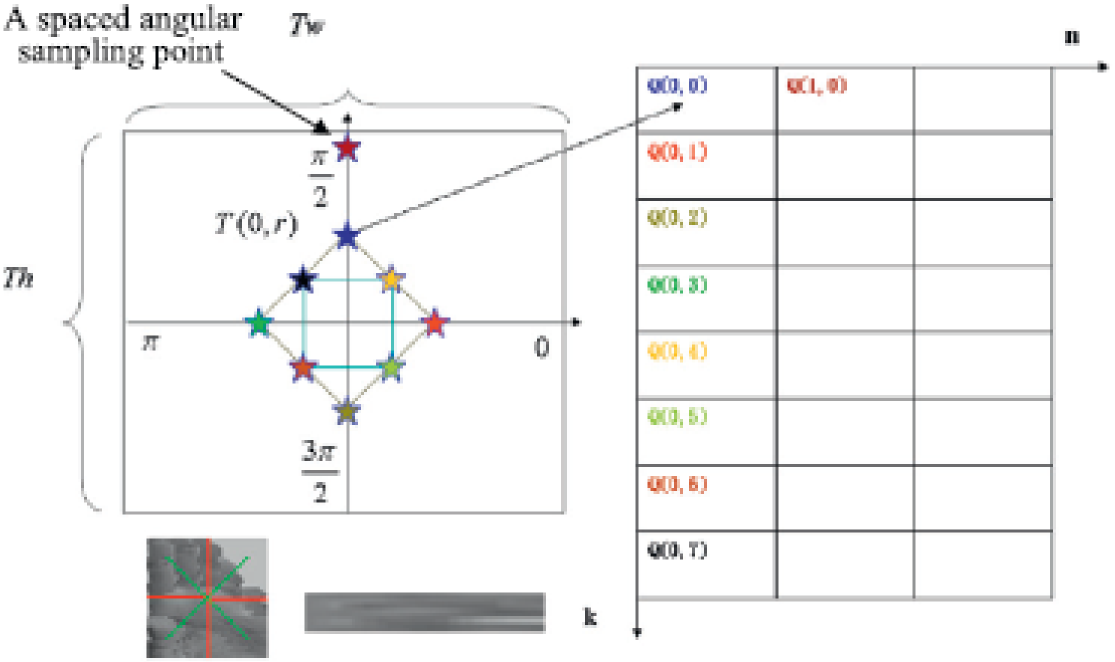


Figure 2.40, Hierarchical pyramid image sampling [49].

2.5 Conclusion

In this chapter, we have presented the literature review and the related work, which was used in this thesis. Gesture recognition is viewed as a natural approach to human computer interaction, where computer vision techniques can be used for the recognition task. Hand shape recognition is an interesting task in this process as the hand is a deformable object. Gestures can be viewed as a series of postures, which represent static hand shapes within a certain period of time. Gesture recognition methods are classified as either glove-based or vision-based. Glove-based methods can give enough information about the pose and shape of the hands using electronic sensors. However, they are expensive and are cumbersome to the user. Vision-based methods are either model-based or appearance-based. Model-based approaches are built based on the 3D kinematics of the hand. They offer a well-defined model to capture the details of the hand shape like fingertips. However, these models are computationally expensive to use. On the other hand, appearance-based approaches depend on visual information by extracting image features. Classification is achieved by comparing these features with the extracted image features from the input video.

The algorithms proposed in this thesis fall into the category of appearance-based approaches. Through this chapter, a review about the related techniques that are used to build the thesis is given. Data pyramids are used as a data structure to analyze images through a multi-scale resolution hierarchy. One of the applications for the data pyramids is a coarse-to-fine search for a pattern that may appear at any scale within an image. Principal Component Analysis is a statistical method that can be used for feature extraction and dimensionality reduction. PCA is a linear method that can visualize manifolds by extracting the embedded low-dimensional data within a high-dimensional space. PCA has some disadvantages. It is not robust to transformations and does not work well with nonlinear manifolds. Perpendicular distance can be used to measure the distance between an image and a manifold in an eigenspace. However, this distance measure assumes that the surface of the manifold is flat. Multidimensional Grids can organize the space to cluster the data into groups of similar objects. The cells of a MDG are hyper-cubes in a multidimensional space. The information within these cells can be used for further analysis. Artificial neural networks are used to simulate biological neurons in order to achieve the learning process. ANNs are used for shape recognition. However, it is difficult to know the best architecture design prior to the training stage. Support vector machines are used to build binary classifiers. The decision boundary from the SVM classifier, is a soft margin hyper-plane that separates most of the points for the two classes. Image blurring can be used as a low-pass filter by convolving the images with a Gaussian function. Gaussian blur is commonly used for image smoothing and reducing details between images. Example-based approaches are reviewed as a method to build a large training set of computer-generated example images. This training set can densely sample the space to provide the variability in the hand shapes. The images can be easily generated to pre-defined orientations where each image can be labeled according to its pose parameters. Indexing is used by many example-based algorithms for accelerating the search process. However, it is sensitive to the choice of the hash functions and the size of the hash tables.

Chapter 3

Proposed Algorithms

3.1 Introduction

In this chapter four algorithms are introduced to present different multistage hierarchical strategies for hand shape recognition using Principal Component Analysis as a dimensionality reduction and feature extraction method. The proposed algorithms are based on using image blurring to reduce the nonlinearity in the manifolds generated by applying PCA to a set of example images using a computer-generated model. Flattening the space helps in classifying different hand shapes images more accurately where the images in the original space are intersected at different orientations. The proposed algorithms present an example-based strategy where the training set contains 111320 images for 20 shapes from the Irish Sign Language at different orientations.

The proposed multistage hierarchy analyzes new patterns by projecting them into the different levels of a data pyramid using different PCA spaces. Each PCA space is computed for a group of images that represent a certain shape or a certain pose. The four algorithms study the problem of hand shape recognition using different pattern recognition techniques. The first algorithm is based on using perpendicular distance to measure the distance between the incoming object and the nearest manifold. The second algorithm is based on using multidimensional grids to divide the space in a supervised manner into cells of similar objects according to the shape or pose of the hand. The third algorithm uses unsupervised multidimensional grids to cluster the space into cells of close objects with different shapes and orientations. The fourth algorithm is based on training a set of simple-architecture multi-layer neural networks in each level of the pyramid using different PCA spaces by the Feed-forward Back-propagation training algorithm.

3.2 Example-based Data Set

A dataset of computer-generated images of a human arm and hand is used as a training set. The images were generated by a software called Poser [68]. The dataset represents 20 different hand shapes from the Irish Sign Language alphabet as illustrated in Figure 3.2. The images hold the hand and arm of a signer against a uniform black background. Computer-generated images are used as it is possible to control the position and orientation of the hand and arm at regular intervals. The whole space of possible hand shapes, translations, and rotations can be covered. For a living human being it is difficult to control the hand to this level of accuracy. The example images densely sample the space. The signer arm is rotated from +90 degrees to +180 degrees in increments of 2 degreest. Figure 3.1 shows Sign ‘A’ in the horizontal and vertical position. In the dataset, each shape is translated 5 pixels in the horizontal and vertical position creating 121 translated objects from the original position forming a data set of 111320 images. Poser can easily generate thousands of images for different shapes and poses labeled by their shape, rotation and translation position. The next figure shows Sign “A” in the horizontal and vertical positions.

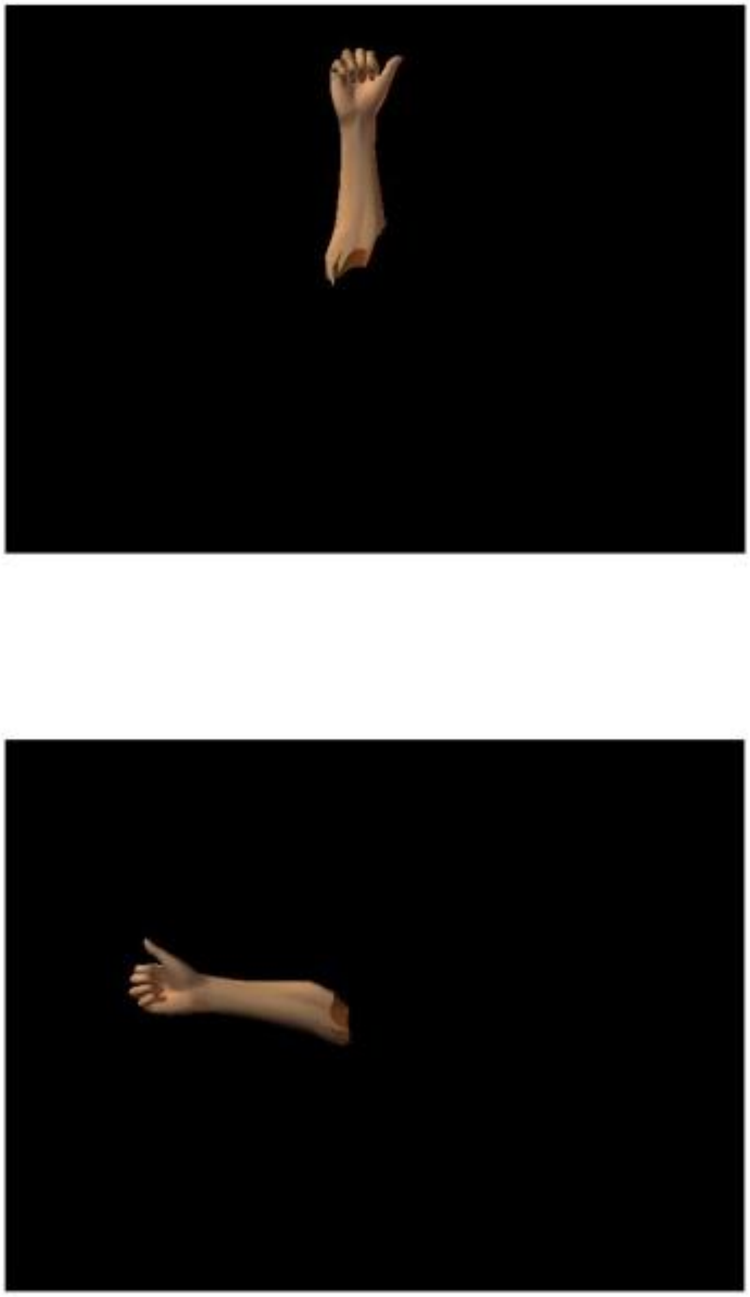


Figure 3.1, Computer-generated images for sign “A” in the Irish Sign Language at the horizontal and vertical position.

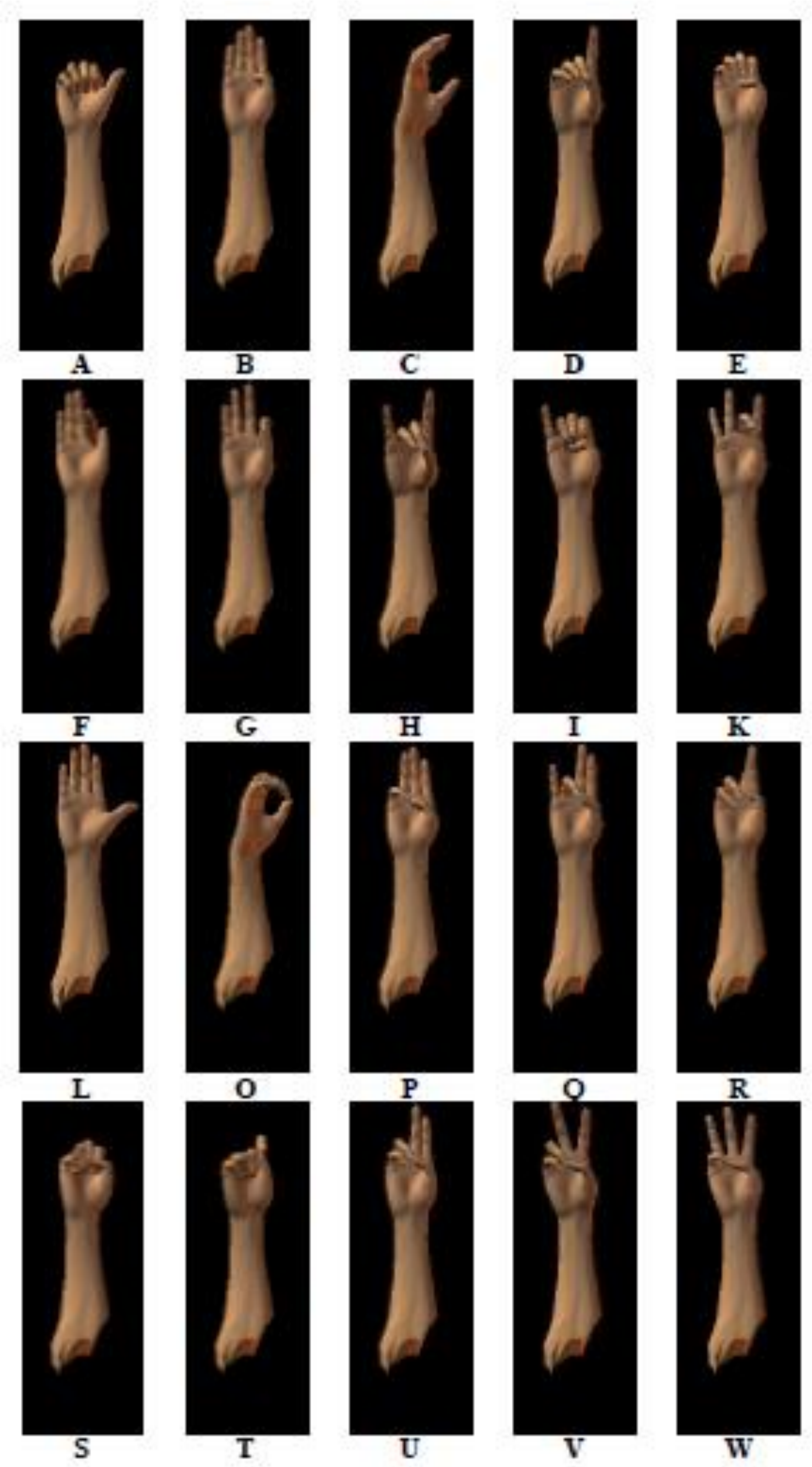


Figure 3.2, Computer generated images for 20 shapes from the Irish Sign Language.

3.3 PCA Manifolds

The proposed algorithms are based on calculating a series of manifolds to represent the different shapes, rotations and translations for the signer arm and hand. Applying PCA to the set of example images generates these manifolds. PCA is used as a dimensionality reduction and features extraction method. The eigenvectors with the largest eigenvalues represent the directions in feature space along which there is the greatest variation in the data. For example, three eigenvectors with the largest eigenvalues can be chosen to form the basis of a low-dimensional space. The images are projected into the eigenspace using the sufficient number of eigenvectors and each image is represented by a point in that eigenspace. As pre-processing usually is computationally expensive, the input to the PCA is the original images without any pre-processing

“Translation manifolds” are generated by applying PCA on a set of 121 images that represent the translations 5 pixels in each direction for a certain shape at a certain rotation of the signer arm. At each rotation angle, a set of 20 translation manifolds are generated to represent 20 different hand-shapes. As the range of angles to be represented increases, the number of manifolds increases and consequently the space of manifolds to be searched becomes larger. Our objective is to classify the hand-shape and estimate the in-plane rotation angle and the translation position in an efficient and accurate way. Figure 3.3 shows an eigenspace for a translation manifold for sign “A” at a certain rotation angle using the highest three eigenvectors.

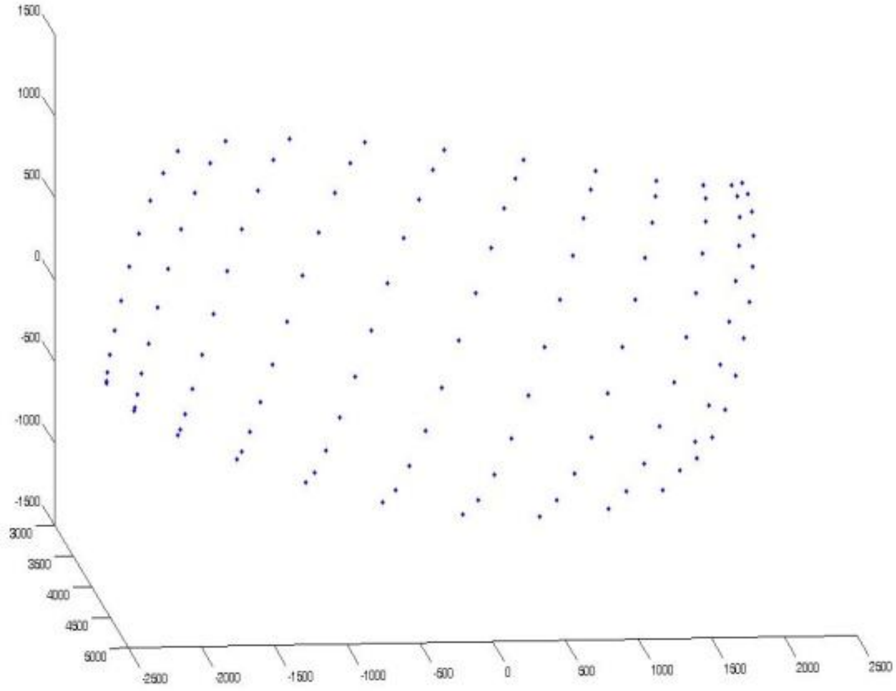


Figure 3.3, A 3D translation manifold for sign “A”

A “rotation manifold” is generated by building one eigenspace for a certain shape at a range of different rotations. A rotation manifold is constructed by applying PCA directly to the images for a certain shape at a range of rotation angles and at a range of 121 translations for each angle. These objects share the same eigenspace. Figure 3.4 shows a rotation manifold for sign “A”.

“Rotation sub-manifolds” can be constructed as separate manifolds within the rotation manifold eigenspace. A rotation sub-manifold is created for each rotation angle. The sub-manifold for a particular rotation angle is generated by applying PCA to the set of 121 points in the parent eigenspace for that angle. Figure 3.5 shows the pseudo-code for generating rotation sub-manifolds.

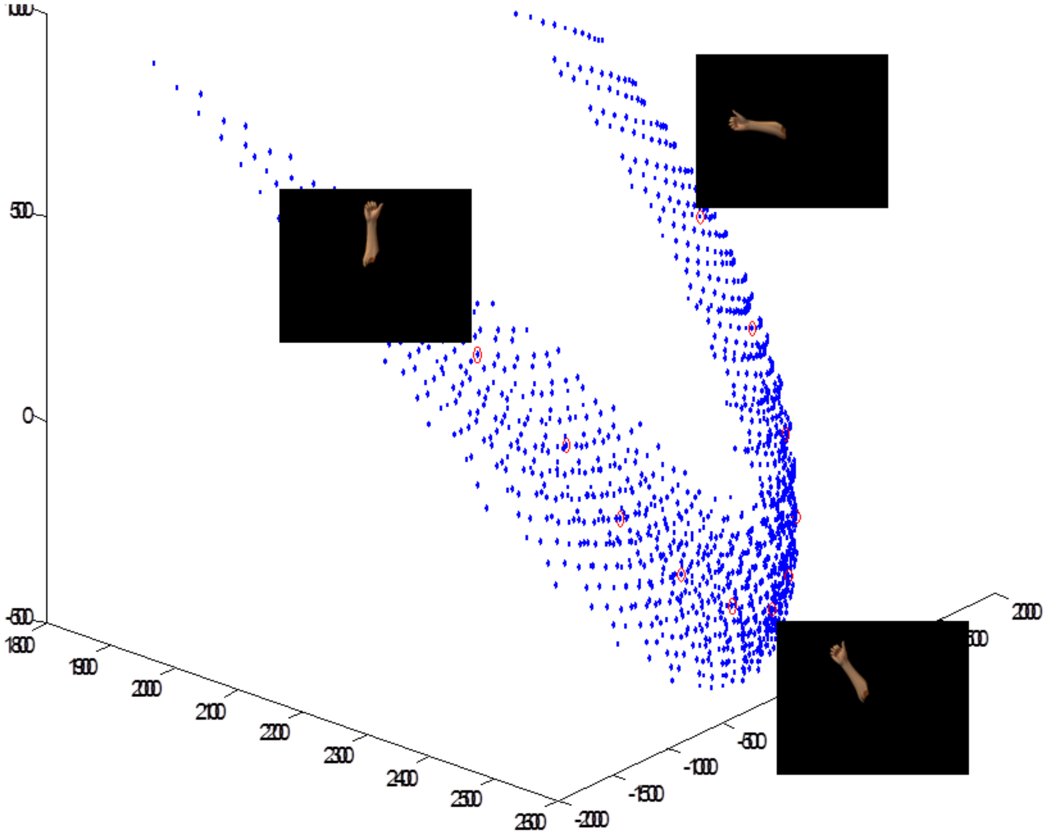


Figure 3.4, A rotation manifold for sign ‘A’

```
To Construct a Rotation Manifold
R = set of rotation angles from horizontal to vertical position for shape S
Z=[ ]
for each $r \in R$ do
        z = set of 121 translated images of shape S at rotation angle r
        Z = [Z;z]
end
apply PCA to Z to obtain eigenvectors V
project Z onto V to obtain the set of points P, which represents the rotation manifold

To Construct Rotation Sub-manifolds
for each $r \in R$ do
        z = sub-set of points P at rotation angle r
        apply PCA to z to obtain eigenvectors v
        project z onto v to obtain the set of points p, which represents a rotation sub-manifold at r
end
```

Figure 3.5, Pseudo-code for constructing rotation sub-manifolds.

A “shape manifold” is a manifold that holds different shapes within the same eigenspace. It is constructed by applying PCA directly to the images of the 20 shapes at the same angle of rotation. Figure 3.6 shows a shape manifold for 2420 objects representing 121 translated objects for 20 shapes at the same rotation angle. These objects are projected into the same eigenspace.

“Shape sub-manifolds” can be constructed as separate manifolds within a shape manifold eigenspace. A shape sub-manifold is constructed for each shape, as follows. PCA is applied to the set of 121 projected points within the parent eigenspace for a particular shape to get the sub-manifold for that shape. Figure 3.8 shows the pseudo code for generating shape sub-manifolds.

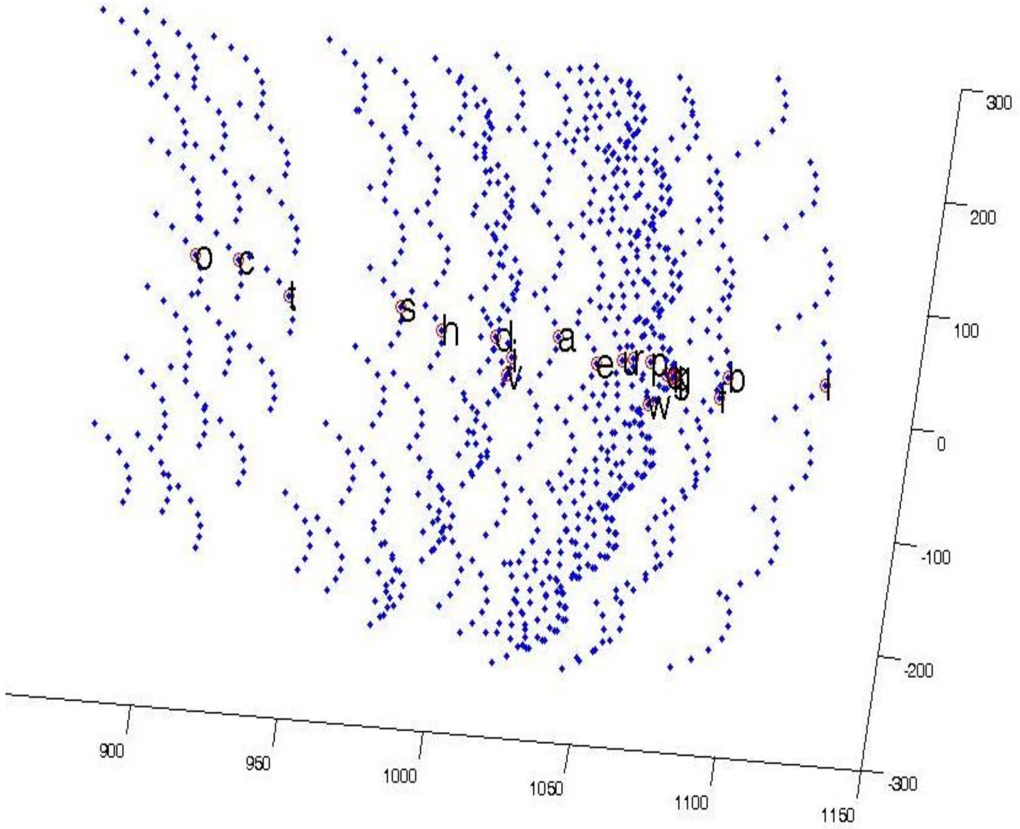


Figure 3.6, A shape manifold, the associated shapes are shown in Fig. 3.7.

The order of the shapes within this manifold is quite interesting. It illustrates how PCA extracts the underlying structure within the data. Hand shapes which are close together in the manifold have similar images. For comparison Figure 3.7 shows the images for the different hand shapes based on the sequence that appears in the shape manifold in Figure 3.6. The sequence starts with "O" which is a closed compact shape and ends with "L" which is a broad open shape. It is apparent that similar shapes are grouped together.

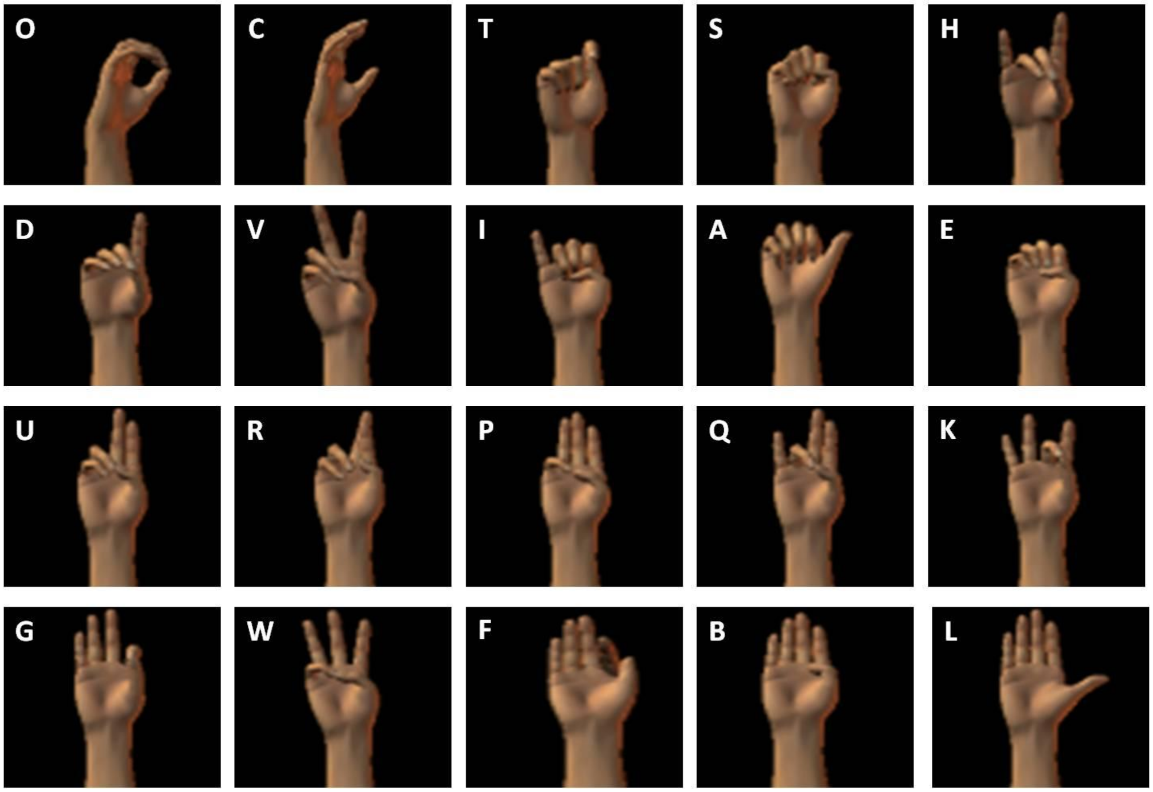


Figure 3.7, Computer-generated images for 20 shapes in the Irish Sign Language in the order in which they appear in Figure 3.6.

```
To Construct a Shape Manifold
S = set of 20 shapes at rotation angle R
Z=[ ]
for each $s \in S$ do
        z = set of 121 translated images for shape s at rotation angle R
        Z = [Z;z]
end
apply PCA to Z to obtain eigenvectors V
project Z onto V to obtain the set of points P, which represents the shape manifold

To Construct Shape Sub-manifolds
for each $s \in S$ do
        z = sub-set of points P for shape s
        apply PCA to z to obtain eigenvectors v
        project z onto v to obtain the set of points p, which represent the shape sub-manifold for s
end
```

Figure 3.8, Pseudo-code for constructing shape-sub manifolds.

3.4 Nonlinearity Reduction using Image Blurring

The proposed algorithms explore the effect of image blurring on the classification process using a Gaussian Kernel. A Gaussian blur can be used to reduce image noise and to reduce details from images. As blurring has the effect of removing small changes between objects, the generalization factor for the classification process increases and it becomes easier to classify new incoming objects. By blurring the training set as well as the incoming object at the same level, it is possible to classify it using a suitable distance measure.

The effect of blurring on a computer-generated image for sign ‘A’ is shown in Figure 3.9. Although blurring removes some of the details of the hand shape (e.g. the thumb in Figure 3.9 is reduced, making it more similar to the sign ‘E’), this is compensated by the positive effects of blurring, such as making manifolds more linear and more separable which will be demonstrated in the rest of this section.

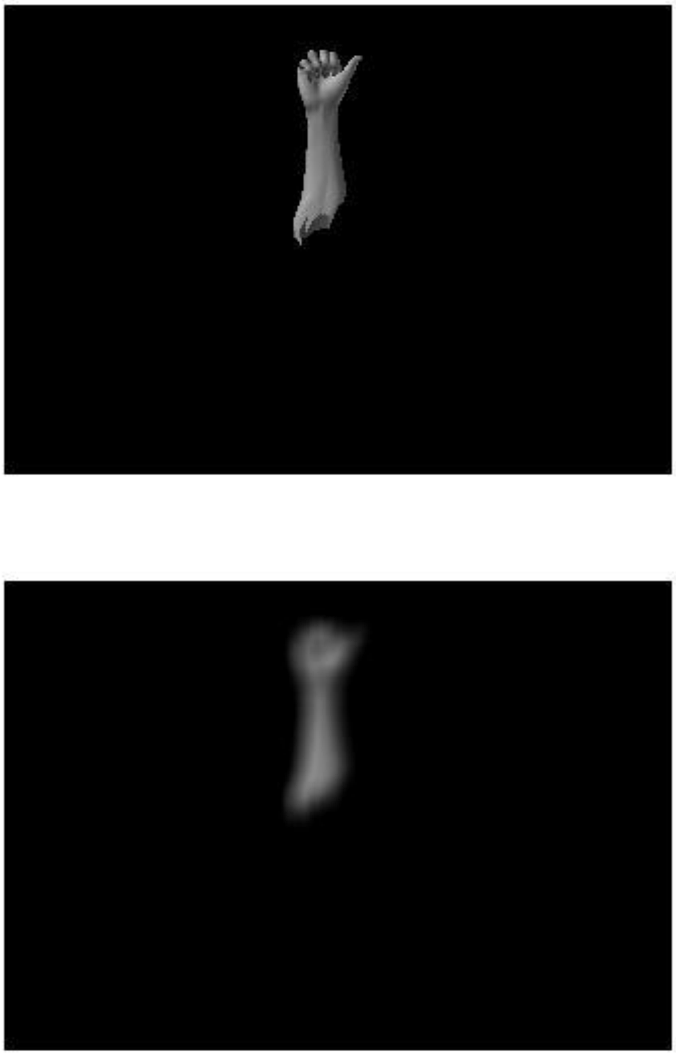


Figure 3.9, The effect of blurring on a computer-generated image for sign ‘A’ in the Irish Sign Language.

Figure 3.10 shows two translation manifolds for sign ‘A’ (projected onto the 2nd and 3rd eigenvectors) before and after blurring. Each point in the figure is generated by a translated version of ‘A’. From the figure it can be seen that image blurring has an effect on the structure of the translation manifolds. Image blurring reduces the nonlinearity of the translation manifolds and makes the manifolds more flat.

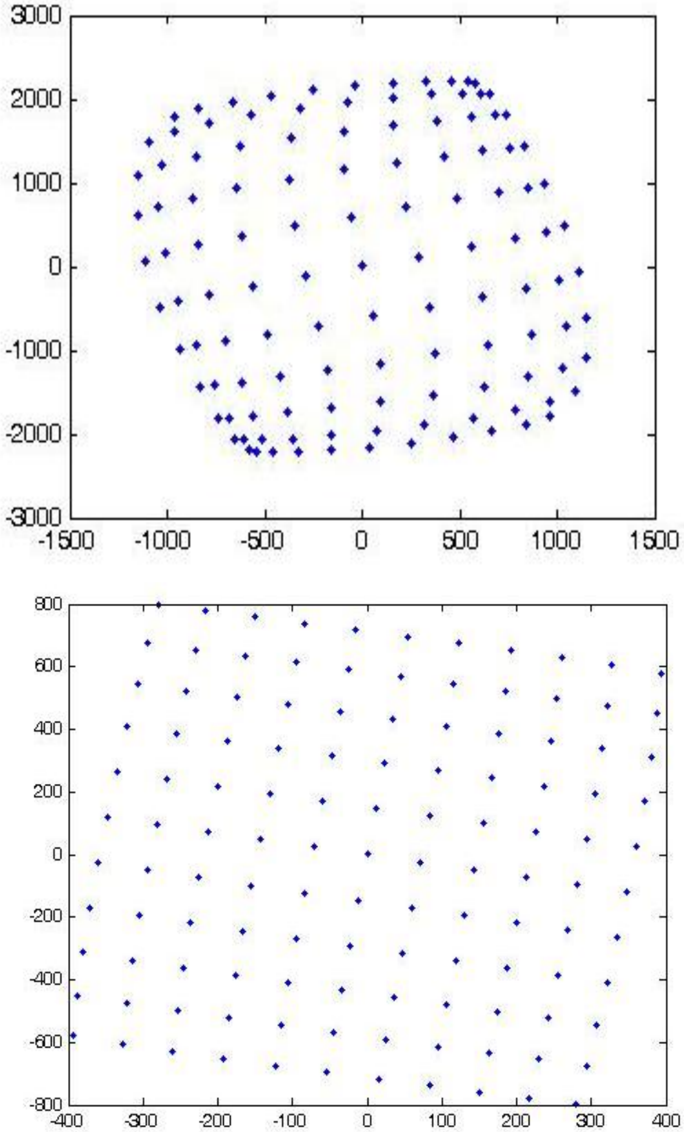


Figure 3.10, The effect of image blurring on flattening a translation manifold for sign ‘A’ as shown in Fig. 3.9.

It has also the effect of grouping objects together as the feature space of the data starts to shrink and small changes between objects are removed. Also, reducing the nonlinearity of the manifolds makes the manifolds more separated in the space. This makes the classification process more accurate as it is easier to differentiate between neighbouring manifolds. Figure 3.11 shows two neighbouring manifolds before and after blurring (using the 1st and 2nd eigenvectors).

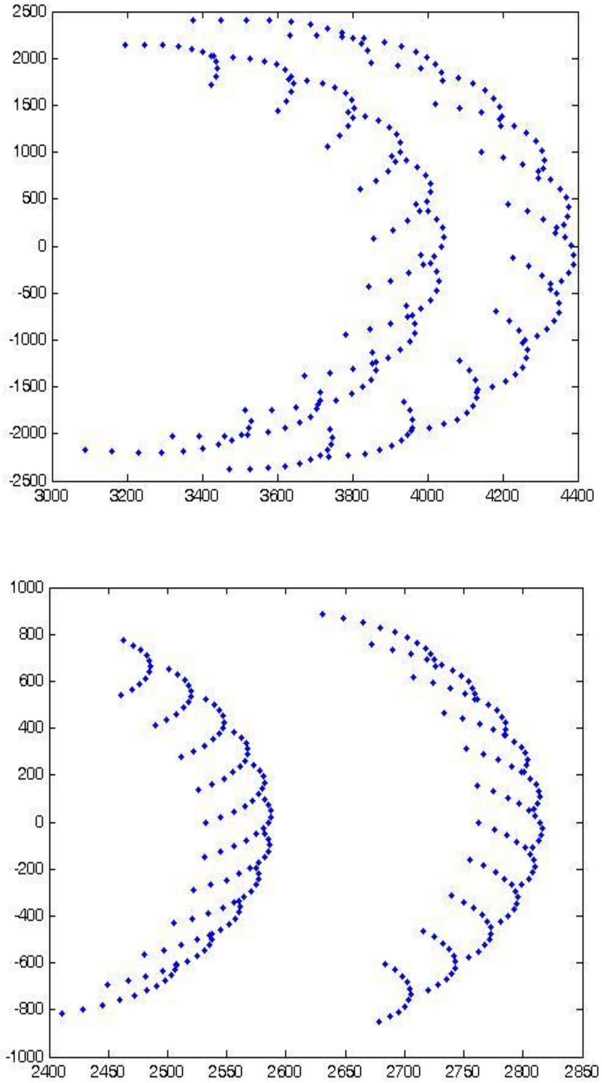


Figure 3.11, The effect of blurring on separating the manifolds.

Figure 3.12 shows two rotation manifolds before and after blurring using three rotation angles from the whole range of the training images. The data objects are projected into the eigenspace using the second and third eigenvectors. The manifolds are more flat and less nonlinear decision boundaries are needed.

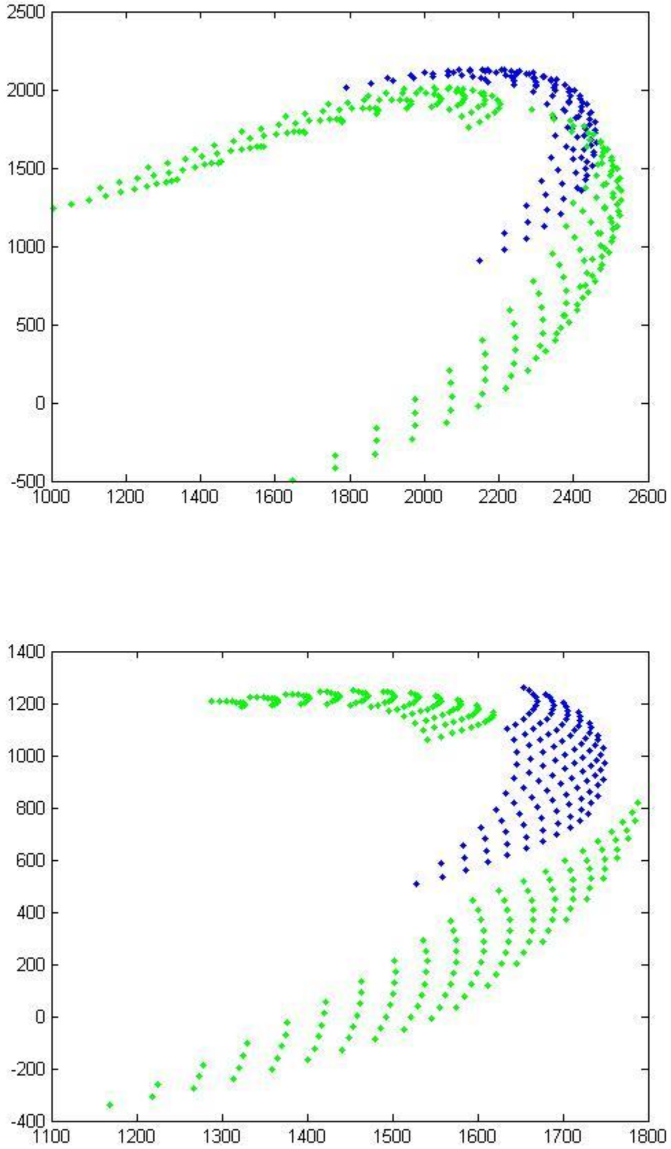


Figure 3.12, The effect of blurring on flattening a rotation manifold.

3.5 Proposed Multistage Hierarchy

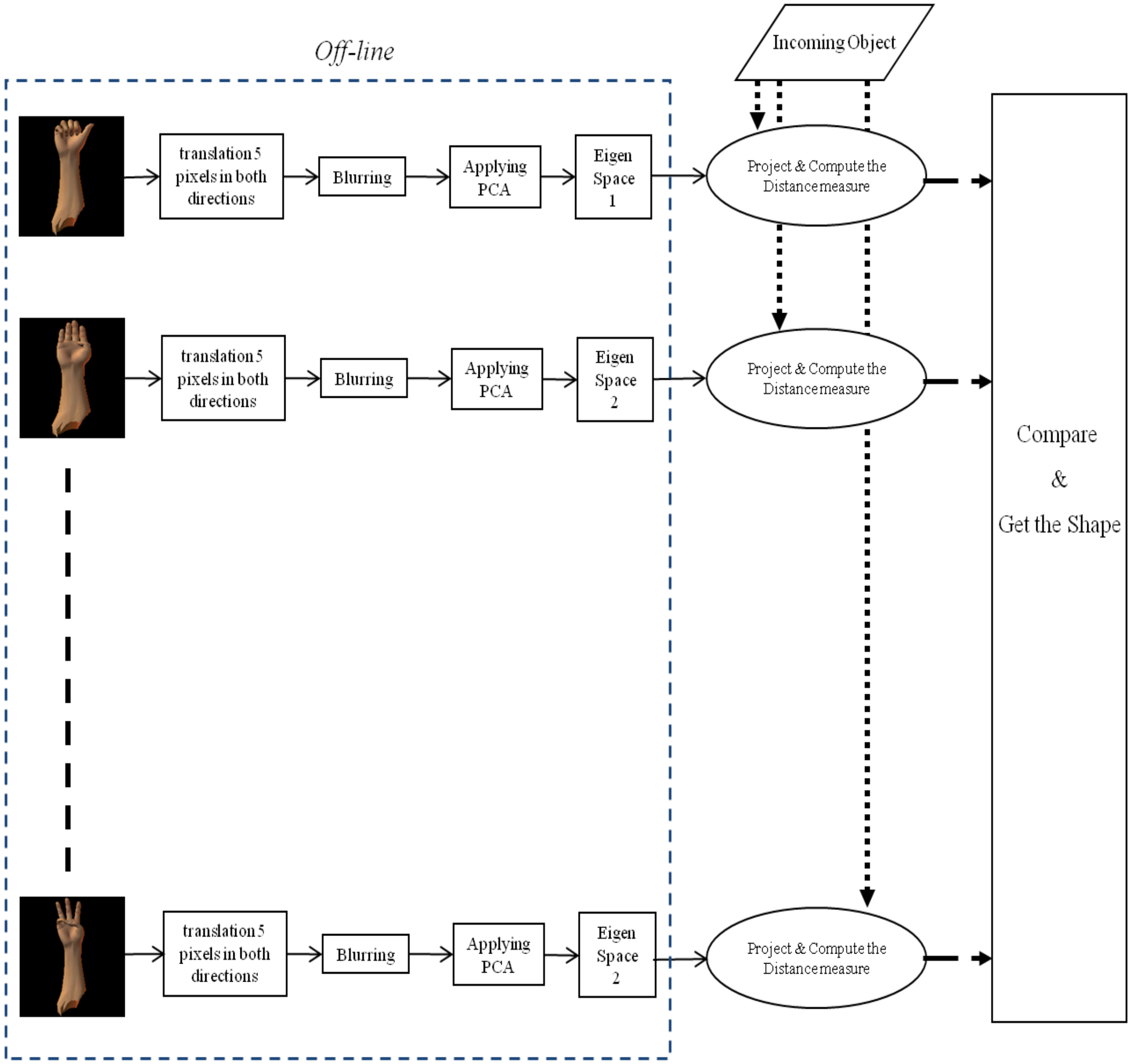

Off-line
Incoming Object
translation 5 pixels in both directions
Blurring
Applying PCA
Eigen Space 1
Project & Compute the Distance measure
translation 5 pixels in both directions
Blurring
Applying PCA
Eigen Space 2
Project & Compute the Distance measure
translation 5 pixels in both directions
Blurring
Applying PCA
Eigen Space 2
Project & Compute the Distance measure
Compare
&
Get the Shape

Figure 3.13, The frame-work for a PCA classifier.

A large number of translation manifolds are generated for the different hand shapes at different rotations of the signer arm, in order to be invariant to these linear transformations as shown in Figure 3.13. It is inefficient to apply an exhaustive search to search linearly through these manifolds in order to classify a new incoming object, when each manifold has its own set of eigenvectors that build its own eigenspace.

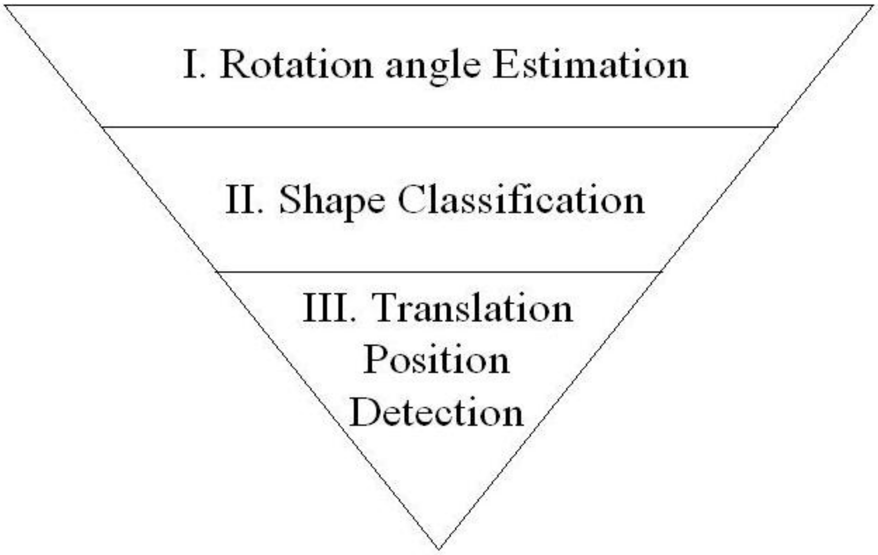


Figure 3.14, The proposed multistage hierarchy.

A multistage hierarchical structure is used in the proposed algorithms to reduce the search space from one stage to another to find the closest manifold to search in and hence decide the shape, rotation, and translation position of an incoming sign object to the nearest neighbour. At the top level of the search pyramid, estimation of the rotation angle for the signer arm in a new pattern is done using rotation manifolds. Estimation of rotation is done first as it has the largest variation within the data. As blurring removes small changes between objects, different shapes start to be more similar and the dominant effect will be for the rotation of the signer arm using the suitable blurring level. So it is possible at the first stage to estimate a range of rotation angles for the new incoming object in an easy way. The second stage of the pyramid is classifying the hand shape for new patterns, as the hand shape has the second largest variation within the data. At this stage, shape manifolds are used. The incoming object is projected into the shape manifolds that only cover the range of angles estimated in the first stage. A final third stage is done to estimate the translation position using a nearest neighbour search. The incoming object is projected into the translation manifolds that represent the classified shape at the estimated angle of rotation from the previous two stages. Figure 3.14 shows the different stages in the proposed multistage hierarchy.

Originally, PCA is an unsupervised learning technique that builds the eigenspaces based on the feature space of the original data without taking into consideration the different classes which represent the different poses and shapes of the hands. In the next sections four algorithms are proposed to show how manifolds can be organized in different ways to help in building the proposed hierarchical strategy. The first algorithm is based on using perpendicular distance to measure the distance between the incoming object and the different manifolds to classify it by the nearest neighbour. The second algorithm is based on using supervised Multidimensional Grids to divide the eigenspace into cells of similar objects according to the rotation or shapes it has. An incoming object can be classified according to the information in the cells it is projected into. The third algorithm is based on using unsupervised Multidimensional Grids to build the hierarchy. The fourth algorithm is based on training simple-architecture multilayer neural networks at the different levels of the pyramid to map the incoming object to its nearest shape and orientation through the hierarchy.

3.6 A Multistage Hierarchical Algorithm for Hand Shape Recognition using Perpendicular Distance

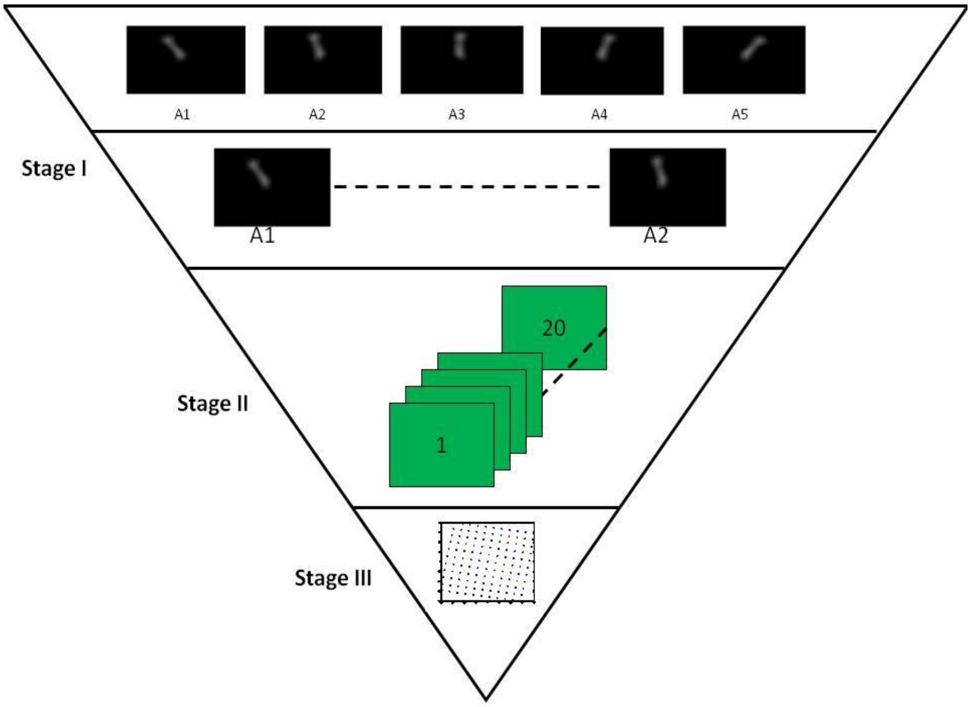


Figure 3.15, Multi-stage hierarchy using Perpendicular distance.

The proposed algorithm is based on using the perpendicular distance to measure the distance between the incoming object and a set of translation manifolds. These translation manifolds are for blurred images that represent different hand shapes at different rotations of the signer arm. The same blurring level is used through the hierarchy. The algorithm follows the same hierarchy that has been discussed in Section 3.5. Figure 3.15 shows a summary for the first proposed multistage hierarchical structure.

3.6.1 Stage 1: Rotation Angle Estimation

At Stage 1a we make an initial rough estimate of rotation angle for the incoming object, by constructing a rotation manifold for one of the 20 shapes. Sign “H” is chosen as it is the most centrally located within the range of shapes. Translation manifolds for “H” are constructed at a set of rotation angles at intervals of 10 degrees throughout the whole range. Perpendicular distance is used to compute the distance between the incoming object and each manifold to find the first and the second nearest manifolds. These two manifolds represent the start and the end of an interval for a smaller range of angles to search in at Stage 1b to find the closest estimated angle from the training data. As blurring removes only small changes between objects, a single selected shape is not enough to compare with in Stage 1b. Therefore, six objects from the 20 shapes are chosen to compare with. The value six was chosen experimentally, as will be discussed in the experimental results in Chapter 4. To pick the six objects, a k-medoid clustering algorithm [69] can be used with k = 6. Each medoid object represents the centre of a cluster. Figure 3.16 shows a pseudo-code for Stage 1 in the first proposed algorithm.

3.6.2 Stage 2: Shape Classification

After the angle of rotation has been estimated in Stage 1, a search through the translation manifolds for the 20 shapes at the estimated angle is needed. The perpendicular distances between the incoming object and the different manifolds for the 20 shapes are computed. The shortest distance leads to the nearest neighbor shape. Figure 3.17 shows a pseudo code for stage 2 and 3 in the first proposed algorithm.

3.6.3 Stage 3: Translation Position Estimation

At Stage 1 and 2, the rotation angle and the shape for the incoming object are classified. At Stage 3, a search through the 121 objects that represent the translation manifold for the classified shape at the estimated angle of rotation is achieved. The nearest neighbour object from the 121 objects leads to the translation position of the new pattern. The incoming object is projected into that eigenspace where the Manhattan distance is used to compute the distance measure between it and the translated objects within that manifold. Manhattan distance is used as it is one of the fastest distance measures to compute.

```
Classification Phase
Stage 1_a

P = new incoming image
R = set of rotation angles for ‘H’ in increments of 10 degrees
for each $r \in R$ do
        z = set of 121 translated objects at rotation r
        v = set of eigenvectors(‘H’,r) constituting the translation-manifold of ‘H’ at r
        find D(r) the perpendicular distance of P from v
end
find R1 the member of R with the lowest value of D
find R2 the member of R with the second lowest value of D

Stage 1_b

S = set of k shapes given by the k-medoid algorithm
R = set of rotation angles from R1 to midpoint(R1,R2) in increments of 2 degrees
for each $s \in S$ do
        for each $r \in R$ do
             z = set of 121 translated objects for shape s at rotation angle r
             v = set of eigenvectors(s,r) constituting the translation-manifold for s at r
             find D(s,r) the perpendicular distance of P from v
         end
end
find the values of s and r which minimize D
Estimated Rotation Angle = r
```

Figure 3.16, Pseudo-code for Stage 1 in the 1st proposed classifier.

```
Stage 2

P = new incoming image
R = Estimated Rotation Angle form Stage 1_b

S = set of 20 shapes at rotation angle R
for each s∈S do
        z = set of 121 translated objects for shape s
        v = set of eigenvectors(s,R) constituting the translation-manifold of s at R
        find D(s) the perpendicular distance of P from v
end
find the value of s which minimizes D, this is the classified shape of P

Stage 3

S = classified shape
z = set of 121 translated objects for shape S at rotation angle R
v = set of eigenvectors(S,R) constituting the translation-manifold of S at R
find zp, the projection of z onto v
find Pp, the projection of P onto v
for i=1:121 do
      find D(i) the Manhattan distance from zp(i) to Pp
end
find the value of i which minimizes D. this is the estimated translation of P
```

Figure 3.17, Pseudo-code for Stage 2 and 3 in the 1st proposed classifier.

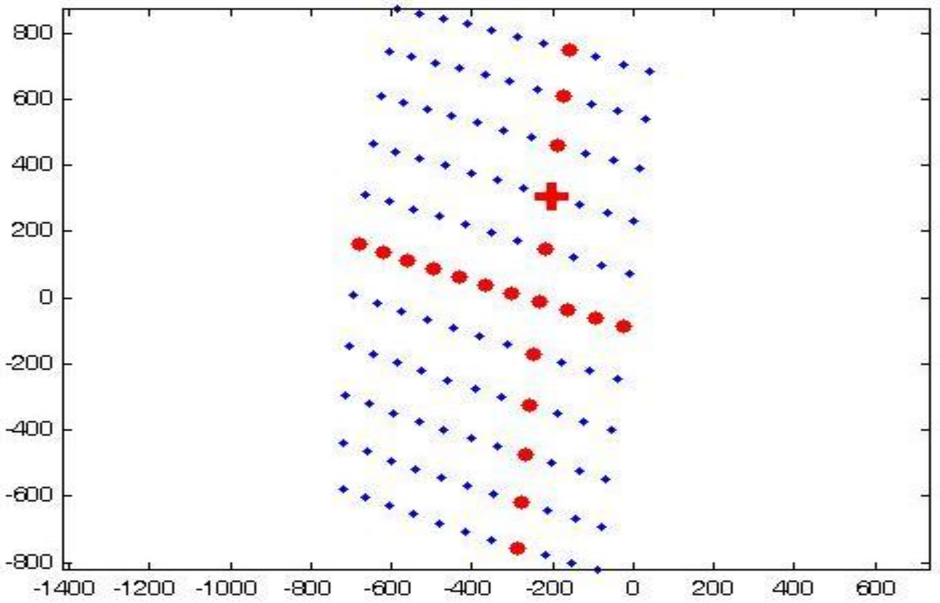


Figure 3.18, Variable separation at the third stage.

A variable separation technique can be applied, instead of searching linearly through the 121 objects as shown in Figure 3.18. The object with the cross sign represents an incoming object with translation 2 pixels to the right and 2 pixels up with respect to the origin, which is the central object. The variable separation is done first by searching through the 11 objects, which represent the horizontal translations with zero vertical translation. The nearest neighbour leads to the horizontal translation of the incoming object, where the vertical translation is detected by searching through the 11 objects that represent the different vertical translations at the classified horizontal translation as shown in the figure. The variable separation reduces the search from 121 objects to only 22 objects.

3.6.4 Fine Search

When moving in a top-down direction or when moving from a higher to a lower level through the multistage hierarchical algorithm, the scope of the search narrows. Therefore, if the higher level decision is wrong there is no way to backtrack. Therefore, a fine search strategy is needed to increase the accuracy of the proposed algorithm. Instead of taking a hard decision about the angle of rotation, Stage 2 is applied for the k nearest estimated angles for the incoming object. The perpendicular distances between the incoming object and the 20*k manifolds for the 20 shapes of the k estimated angles are sorted. The manifold with shortest distance represents the top match for the shape of the incoming object. A decision about the translation position can be then taken at Stage 3. A discussion about the best choice for the parameter k is presented later in the experimental results section.

3.6.5 Variations on the Algorithm using Sub-manifolds

Some variations on the proposed algorithm can offer improvements in its speed and accuracy. In the algorithm which has been presented, a separate manifold for each hand-shape at each angle of rotation is generated offline. The number of eigenvectors used for the classification process is important as it can affect both efficiency and accuracy. The length of each eigenvector is also important as it has the same effect. In each separate manifold, the length of each eigenvector is actually the number of pixels in the original image. In order to reduce the length of the eigenvectors, the concepts of rotation sub-manifolds and shape sub-manifolds are applied.

At Stage 1 of the algorithm, which has been just presented, a set of separate translation manifolds at different rotations are used to estimate the angle of rotation for an incoming image. Instead of this, rotation sub-manifolds can be used in Stage 1. They are more computationally efficient, as the length of their eigenvectors is much shorter than the length of the eigenvectors in the separate translation manifolds. The length of the sub-manifold eigenvectors is equal to the number of eigenvectors used in the parent eigenspace of the rotation manifold. See Figure 3.5.

In Stage 2 of the algorithm, which has been presented, 20 eigenspaces are used representing the 20 shapes at the estimated angle of rotation to classify the shape of a new incoming image. Shape sub-manifolds could be used to enhance the performance in Stage 2 as the length of the eigenvectors for the shape sub-manifolds is shorter. It is equal to the number of eigenvectors used for the parent eigenspace of the shape manifold.

3.7 A Multistage Hierarchical Algorithm for Hand Shape Recognition using Supervised Multidimensional Grids

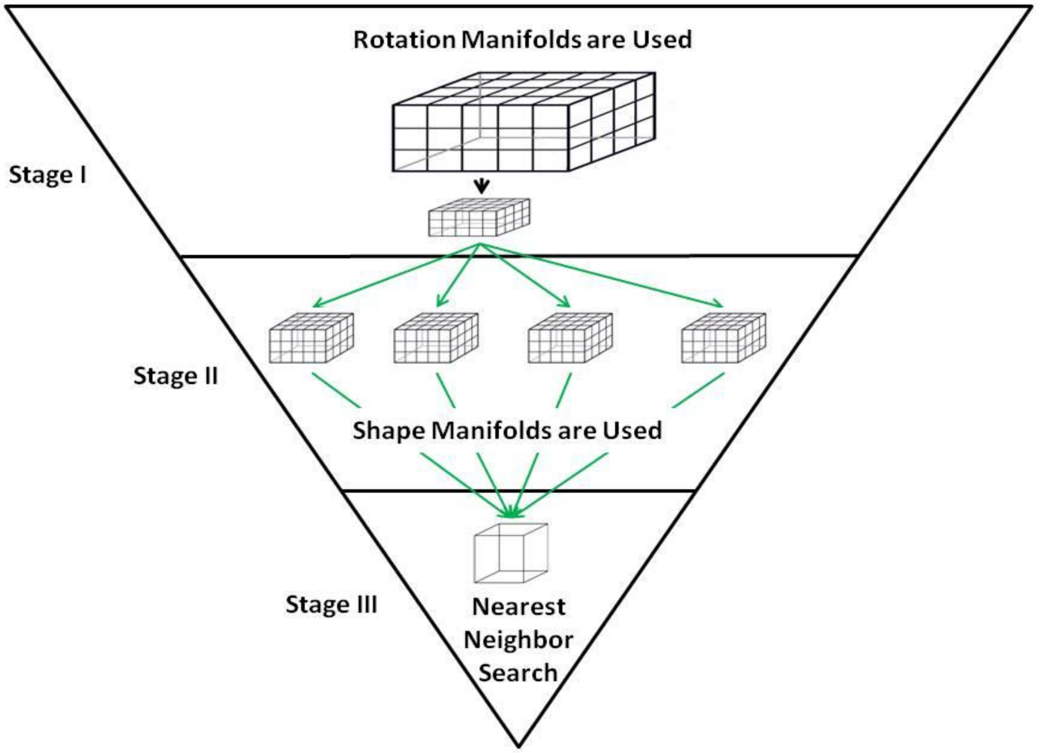


Figure 3.19, Multi-stage hierarchy using supervised MDGs.

A Multidimensional Grid (MDG) is a structure that is used to store multidimensional data within a grid of cells, which are hyper-cubes. The second proposed algorithm extends the ideas of rotation manifolds and shape manifolds by using Multidimensional Grids that divide these different spaces into cells of hyper-cubes. The objects within each cell can give enough information to classify a new incoming object in an accurate and efficient way, as determined during the training phase by varying the grid size. This will be discussed in Chapter 4 in the section on experimental results.

The Multidimensional Grid structure can be built based on the set of eigenvectors that have the highest eigenvalues. The grid divides each direction of the space into equal intervals based on the range of feature values of the objects that is used to build that eigenspace. The eigenspace is divided by the grid into multidimensional cells of hyper-cubes. Each cell can be labeled according to the information from the objects it holds. The algorithm follows a hierarchical strategy to classify the new patterns in an efficient and accurate way, as described in Figure 3.16. The new incoming object is projected into the eigenspace for the associated MDG at each level of the pyramid to analyze it. A study of the effect of blurring and the grid size on both accuracy and performance is given in the next chapter.

3.7.1 Stage 1: Rotation Angle Estimation

Stage 1.A: The first stage of the algorithm is to estimate a range of rotation angles for an incoming object. Rotation manifolds are used at the first stage to build the grid structures at this level. Each cell in the MDG is labelled according to the minimum and maximum rotation angles for the objects it holds. In order to classify a new incoming object it is blurred to the same level as the images of the MDG and projected into it. According to the label of the cell it is projected into, estimation for the range of angles for that object is done.

A rotation manifold for sign “H” as the most centrally located object is used to compute the eigenvectors and cells of the MDG. Every fifth image from the training set is used. To densely fill all the cells of the MDG, other objects from different shapes at the same rotations are projected into the space to fill in some of the empty cells in that grid.

This improves the accuracy of the proposed algorithm as projecting an object into an empty cell will lead to incorrectly estimating its rotation angle and hence may cause a misclassification for the shape in the next stage of the proposed algorithm. The size of the MDG, in terms of the number of eigenvectors and the number of cells in each direction, has an effect on the number of objects within the cells and hence the range of angles within each cell. The blurring level also has an effect on the distribution of objects within the grid as it makes the manifolds more flat and reduces the distance between objects. A study of these factors and the effect on both accuracy and performance is given in the experimental results section in the next chapter.

Stage 1.B: A sub-stage is applied by projecting the incoming object into a smaller MDG. The goal of this sub-stage is to increase the precision by estimating a narrower range of angles while preserving the accuracy. Smaller MDGs are constructed using the 6 angles from the training set representing the range of angles intermediate between the angles used for the bigger grid. The new incoming object is projected again into one of these smaller MDGs based on the range of angles that has been estimated from the first big MDG. The range of angles estimated from this small MDG is passed to Stage 2.

Figure 3.20 shows a rotation manifold for objects that represent the signer arm from the horizontal to the vertical position and the MDG construction for it is of size [2x2] using the first 2 highest eigenvectors. The figure shows four different coloured regions that represents the four cells of the MDG.

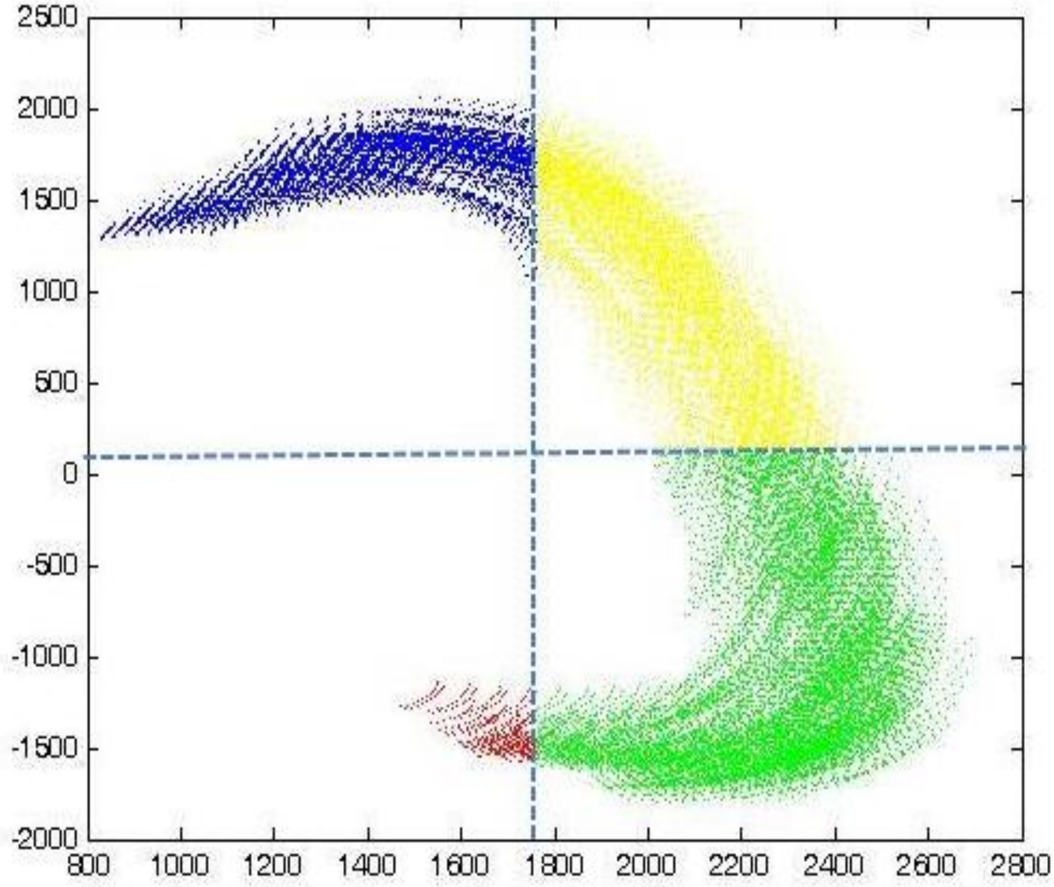


Figure 3.20, MDG [2x2] for a rotation manifold at the 1st stage in the 2nd algorithm.

3.7.2 Stage 2: Shape Classification

In the second stage of the proposed algorithm, MDGs are constructed using shape manifolds. For each consecutive pair of angles, an eigenspace is constructed, which contains all shapes and translation at those two angles. Then a MDG is constructed around these data. According to the range of angles that has been estimated by Stage (1), a new incoming object will be projected into a number of MDGs in Stage (2) that cover that range. The smaller the number of angles from Stage (1) to be searched in, the fewer MDGs will be searched and the more speed is gained.

3.7.3 Stage 3: Final classification

According to the cell, which the incoming object is projected into in Stage (2), a nearest neighbour search is performed in Stage 3 for the objects within that cell. If more than one MDG is used to cover the estimated range of angles from Stage (1), the nearest neighbour object among the all MDGs from Stage (2) will be used to classify the new incoming object to give a final decision about its shape, rotation, and translation. Manhattan distance is used as a distance measure in this stage, as it is one of the fastest distance measures. The number of eigenvectors used to compute the distance measure will affect the accuracy of the algorithm as will be discussed later in the experimental results in the next chapter. Figure 3.21 shows a pseudo-code for the second proposed algorithm.

3.7.4 Handling Outliers and Empty Cells

Some objects can be misclassified, if they are projected out of the range of the features for a MDG or projected into one of the empty cells. An enhancement to the accuracy of the classification process for the proposed algorithm can be achieved by labeling outliers and patterns which are projected into empty cells. Instead of labeling them with an arbitrary class value, they are labeled by the objects in the nearest nonempty cell. This is achieved by searching through the dimension that represents the eigenvector with the lowest eigenvalue that is used to build that eigenspace as it has the lowest effect on the data.

```
Classification Phase

Stage 1_a

P = new incoming image
M = the MDG for the rotation manifold at increment of 10 degrees
v = eigenvectors(M)
p = projection of P onto v
find cell in M that holds p
R1 = StartAngle of cell
R2 = EndAngle of cell

Stage 1_b

M = the MDG for a rotation manifold at increments of 2 degrees covering R1:R2
v = eigenvectors(M)
p = projection of P onto v
find cell in M that holds p
R1 = StartAngle of cell
R2 = EndAngle of cell

Stage 2

for r = R1:R2 in increments of 2 degrees
    M = The MDG for a shape manifold covering r and r+1
    v = eigenvectors(M)
    p = projection of P onto v
    final cell in M that holds p
    L = list of points within cell
end

Stage 3

Compute Manhattan distance of p from each point in L
Find the point with minimum distance from p
Classify P with the rotation angle, shape and translation position of that point
```

Figure 3.21, Pseudo-code for the 2nd proposed classifier.

3.8 A Multistage Hierarchical Algorithm for Hand Shape Recognition using Unsupervised Multidimensional Grids

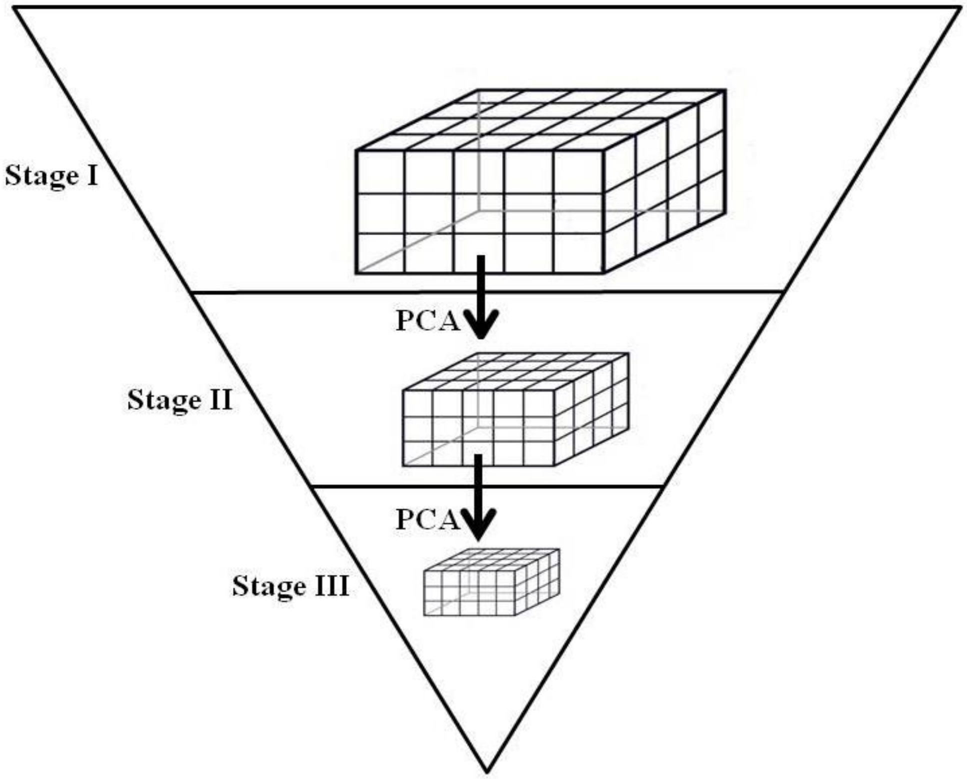


Figure 3.22, Multi-stage hierarchy using un-supervised MDGs.

The third proposed algorithm represents an unsupervised multistage hierarchy using Multidimensional Grids. The data is clustered into cells of similar objects according to the feature space computed by applying PCA on the blurred training images. The top level of the hierarchy uses a rotation manifold for sign 'H' as the most centrally located shape. That eigenspace is used to project the whole data set where a Multidimensional Grid can divide the objects into cells that hold different shapes and orientations according to the similarity between objects. Each cell can build a new grid in the next level if it has number of objects exceeding a threshold. The value of the threshold is determined experimentally as will be discussed in Chapter 4. At a new level, a new PCA space is computed for the group of objects inherited from a cell in the parent grid. PCA is computed for these images in the original space where this new PCA space should reflect the distribution of these particular objects in more detail. The process can continue to build new levels of grids until the current cell is empty or holds a number of objects lower than a predefined threshold. These cells then form the leaf nodes in the hierarchical tree. Figure 3.22 shows the proposed hierarchy.

For a new pattern, it is projected into the top grid and, according to the cell it is held by, it can be projected again to a grid in the next level. The process continues from one level to another. The new pattern can be classified by finding the nearest neighbour object in the cell it is projected into at the bottom level of that hierarchy. Manhattan distance is used to compute the distance measure between the new pattern and the objects in that cell. A sufficient number of eigenvectors is used in the distance measure in order to get the highest accuracy. The number of eigenvectors is determined during the training phase by finding the number, which maximizes the accuracy measure.

A backtracking step can be applied for the classification process, if the new pattern is projected into an empty cell or a cell with a number of objects lower than a backtracking threshold. In that case this object is re-projected into the corresponding parent grid cell to find the nearest neighbour object to be classified with. Figure 3.23 shows the training set projected into the top-level eigenspace. Figure 3.24 shows objects for one of the cells in Figure 3.23 projected into the PCA space at the second level. Figure 3.25 shows the objects for one of the cells in Figure 3.24 projected into the PCA space at the third level. The three figures illustrate how the data becomes sparser and more linear at lower levels. The green points in the figures fall into the eigenspace in the next figure. Figure 3.26 shows pseudo-code for the third proposed algorithm.

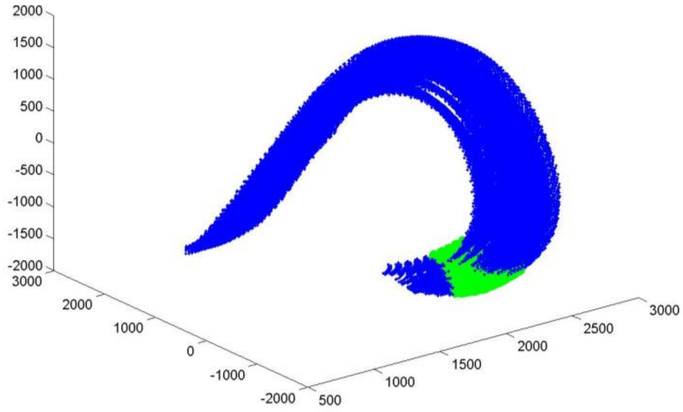


Figure 3.23, The top level eigenspace.

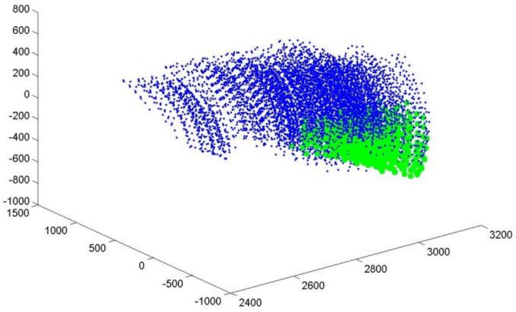


Figure 3.24, An eigenspace for a cell at the second level.

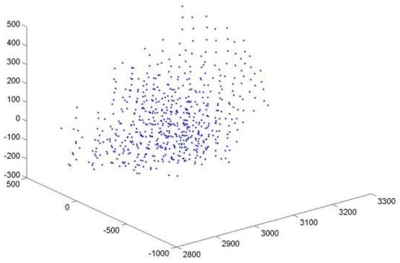


Figure 3.25, An eigenspace for a cell at the third level.

```
Classification Phase
P = new incoming image
Cell_Index = 1
Level = 0
Seq = [ ]
do
    Seq = [Seq;Cell_Index]
    Level = Level+1
    M = MDG(Seq)
    v = eigenvectors(M)
    p = projection of P onto v
    find cell in M that holds p
    Cell_Index = the index of that cell
while (size(M(Cell_Index)) > MinObjs));

while (size(M(Cell_Index)) < Backtrack) && (Level > 1)
        Level = Level-1
        Seq = Seq(1:Level)
end
L = list of points within the final cell
Compute Manhattan distance of p from each point in L
Find the point with minimum distance from p
Classify P with the rotation angle, shape and translation position of that point
```

Figure 3.26, Pseudo-code for the 3rd proposed classifier.

3.8.1 An Extension to Real Hands

The third proposed algorithm is capable of handling real images as it is an unsupervised classifier. For a real signer it is difficult to show the signs at specified pose parameters. Figure 3.27 shows the 20 shapes in the Irish Sign Language using a real hand. Real hand videos were recorded for each sign by performing a gesture by moving the hand shape from the vertical to the horizontal position. The videos were converted into 100 frames for each sign. Each image was translated 5 pixels in each direction forming 121 translations. The odd numbered frames were gathered for training forming 96,800 objects as an example-based dataset. The even numbered frames were used for testing representing the intermediate rotation angles as will be discussed in the experimental results chapter.

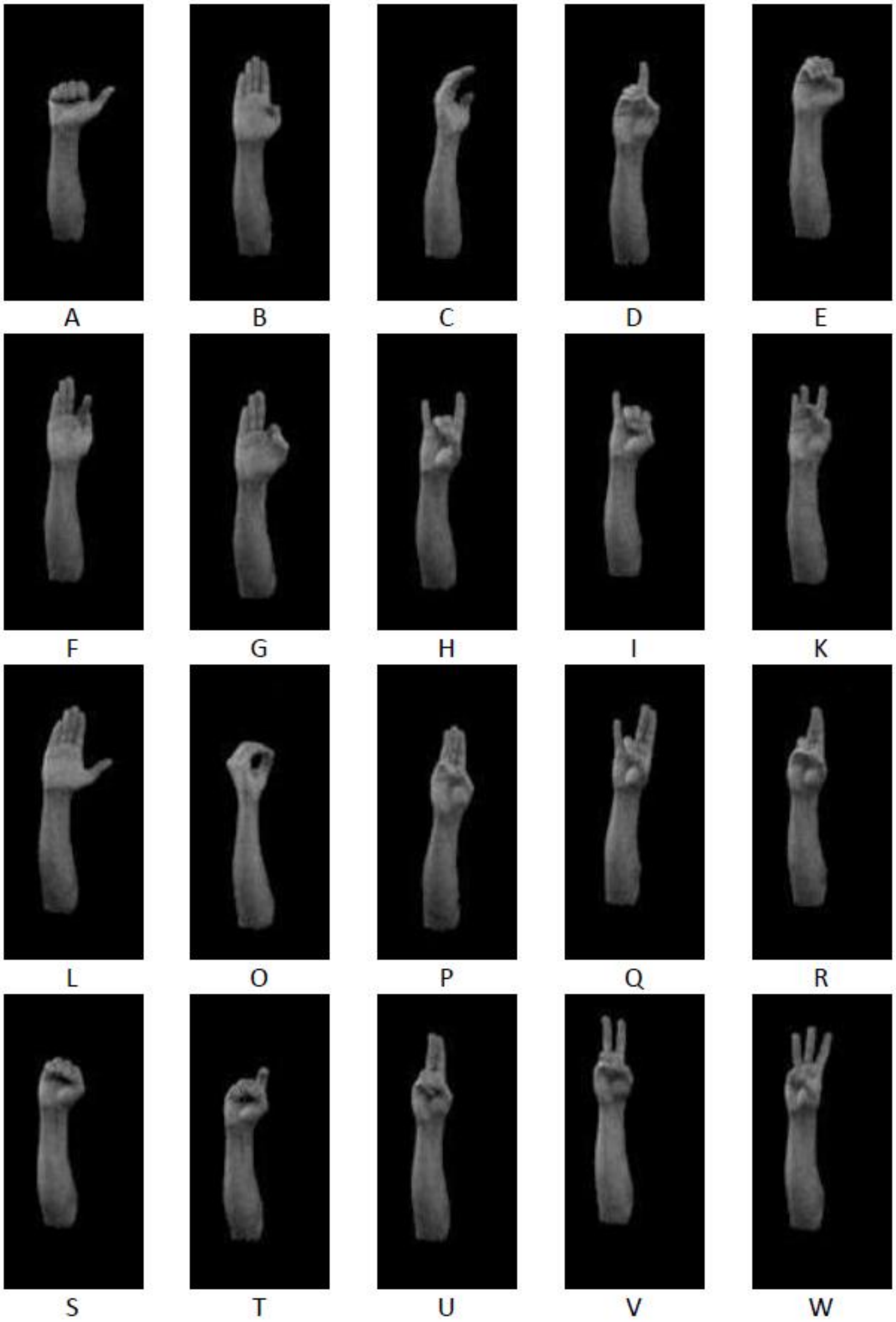


Figure 3.27, Real hands for 20 shapes in the Irish Sign Language.

3.9 A Multistage Hierarchical Algorithm for Hand Shape Recognition using Artificial Neural Networks

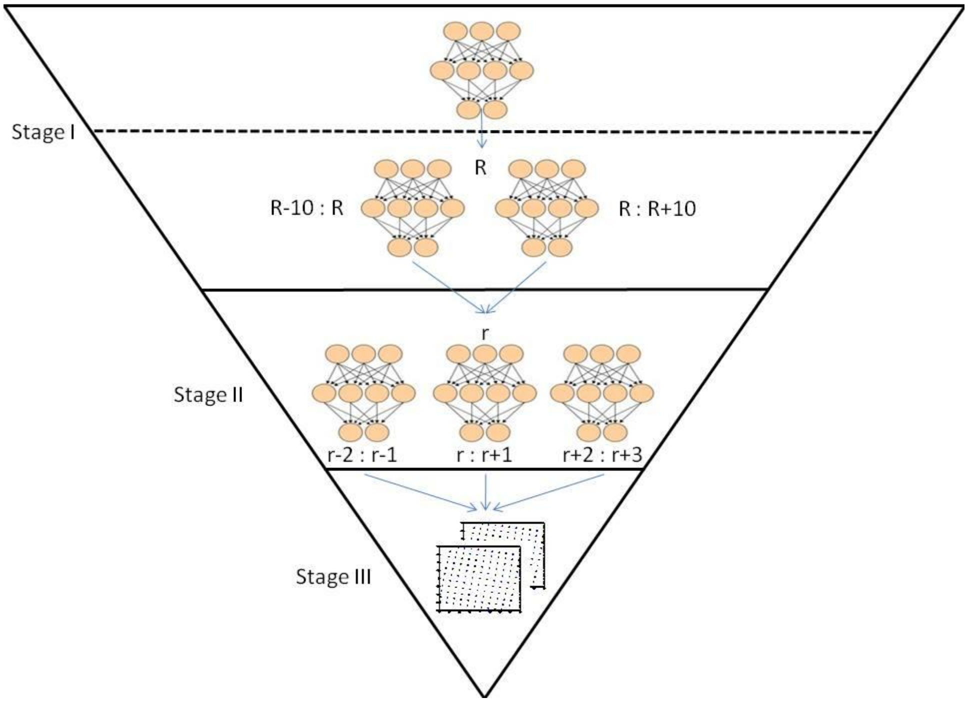


Figure 3.28, Multistage hierarchy using ANNs.

The fourth proposed algorithm presents a data pyramid using a set of neural networks at the different levels. The data pyramid organizes the different eigenspaces in a supervised manner according to different pose and shape parameters. It follows a three stage pyramid as shown in Figure 3.28. At the first stage ANNs are used to map a new pattern to a range of in-plane rotation angles as estimation for the rotation of the signer arm. At the second stage, ANNs are trained to classify the shape of the hand in the incoming pattern. At the third stage, the translation position is estimated based on a nearest neighbour search for the objects within the manifolds that represent the classified shape at the estimated rotation.

It is difficult to design or train a single neural network based on the eigenspace for the whole training set. Therefore we subdivide the global task by creating different neural networks for different sub-spaces. The algorithm follows a pyramid strategy to design and train simple neural networks architectures. At each level of the pyramid, different neural networks are trained on the PCA subspaces that represent a certain pose or shape from the training set. The output from the neural networks at each level gives an estimate for a different parameter for the classification process. Neural networks are used to map new patterns using the nearest class parameter. Therefore, a narrower range of eigenspaces can be used for the classification process throughout the hierarchy.

The proposed pyramid is based on training multilayer neural networks with two hidden layers using a Feed-forward Back-propagation training algorithm. The hidden layers are trained with a sigmoid function and the output layers are trained with a linear function. The sizes of the input and hidden layers in terms of the number of neurons are chosen based on the number of eigenvectors used in the input space. The number of eigenvectors is chosen during the training stage in order to acheive the highest possible accuracy, as will be discussed later in the experimental results chapter.

As image blurring has the effect of reducing the nonlinearity in the training manifolds, nonlinearity in the decision boundaries is reduced. Therefore, simpler neural network architectures can be designed to fulfill the requirements for the classification process. An improvement in the speed of training the neural networks can be achieved as it is easier to find the decision boundaries as smaller numbers of neurons are used in the design of the neural network architecture.

3.9.1 Stage 1: Rotation Angle Estimation

Neural networks are trained with rotation manifolds at the first stage in order to estimate the rotation angle for new incoming patterns. Images for sign ‘H’ are used to construct the rotation manifolds at this stage. Shape ‘H’ is considered as the most centrally located shape where image blurring is used to remove small changes between objects.

In Stage 1.A, a four-layer neural network is trained with a rotation manifold covering the range from the horizontal to the vertical position for the signer arm at intervals of 10 degrees. The neural network at this sub-stage, maps a new pattern to the closest discrete angle of rotation from the training examples in that range of angles. The output layer for the neural network represents a binary valued code for one of these discrete angles. Each neuron in the output layer represents 1 bit in the binary valued code.

In Stage 1.B, estimation for a narrower range of rotation angles is achieved. A search window of 10 degrees before and after the estimated angle from Stage 1.A is applied. Neural networks are constructed for each range of consecutive angles from the intermediate range in Stage 1.A and are trained with rotation manifolds for these narrower ranges. The output layer for the neural networks at this stage represents a binary code for the rotation angle sequence within the training range. For a new pattern, it is mapped using a pair of networks that represent the search window defined by the estimated angle from Stage 1.A. The output from this stage is considered as the output angle from the network that has the minimum mean square error between the desired and the actual output.

3.9.2 Stage 2: Shape Classification

At the second stage, shape manifolds containing all 20 shapes are constructed at each pair of consecutive angles. A neural network is trained with each shape manifold where the output of the network represents a binary valued code for each of the twenty shapes. The shape for a new pattern can be classified after projecting it into the corresponding set of eigenvectors. The new pattern is mapped by the neural network that covers the estimated angle from Stage 1.B. To fine tune the results, a search window can be applied considering two more neural networks covering 2 more ranges on either side of the estimated angle. The output from this stage is considered as the classified shape from the neural network that has the minimum mean square error between the desired and actual output.

3.9.3 Stage 3: Translation Position Estimation

At the final stage, estimation for the translation position for a new pattern is achieved. Labeling is done according to the closest object using a nearest neighbour search to a pair of translation manifolds at the estimated pose and classified shape. The pair of manifolds covers the range of angles for the winning neural network from the second stage. The new pattern is projected into each eigenspace. A distance measure between it and the objects within each translation manifold is computed based on the Manhattan distance measure as one of the fastest distance measures. A sufficient number of eigenvectors is used to compute the distance measure in order to achieve the highest accuracy by varying the accepted number of eigenvectors during the training stage as will be discussed in the experimental results chapter. Figure 3.29 shows pseudo-code for the fourth algorithm.

```
Classification Phase

Stage 1_a
P = new incoming image
A = The ANN for the rotation manifold in increments of 10 degrees
v = eigenvectors(A)
project P onto v to obtain point p
input p into A and obtain estimate for the rotation angle R
R1 = R-5
R2 = R+5

Stage 1_b
for i = R1 : R2 in steps of 5
    A = The ANN for the rotation manifold in increments of 2 degrees covering i:i+5
    v = eigenvectors(A)
    project P onto v to obtain point p
    input p into A and obtain estimate for the rotation angle R and error E
end
find the value of R which minimizes E
R1 = R-2
R2 = R+2

Stage 2
for i = R1 : R2 in steps of 2
    A = The ANN for the shape manifold covering i and i+1
    v = eigenvectors(A)
    project P onto v to obtain point p
    input p into A and obtain estimate for the shape S and error E
end
find the value of angle i and shape S which minimizes E
R=i

Stage 3
for i =R:R+1 do
    z = set of 121 translated objects for shape S at rotation angle i
    v = set of eigenvectors(S,i)
    project z onto v to obtain points zp
    project P onto v to obtain point p
    for each point zp, find the Manhattan distance from p
end
find the point which minimizes the Manhattan distance from p
Label P with rotation angle, shape and translation of that point
```

Figure 3.29, Pseudo-code for the 4th proposed classifier.

3.10 Conclusion

The proposed algorithms are presented as an example-based hand shape classifiers. A large training set of 111320 images is used to cover the variability of 20 hand shapes from the Irish Sign Language covering the possible translations and rotations. The proposed algorithms are invariant to these rigid transformations. PCA is used for feature extraction and dimensionality reduction. In comparison with nonlinear manifold learning techniques, the proposed pyramids can help in applying linear PCA to nonlinear data. Although PCA doesn't work well with nonlinear data, the proposed pyramids provide a solution by applying PCA to subspaces which are approximately linear. A hierarchical pyramid strategy is used to accelerate the search process by approximating the k-nearest neighbor algorithm by analyzing the new patterns using different PCA spaces through the pyramid hierarchy. Estimation of the rotation angle of the signer arm is achieved at the first level of the hierarchy using rotation manifolds, as rotation has the highest variation within the data. Then shape manifolds are used for classifying the shape at the second stage. A final decision can be made at the bottom end level of the pyramid to label the incoming pattern with the nearest neighbour object from the training set. Image blurring is used to remove small changes between objects to increase the generalization factor. The proposed algorithms explore the effect of image blurring on flattening the manifolds. Image blurring has the effect of reducing the nonlinearity in manifolds and making data more separable as discussed in Section 3.4. Four pattern recognition techniques are used to implement the proposed hierarchy.

- The first algorithm is based on finding the nearest neighbour manifold in each level using the perpendicular distance measure. linearising the manifold helps in measuring the distance more accurately where it assumes that the manifolds are ideally flat. Sub-manifolds are used to get more dimensionality reduction by applying PCA a second time to rotation and shape manifolds.
- The second algorithm is based on organizing the eigenspaces using Multidimensional Grids in a supervised manner. MDGs are used to divide the eigenspaces into cells of hyper-cubes, which are labeled by the range of rotations at the first level and shapes at the second level. Using a suitable level of blurring, manifolds are separable enough to be clustered using MDGs.
- The third algorithm is based on building a hierarchy of MDGs in an unsupervised manner. PCA is applied to each nonempty cell from a parent space to build a new level in the pyramid. New patterns are classified based on a nearest neighbour search among the objects of the cell it is projected into at the bottom end level.
- The fourth algorithm is based on training a set of ANNs by different eigenspaces of the training data at each level. ANNs can map new patterns to the parameters space of the training set. The ANN with the highest response in each level leads to the closest class.

Chapter 4

Experimental Results

4.1 Introduction

In this chapter four sets of experimental results are presented to examine the accuracy and performance of the proposed algorithms under the different aspects discussed in the previous chapter. The proposed algorithms have been implemented in Matlab under the Windows 7 operating system. All the experiments are done on Intel i5 CPU @ 2.3 GHz, 2.00 GB of RAM. All the images are 250x330 pixels and converted into gray-scale before blurring. Blurred images are created using a two-dimensional Gaussian low-pass filter of size [6,6] and with variance equal to 10. Table 4.1 shows the variations on the blurring level through this chapter. The performance target in the experiments is to classify 10 frames per second.

	Size	Variance
B1	[6,6]	10
B2	[12,12]	20
B4	[24,24]	40
B6	[36,36]	60
B8	[48,48]	80
B10	[60,60]	100

Table 4.1, Different blurring levels and their filter size.

The results are based on a data sample of 111320 objects. The sample represents the 20 shapes from +90 degrees to +180 degrees. The training examples are 2 degrees apart and each object is translated 5 pixels in the horizontal and vertical directions. The accuracy of the proposed algorithm is examined on three different test-sets. The first test-set is generated from the original training images by rotating each image by an angle randomly generated from a uniform distribution over the interval [-1,1] filling the intermediate angles between the training images. The second test-set is generated by adding a deformation to each image of the previous set by rotating each hand at the wrist by an angle randomly generated from a uniform distribution over the interval [1,5] degrees. To examine the robustness of the algorithm to noise, the third test-set is generated by adding white Gaussian noise to the second test set. Normally distributed random numbers with a standard deviation of 20 and mean 0 are added to each pixel of the segmented arm of the signer. Figure 4.1(a) shows the superposition of two images for sign 'A' in the horizontal position and after rotating the arm and hand 1 degree anti-clockwise and rotating the hand at the wrist 5 degrees anti-clockwise. Figure 4.1(b) shows the same sign after applying random noise to the skin region.

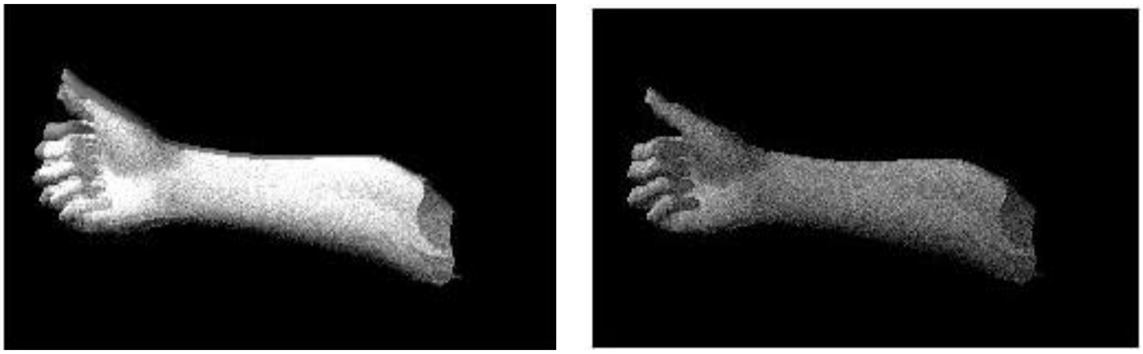


Figure 4.1, The effect of rotation, deformation and noise on a computer generated image.

4.2 First Algorithm Experimental Results

In order to examine the performance and the accuracy, the first algorithm has been tested using different manifold structures. The same blurring level (B6) was used throughout the hierarchy. Signs 'A', 'C', 'H', 'I', 'P', 'R' are selected at the first stage using a k-medoid algorithm. Different values of k were tested and it was found that the highest accuracy for Stage 1 was reached when k = 6.

A study of the selected number of eigenvectors, performance and accuracy is given by varying the number of eigenvectors. The number of eigenvectors can affect the results for the proposed algorithm. It was interesting to find the best number of eigenvectors to use on the test data in order to reach the highest accuracy. Figure 4.2 shows the effect of the number of eigenvectors in Stage 2 on the classification process for the hand shape based on using shape sub-manifolds and rotation sub-manifolds. From the figure, the highest accuracy was reached at 16 eigenvectors. The experimental results show that as the number of eigenvectors increases, the accuracy increases monotonically until saturation. The highest 30 eigenvectors from the parent eigenspace were chosen in order to build the eigenspace for the sub-manifolds as discussed in the previous chapter in Section 3.6.5. The results in the figure are based on using only 4 eigenvectors to estimate the rotation angle at Stage 1. Just 4 eigenvectors are enough at the first stage to let the dominant effect be for the rotation of the arm.

Figure 4.3 shows the accuracy measures versus different blurring levels where the best accuracy measure is gained at blurring level 6. Figure 4.4 shows the accuracy measures based on using shape sub-manifolds, rotation sub-manifolds with fine search. The highest accuracy is obtained when a fine search is applied at *k* equals to 3.

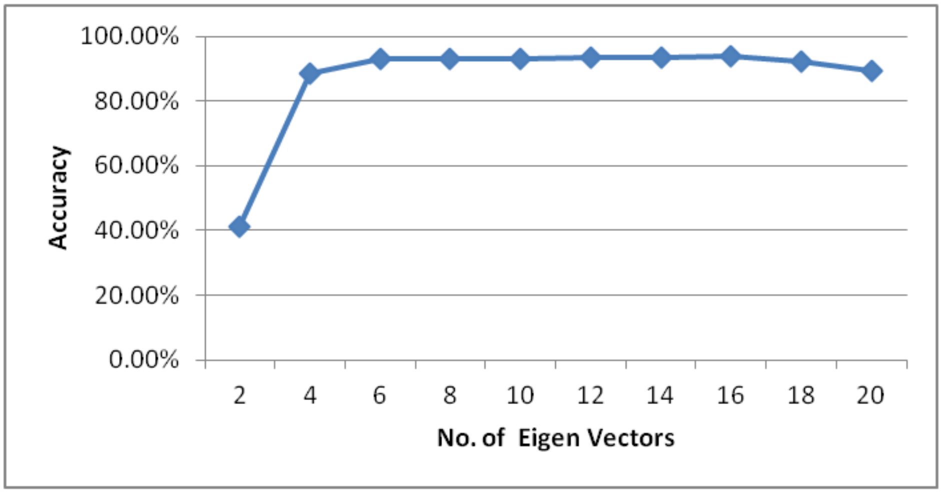


Figure 4.2, Accuracy for the 1st algorithm versus different number of eigenvectors at the 2nd stage using Rotation sub-manifolds and Shape sub-manifolds.

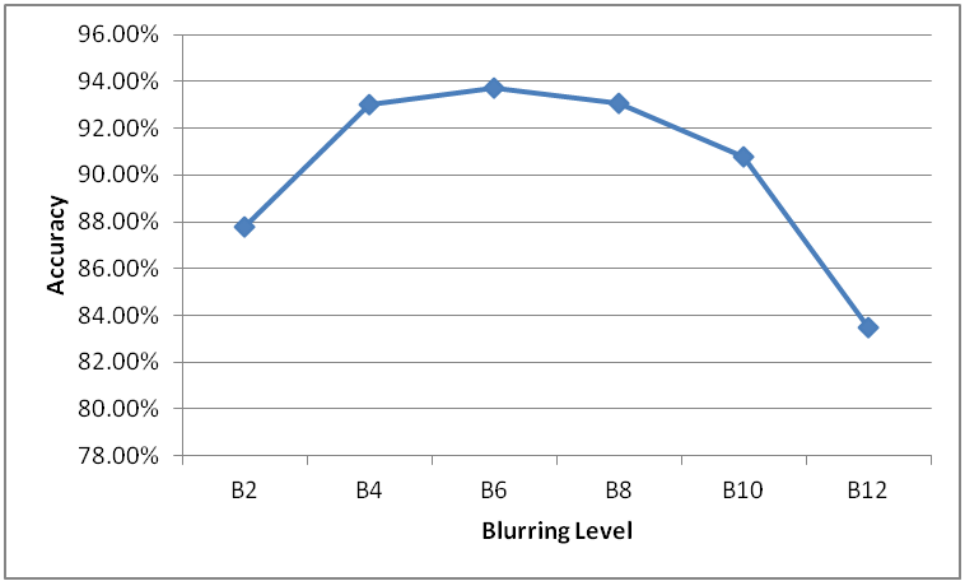


Figure 4.3, Accuracy for the 1st algorithm versus different blurring levels.

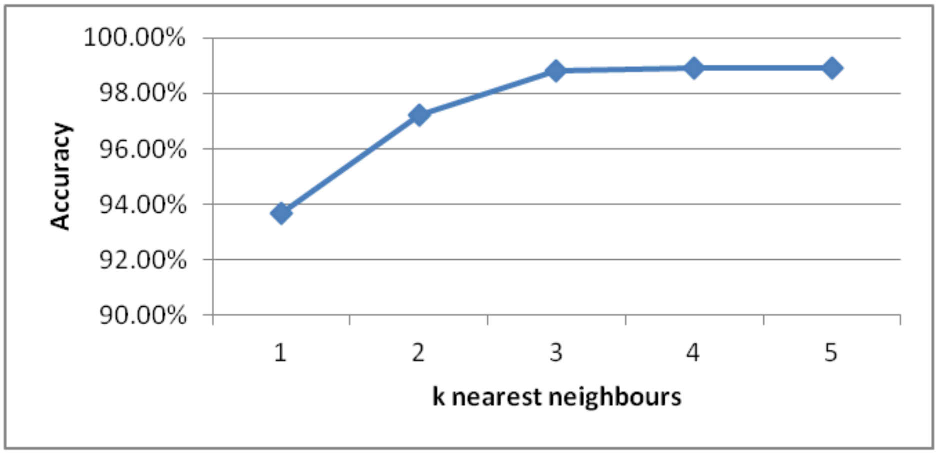


Figure 4.4, Accuracy of the 1st Algorithm under a fine search strategy.

Table 4.2 lists the accuracy measure for the three test sets under the different PCA spaces discussed in Section 3.6.5. The table also gives information about the speed of each aspect. The experimental results show that there is an enhancement in the performance using sub-manifolds over separate manifolds but a decrease in accuracy. However, the accuracy can be improved by using a fine search strategy with k equal to 3.

	Test set 1	**Test set 2**	**Test set 3**	**Speed**
Separate Manifolds	94.20%	90.46%	89.21%	2.6 sec
Shape Sub-Manifolds + Rotation Sub-Manifolds	93.70%	89.43%	88.58%	0.087 sec
Shape Sub-Manifolds + Rotation Sub-Manifolds + Fine Search	98.81%	94.35%	93.46%	0.094 sec

Table 4.2, Accuracy and performance of the 1st algorithm under different PCA spaces.

Based on using shape sub-manifolds and rotation sub-manifolds with fine search, Figure 4.5 shows the error analysis of the 121 translation positions within the manifolds over all objects. From the figure, it is clear that the error of the outer translated objects is greater. This is because of the non-linearity in the surface of the manifolds. Although image blurring is used to reduce the non-linearity, the surface of the generated manifolds is not completely flat.

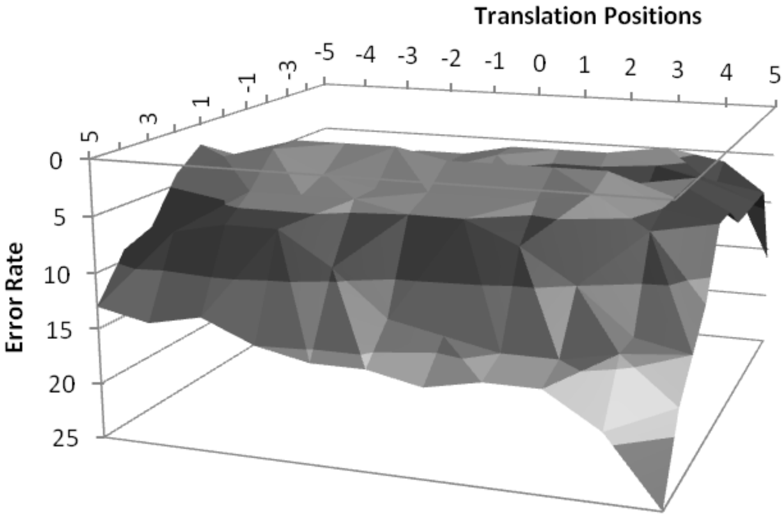


Figure 4.5, Error analysis for the 1st Algorithm.

4.3 Second Algorithm Experimental Results

To test the second proposed algorithm, different grid sizes under different blurring levels were used to examine the performance and accuracy. Different structures for the MDG in terms of the number of eigenvectors and the divisions along each direction are tested using different blurring levels. At Stage 1.A, the MDG size with the highest accuracy in estimating the rotation angle is chosen, subject to the constraint that the range of angles within all non-empty cells should be not be greater than 6 as discussed in section 3.7.1. Table 4.3 shows the maximum number of angles within the cells using five different structures for the MDG and at different blurring levels as well. Table 4.4 shows the accuracy of estimating the rotation angle at the same grid structures.

	10x10	**10x10x10**	**5x5x5x5**	**7x7x7x7**	**10x10x10x10**
B0	21	11	11	11	11
B2	21	16	11	11	11
B4	16	11	11	11	**6**
B6	16	11	11	**6**	**6**
B8	16	11	11	**6**	**6**
B10	16	11	11	11	**6**
B12	16	11	11	11	**6**

Table 4.3, Maximum range of angles at Stage 1.A for the 2nd algorithm.

	10x10	**10x10x10**	**5x5x5x5**	**7x7x7x7**	**10x10x10x10**
B0	98.2%	90.8%	94.9%	94.5%	85.8%
B2	98.6%	94.1%	94.7%	93.2%	84.9%
B4	99.0%	92.9%	94.2%	92.6%	**82.8%**
B6	98.9%	91.9%	95.3%	**91.2%**	**81.3%**
B8	99.0%	93.1%	92.9%	**91.1%**	**80.5%**
B10	98.0%	89.7%	87.7%	82.0%	**77.9%**
B12	98.5%	90.1%	93.8%	79.5%	**76.0%**

Table 4.4, Accuracy at Stage 1.A for the 2nd algorithm.

From the previous two tables, increasing the MDG size increases the number of cells and decreases the range of angles, as fewer objects are held by each cell. However, this reduces the accuracy of the process as more empty cells are generated within the MDG and the rotation angle of an incoming object can be misestimated in that case. Grid size of [7x7x7x7] with a blurring level of 6 is the best for constructing the MDG at stage 1.A. This grid size is based on the highest 4 eigenvectors where each direction is divided equally into 7 intervals. All the nonempty cells under this structure have a maximum of 6 different angles and an accuracy of 91.2% for estimating the rotation angle. To reach a precision of 4 rotation angles within the nonempty cells at Stage 1.B, the MDG size of [7x7x7x7] is used under blurring level 6. The same process as for Stage 1.A was repeated for Stage 1.B to determine the optimum MDG size. The accuracy of estimating the rotation angle at this size reached 94.2%. At Stage 2, MDGs of size [4x4x4] and blurring level 6 are used to partition the data into groups of similar objects and accelerate the search process. Figure 4.6 shows the effect of using different numbers of eigenvectors on the accuracy of detecting different hand shapes at the third stage using the first test-set.

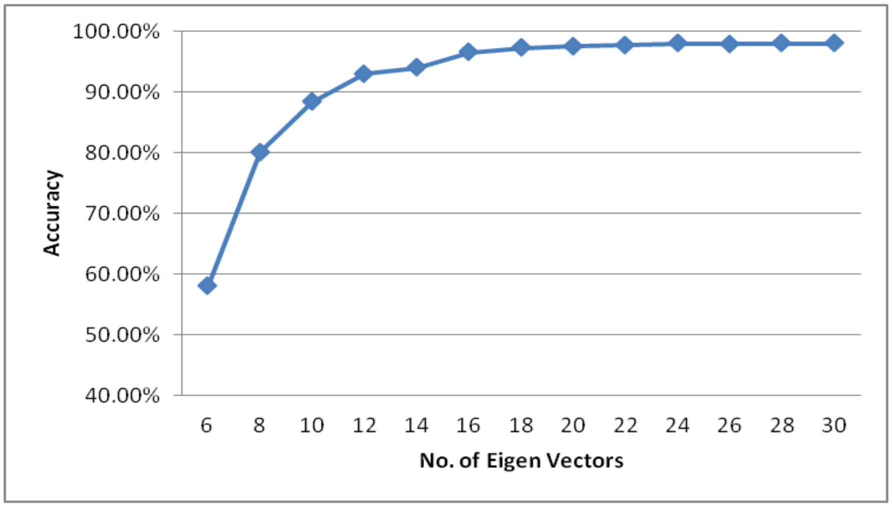


Figure 4.6, Accuracy versus different number of eigenvectors for the 2nd algorithm.

The following table shows the accuracy of the proposed algorithm in classifying different hand shapes using the three test-sets. At Stage 3, the first 30 eigenvectors were chosen to compute the Manhattan distance to find the nearest neighbour object, where each object takes up to 0.047 sec to be classified.

2^{nd} Algorithm	**Accuracy**
Test set 1	98.46%
Test set 2	93.56%
Test set 3	92.39%

Table 4.5, The Accuracy for the 2^{nd} algorithm.

Figure 4.7 shows the error analysis of the 121 translation position within the manifolds over all objects. The error analysis is for the first test set under the classification process for the hand shapes using the 2^{nd} proposed algorithm. From the figure, it is clear that the error of the outer translated objects is greater. This is because of the non-linearity in the surface of the manifolds. Although image blurring is used to reduce the non-linearity, the surface of the generated manifolds is not completely flat.

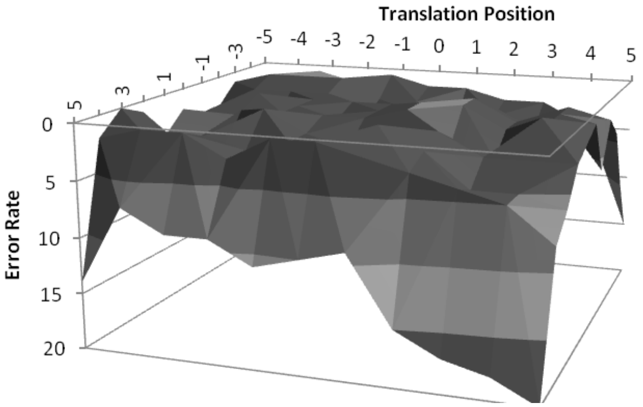


Figure 4.7, Error Analysis for the 2^{nd} Algorithm.

4.4 Third Algorithm Experimental Results

In order to examine the third algorithm, different grid sizes at different thresholds are used in testing to get the highest accuracy. Table 4.5 shows the accuracy of three different structures at minimum number of objects of 350 to build a new grid level. The classification process is achieved by finding the nearest neighbour using Manhattan distance. The highest 30 eigenvectors are used in the distance measure from applying PCA on the group of objects for each grid at the different levels to reach the highest accuracy. Figure 4.8 shows the effect of using different numbers of eigenvectors on the distance measure for the classification process for the first grid structure. Table 4.6 shows the grid analysis in terms of the number of grids in each level of the pyramid for the three grid structures. From Table 4.6 and Table 4.7, the first grid structure has the highest accuracy measure however most of the grids are at level II. The number of objects within the cells of the grids affects the accuracy. The more objects each cell holds, the higher accuracy is gained but slower the speed of the search.

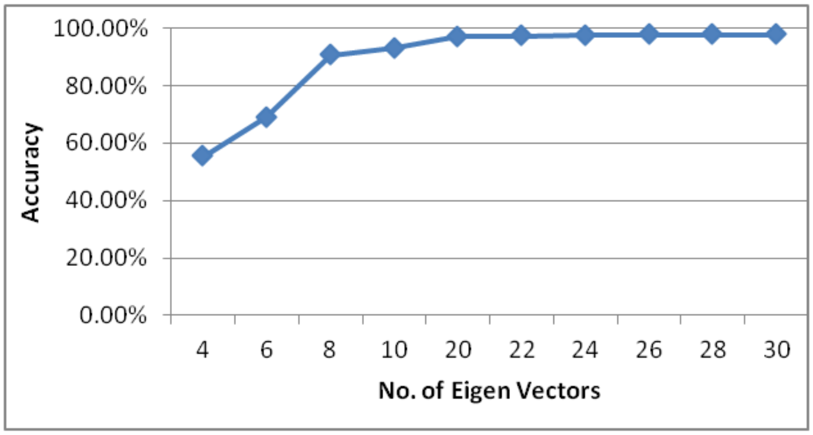


Figure 4.8, Accuracy versus different number of eigenvectors for the 3rd algorithm.

	[7x7x7x7] **[2x2x2x2]** **[2x2x2x2]**	**[4x4x4x4]** **[2x2x2x2]** **[2x2x2x2]**	**[4x4x4x4]** **[2x2x2x2]** **[2x2x2]**
Test set 1	97.75%	94.92%	95.45%

Table 4.6, Accuracy using different grid structures for the 3rd algorithm.

	[7x7x7x7] **[2x2x2x2]** **[2x2x2x2]**	**[4x4x4x4]** **[2x2x2x2]** **[2x2x2x2]**	**[4x4x4x4]** **[2x2x2x2]** **[2x2x2]**
Level II	94	36	36
Level III	13	149	149

Table 4.7, The number of grids in each level of the pyramid for three different grid structures.

To get a better pyramid hierarchy analysis, the grid size at the top level is reduced from [7x7x7x7] to [4x4x4x4]. The number of grids at the third level increases and fewer objects are held by each cell, Therefore, the accuracy drops. In order to enhance the results, the grid size in the third level was reduced from [2x2x2x2] to [2x2x2]. However, no enhancement in the accuracy was achieved. As each new level in this pyramid inherits the data objects from a cell in the parent grid, the number of dimensions used to build the new grid should be equal or greater than the dimensions used in the parent grid. Reducing the dimensionality causes a change in clustering the data which affects the accuracy.

To get a comparable accuracy with a better grid hierarchy, the second grid hierarchy is tested under different thresholds for the minimum number of objects needed to build a new level. Table 4.9 shows the accuracy measure for the second grid structure under three different conditions using the first test set and Table 4.8 shows the pyramid grid analysis for them.

	Min Objs = 400	Min Objs = 600	Min Objs = 800
Test set 1	95.09%	98.50%	98.91%

Table 4.8, Accuracy measure under different thresholds for the second grid structure.

	Min Objs = 350	Min Objs = 600	Min Objs = 800
Level II	36	33	29
Level III	149	40	11

Table 4.9, The number of grids in each level of the pyramid for the second grid structure under different thresholds.

From the previous two tables, the best accuracy at the best analysis is at a threshold of 600 as the minimum number to build a new grid at the next level. To enhance the results, a backtracking step can be done if an object is projected into an empty cell or at a cell with number of objects less than 100. The backtracking is achieved by projecting that object into the cell it was held by in the parent grid at the previous level. Table 4.10 shows the accuracy measure with backtracking for the three test sets using the second grid structure with a threshold of 600 to build a new level. Each object needs 0.055 sec to be classified using this proposed algorithm.

3r^{d} Algorithm	Accuracy
Test set 1	99.04%
Test set 2	95.14%
Test set 3	94.13 %
Real Hands	98.91 %

Table 4.10, Accuracy for the 3rd algorithm.

For the real data set, all the images are 250x330 pixels in size. The same grid structure is used to build the pyramid for the acquired images. The algorithm is tested with even frames from 100 frames per shape where the odd frames are used in constructing the grid structures through the hierarchy. The real hand training set is a dataset of 96,800 example images which is near to the size of the synthetic training set which is 111320. So the same grid structure is suitable to distribute the real hand images into the MDGs through the hierarchy where each cell can hold enough data to get a high accuracy measure.

From Table 4.10, the accuracy rate for the real data is almost as high as for the synthetic test set1 and actually higher than test set2 and test set3. The real data contains noise and deformation due to the fact that a human signer cannot keep his hand in exactly the same pose throughout the experiment. From the fact that the accuracy for the real data is comparable to that for the computer-generated data at this proposed algorithm, we can suggest that the proposed algorithms can be generalized to real-world data.

Figure 4.9 shows a surface for error rates of the 121 translated positions within the manifolds over all test objects using the 1st test set. From the figure, it is clear that the error of the outer translated objects is greater. This is because of the non-linearity in the surface of the manifolds. Although image blurring is used to reduce the non-linearity, the surface of the generated manifolds is not completely flat.

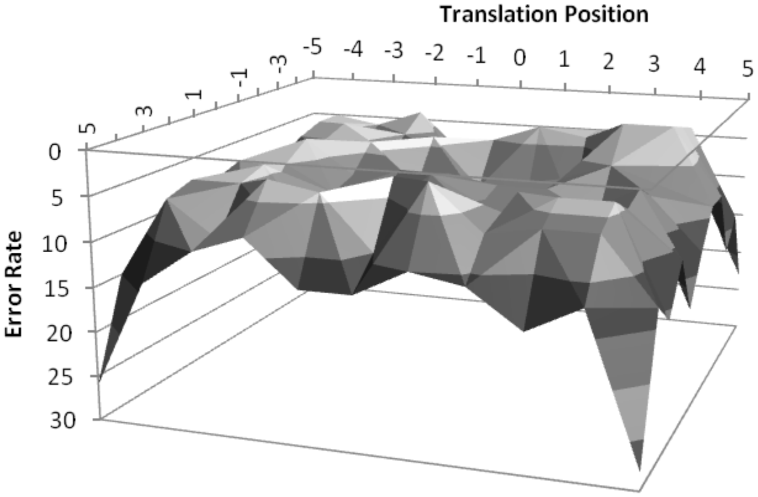


Figure 4.9, Error Analysis for the 3rd Algorithm.

4.5 Fourth Algorithm Experimental Results

To measure the accuracy and performance of the ANN pyramid, multilayer neural networks are trained with a FFBP algorithm based on the architecture shown in Table 4.11. The table shows the different neural network architecture in the different levels of the pyramid. The number of input neurons in the input layer is equal to the number of eigenvectors that are chosen to build the input eigenspaces, where each neuron corresponds to an eigenvector. In-order to choose the best number of eigenvectors, the training stage is repeated varying the number of eigenvectors. Figure 4.10 shows the effect of using different numbers of eigenvectors on the accuracy of estimating the hand shape by a neural network at the second stage where blurring level 6 is applied for the training and test sets. From the figure, 16 eigenvectors are sufficient for the classification process in-order to reach the highest possible accuracy.

	Stage 1.A	**Stage 1.B**	**Stage 2**
Input layer	4	4	16
Hidden layer 1	15	15	25
Hidden layer 2	7	7	20
Output layer	6	3	5

Table 4.11, Neural networks architecture designs at the different stages.

To investigate the effect of nonlinearity reduction using image blurring on training the ANN pyramid, the ANNs are trained using different blurring levels. Figure 4.11 shows the error rate after 1000 epochs in training the same neural network architecture under different blurring levels. It is apparent that increasing the blurring level helps in speeding up the training process because manifolds are more linear and therefore less nonlinear decision boundaries are needed. However, over-blurring may cause a loss in information and hence a drop in the accuracy measure.

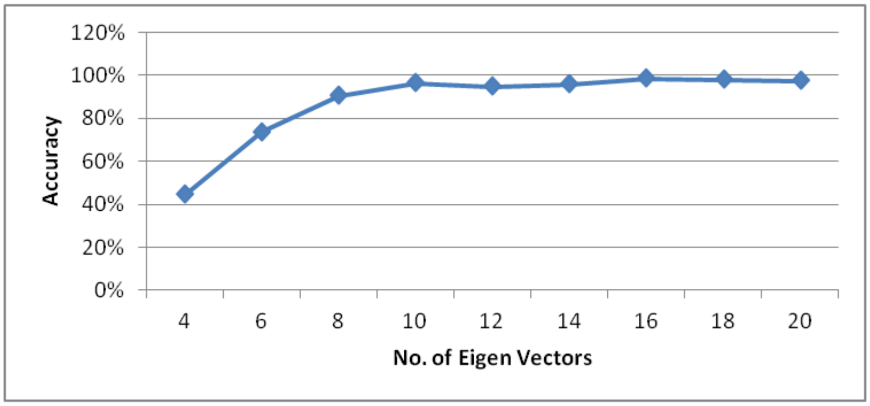


Figure 4.10, Accuracy versus different no. of eigenvectors at the 2nd stage for the 4th algorithm.

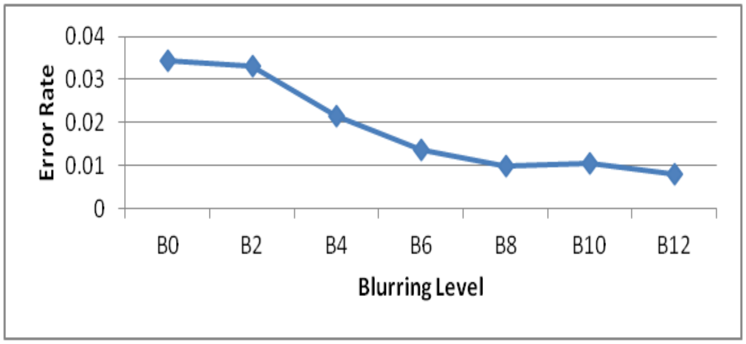


Figure 4.11, Training error rate versus different blurring levels.

Table 4.12 shows the accuracy of the proposed algorithm for the different test sets. Under the specified neural networks design and using 30 eigenvectors at the third stage to compute the Manhattan distance, the results show that neural networks based pyramid can classify different hand shapes with high level of accuracy. However, it is more sensitive to hand deformation and noise. The proposed algorithm achieves real-time performance where each object needs 0.062 seconds to be classified. The results come from the simplicity of the neural networks architecture, which is an advantage of using the proposed multistage hierarchy. Also, using PCA spaces has the advantage of low computation. New patterns need only matrix multiplication to be projected into the eigenspaces for features extraction and dimensionality reduction.

4th Algorithm	**Accuracy**
Test set 1	95.39%
Test set 2	85.36%
Test set 3	81.83%

Table 4.12, Accuracy for the 4th algorithm.

Figure 4.12 shows a surface for error rates of the 121 translated positions within the manifolds over all test objects using the first test set. From the figure, it is clear that the error of the outer translated objects is greater. This is because of the non-linearity in the surface of the manifolds. Although image blurring is used to reduce the non-linearity, the surface of the generated manifolds is not completely flat.

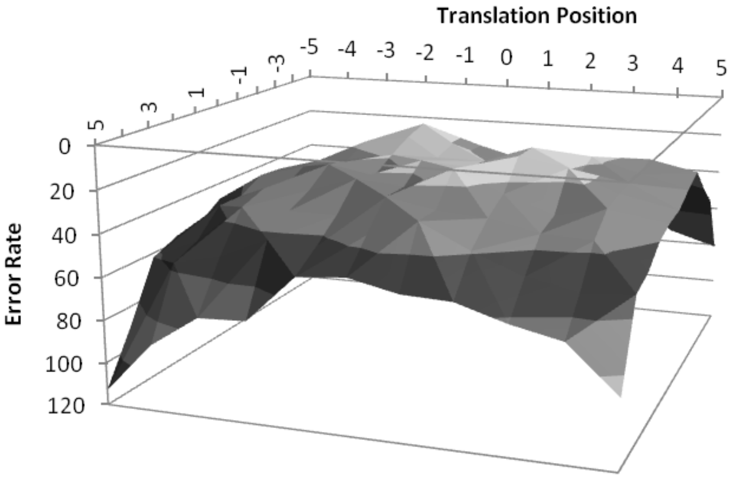


Figure 4.12, Error Analysis for the 4th Algorithm.

4.6 Experimental Results for the Nearest Neighbour Classifier using Exhaustive Search

The original problem involves finding the nearest neighbour object by searching through the whole range of manifolds of the training set using the perpendicular distance measure. A PCA space is generated for each shape at every specified rotation angle from the training range described before. The total number of PCA spaces is 920 where each eigenspace has its own set eigenvectors. The experimental results show that each test object needs 76.58 sec to be classified using the exhaustive search strategy. And it is time consuming to test with the whole test set. Instead, 10,000 random objects as a sample test set is used to test the accuracy of the exhaustive search strategy versus the different proposed classifiers under the settings discussed before. Table 4.13 shows the accuracy for the exhaustive search using the highest 30 eigenvectors. The results are compared with the four proposed algorithms using the specified random sample using three test sets.

	Exhaustive Search	**1st Algo**	**2nd Algo**	**3rd Algo**	**4th Algo**
Test set 1	99.98 %	99.03 %	96.97 %	98.93 %	95.32 %
Test set 2	97.28 %	94.18 %	92.76 %	94.96 %	83.94 %
Test set 3	96.45 %	93.03 %	91.70 %	94.20 %	80.69 %
Speed	76.58 sec	0.094 sec	0.047 sec	0.055 sec	0.062 sec

Table 4.13, Accuracy and Performance for the exhaustive search versus all proposed techniques based on a random sample of 10,000 test objects.

Table 4.14 shows the speed up factor for the proposed algorithms against the exhaustive search strategy.

	1st Algo	2nd Algo	3rd Algo	4th Algo
Speed Up	814.68	1629.36	1392.36	1235.16

Table 4.14, The speed up factor for the proposed algorithms against the exhaustive search.

4.7 Experimental Results - Conclusion

From the previous results, we can conclude that the proposed algorithms are applicable for real-time applications under the assumption that the classification rate should be at least 10 frames per second. The proposed algorithms accelerate the search process by a high speed-up factor reaching 1629.36 times over the exhaustive search process. The test results show a high level of accuracy for the recognition process. However, the ANN pyramid has the lowest robustness to deformation and noise. From the results, the best performance is for the second algorithm using supervised Multidimensional Grids. And the best accuracy measure is for the third algorithm using unsupervised Multidimensional Grids. The third algorithm also is applicable for real hand images where the experimental results show that the accuracy rate for the real data is almost as high as for the synthetic test sets.

Chapter 5

Conclusion and Future Work

7.4 Introduction

In this thesis, we have introduced four algorithms for hand shape recognition using a large set of example images. The example images are computer generated to ease collecting a large training dataset with known pose parameters. The dataset represents 20 different hand shapes from the Irish Sign Language at different rotations and translations. PCA is used as a feature extraction and dimensionality reduction method. A large number of eigenspaces are constructed using PCA. To accelerate the search process to classify new patterns, a data pyramid is used in a multistage hierarchy to approximate the k nearest neighbour problem. Through the pyramid, the new image is analysed with different PCA spaces. Estimation for a parameter of the hand shape is made at each level of the pyramid where a final decision is taken at the bottom end level. The new pattern is labelled according to the nearest neighbour object from the training set which represents the top match. Four pattern recognition techniques are used to implement the proposed pyramid hierarchy viz. perpendicular distance measure, supervised multi-dimensional grids, unsupervised multi-dimensional grids and artificial neural networks. The experimental results showed that the technique is applicable for real-time applications under a high degree of accuracy.

7.5 Summary of Contributions

In this section a brief summary is given for the main contributions that have been discussed in this thesis through the previous chapters.

- The proposed algorithms do not require a pre-alignment of the images as a pre-processing step. Image alignment is required in many techniques, which are found in the literature, to rotate the hand shape or scale it to pre-defined position to ease the classification process. The proposed algorithms fall under the category of example-based approaches. They depend on the strategy of building a training set using computer-generated example images as it is possible to control the position and orientation of the hand and arm at regular intervals, which is difficult for a human being to do. The example images densely sample the space to cover different translations and rotations. The proposed algorithms overcome the disadvantage of PCA as they are invariant to these rigid transformations where PCA originally is not.
- A new way of organizing data pyramids has been discussed through the thesis. The proposed PCA pyramids analyse the example images using different eigenspaces throughout the hierarchy. Usually, data pyramids are constructed under different image resolutions or scales. Estimation for a hand shape parameter can be achieved at each level using the related eigenspaces. Originally, PCA is an unsupervised learning technique that builds the eigenspaces based on the feature space of the original data without taking into consideration the different classes, which represent the different poses and shapes of the hands. Different PCA spaces

were introduced in this thesis. Rotation manifolds are generated by building one eigenspace for a shape at different rotations. A shape manifold is a manifold that holds different shapes at a certain rotation angle within the same eigenspace. Sub-manifolds are generated by applying PCA a second time to each eigenspace in-order to get a further dimensionality reduction. Estimation for the rotation angle of the signer arm is achieved at the first level of the hierarchy using rotation manifolds as rotation has the highest variation within the data. Then shape manifolds are used for classifying the shape at the second stage.

- Nonlinearity reduction in manifolds using image blurring has been introduced in this thesis and tested on our set of example images. Image blurring is used to remove small changes between objects to enhance the generalization factor with respect to new images. However, image blurring has the effect of making the manifolds more flat and more sparable within the eigenspaces. This helps the different classifiers to reach a higher level of accuracy. The first algorithm, which uses perpendicular distance, originally assumes that the manifolds are completely flat and reducing the nonlinearity gets more accurate distance measures. In the second and third algorithms, multi-dimensional grids can be used to cluster the data into groups of similar objects using hyper-cubic cells. At a certain level of blurring, the data is separable enough to label the MDG cells with the class of data they hold. In the fourth algorithm, artificial neural networks can be trained faster with a simpler design, when nonlinearity reduction is achieved using image blurring as fewer nonlinear decision boundaries are needed to solve the problem.

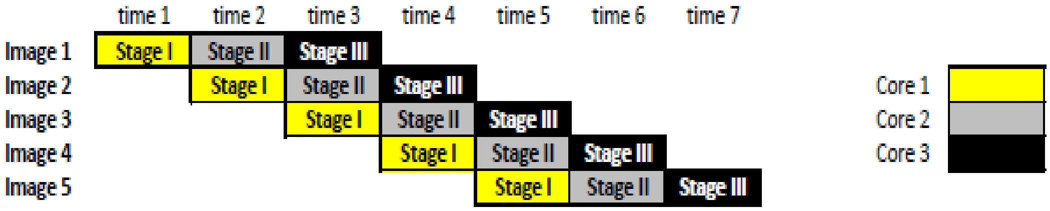
- Acceleration for the search process in the example images over exhaustive search strategy is discussed through the thesis. The proposed PCA pyramids provide an alternative to the indexing method to approximate the k nearest neighbour problem. Indexing methods usually are sensitive to the choice of the hash functions. Through the hierarchy, the search focuses on a group of images according to the estimations of the parameters for the new hand shape which are made in each level.

- The proposed PCA pyramids provide a method to design simple architectures for the classification process. As the PCA eigenspaces are organized into groups of objects sharing the same parameter, simple design architectures can achieve the learning stage. The simplicity of the design can be expressed in terms of the size of MDGs or the number of neurons and layers in the MLP architecture. Nonlinearity reduction in manifolds helps also in finding a simpler design to solve the recognition problem, as less nonlinear decision boundaries are needed to separate the data.

7.6 Directions for Future Research

In this section a suggestion is made to demonstrate the future work based on the research which is made in this thesis.

- The proposed pyramids are parallel by nature. A pipeline can be established using multi-core processors to enhance the speed of the system. The number of pipeline stages is equal to the number of levels the pyramid has. Figure 5.1 show a suggestion for a 3 stage pipeline. The speed of the system can be enhanced to be equal the speed of the slowest stage.



	time 1	time 2	time 3	time 4	time 5	time 6	time 7
Image 1	Stage I	Stage II	Stage III				
Image 2		Stage I	Stage II	Stage III			
Image 3			Stage I	Stage II	Stage III		
Image 4				Stage I	Stage II	Stage III	
Image 5					Stage I	Stage II	Stage III

Core 1 🟨
Core 2 ⬜
Core 3 ⬛

Figure 5.1, Pipeline design for a pyramid hierarchy.

- The proposed pyramid hierarchy is upgradable. The pyramid structure can be appended by extra levels at the start to study and analyze the effect of other factors like scale. This has the effect of increasing the robustness of the proposed algorithms.

- A different implementation for the proposed PCA pyramid hierarchy can be achieved using support vector machines. The nonlinearity reduction in manifolds can help in speeding up the training process of SVMs as less nonlinear decision boundaries can easy be established by the support vectors.

- Towards achieving a user-independent framework, a preprocessing stage can be established for mapping a real signer segmented arm and hand to a computer generated image to be classified by the system. The fact that the accuracy for the real data is comparable to that for the computer-generated data at the third algorithm, we can suggest that the proposed algorithms can be generalized to real-world data. To extend the real-world evaluation, the training and test data can be drawn from different people, separate sequences and over a wider population of transformations.

- A study can be made for checking and generalizing the effect of image blurring on reducing the nonlinearity in manifolds on other nonlinear data sets eg. face recognition, behavior tracking, and vehicle tracking.

- The proposed multistage hierarchy can be applied for different applications of pose estimation eg. head pose estimation and body postures recognition.

Publications arising from the research in this thesis

- Farouk M., Sutherland A. and Shokry A. "A multistage hierarchical algorithm for hand shape recognition." *IEEE 13th International n Machine Vision and Image Processing Conference IMVIP'09.*, 2009, pp. 105-110.

- Farouk M., Sutherland A. and Shokry A., "Nonlinearity Reduction of Manifolds using Gaussian Blur for Handshape Recognition based on Multi-Dimensional Grids" , *2nd International Conference on Pattern Recognition Applications and Methods ICPRAM 13*, 2013, pp.303-307.

References

[1] Moeslund T. and Granum E., "A survey of computer vision-based human motion capture.", *Computer Vision and Image Understanding*, 2001, 8(3), pp. 231-268.

[2] Turaga P., Chellappa R., Subrahmanian V. and Udrea O., "Machine recognition of human activities: A survey.", *IEEE Transactions on Circuits and Systems for Video Technology*, 2008, 18(11), pp.1473-1488.

[3] McInerney T., and Terzopoulos D., "Deformable models in medical image analysis: a survey.", *Medical image analysis*, 1996, 1(2), pp. 91-108.

[4] Binh N., Shuichi E. and Ejima T., "Real-time hand tracking and gesture recognition system." *Proc. GVIP* , 2005, PP. 19-21.

[5] Wang C., and Wang k., "Hand Posture recognition using Adaboost with SIFT for human robot interaction." *Recent progress in robotics: viable robotic service to human*, *Springer Berlin Heidelberg*, 2008, pp. 317-329.

[6] Khan Z., and Ibraheem., "Survey on gesture recognition for hand image postures." *Computer and Information Science*, 2012, 5(3), pp. 110-116.

[7] Wu Y., Lin J. and Huang T., "Capturing natural hand articulation." *IEEE Eighth International Conference on Computer Vision*, 2001, vol. 2, pp. 426-432.

[8] Zaki M. and Shaheen S., "Sign language recognition using a combination of new vision based features.", *Pattern Recognition Letters*, 2011, 32(4) pp. 572-577.

[9] Gorban A., Kegl B., Wunsch D. and Zinovyev A., eds, “Principal manifolds for data visualization and dimension reduction.”, Springer Lecture Notes in Computational Science and Engineering, 2007, Vol. 58.

[10] Potamias M. and Athitsos V., "Nearest neighbor search methods for handshape recognition." *ACM 1st international conference on PErvasive Technologies Related to Assistive Environments*, 2008, p. 30.

[11] Wei J., Qin H., Guo J., and Chen Y., "The hand shape recognition of human computer interaction with artificial neural network." *IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurements Systems, VECIMS'09.*, 2009, pp. 350-354.

[12] Mitra S. and Acharya T., "Gesture recognition: A survey." *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* 2007, 37(3), pp. 311-324.

[13] Wu Y. and Huang T., "Vision-based gesture recognition: A review." *Gesture-based communication in human-computer interaction*, *Springer Berlin Heidelberg*, 1999, pp. 103-115.

[14] Radha H., G., Shruti S. and Kanya B., "DESIGN AND DEVELOPMENT OF AN ASSISTIVE DEVICE FOR SPEECH AND HEARING IMPAIRED." *Internation Journal of Innovation Technology and Research IJITR* 2, 2014, pp. 859-862.

[15] Garg P., Aggarwal N. and Sofat S., "Vision based hand gesture recognition." *World Academy of Science, Engineering and Technology*, 2009, 49(1), pp. 972-977.

[16] Kumar P., Verma J. and Prasad S., "Hand Data Glove: A Wearable Real-Time Device for Human-Computer Interaction." *International Journal of Advanced Science and Technology*, 2012.

[17] Bilal S., Akmeliawati R., El Salami M. and Shafie Amir., "Vision-based hand posture detection and recognition for Sign Language A study.", *IEEE 4th International Conference On Mechatronics (ICOM),* 2011, pp. 1-6.

[18] Tofighi G., Monadjemi S. and Ghasem N., "Rapid hand posture recognition using Adaptive Histogram Template of Skin and hand edge contour." *IEEE 6th Iranian Machine Vision and Image Processing (MVIP),* 2010, pp. 1-5.

[19] Erol A., Bebis G., Nicolescu M., Boyle R. and Twombly X., "Vision-based hand pose estimation: A review." *Computer Vision and Image Understanding*, 2007, 108(1), pp. 52-73.

[20] Takimoto H., Yoshimori S., Mitsukura Y., and Fukumi M., "Classification of hand postures based on 3d vision model for human-robot interaction." *IEEE RO-MAN*, 2010, pp. 292-297.

[21] Adelson E., Anderson C., Bergen J., Burt P. and Ogden J. "Pyramid methods in image processing." *RCA engineer*, 1984, 29(6), pp. 33-41.

[22] Lazebnik S., Schmid C. and Ponce J., "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2006, vol. 2, pp. 2169-2178.

[23] Roth P. and Winter M., "Survey of appearance-based methods for object recognition." *Technical Report ICGTR0108 Inst. for Computer Graphics and Vision, Graz University of Technology, Austria*, 2008.

[24] Pless R. and Souvenir R., "A survey of manifold learning for images." *IPSJ Transactions on Computer Vision and Applications*, 2009, pp. 83-94.

[25] LaViola J. and Joseph J., "A survey of hand posture and gesture recognition techniques and technology." *Brown University, Providence, RI* ,1999.

[26] Huang D., Hu W., Chang S. and Chen M., “Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination.” *Expert Systems With Applications*, 2010, pp.6031–6042.

[27] Shahbudin S., Hussain A., Hussain H., Samad S. and Tahir M., "Analysis of PCA based feature vectors for SVM posture classification." *IEEE 6th International Colloquium on Signal Processing and Its Applications (CSPA)*, 2010, pp. 1-6.

[28] Gastaldi G., Pareschi A., Sabatini S., Solari F. and Bisio M., "A man-machine communication system based on the visual analysis of dynamic gestures.", *International Conference on Image Processing*, 2005, pp.397-400.

[29] Coogan T., Awad G., Han J., and Sutherland A., "Real time hand gesture recognition including hand segmentation and tracking." *Advances in Visual Computing, Springer Berlin Heidelberg*, 2006, pp. 495-504.

[30] Coogan T and Sutherland A., "Dynamic gesture recognition using transformation invariant hand shape recognition." *PhD diss., Dublin City University*, 2007.

[31] Schikuta E., "Grid-clustering: An efficient hierarchical clustering method for very large data sets." *IEEE 13th International Conference on Pattern Recognition,* 1996, vol. 2, pp. 101-105.

[32] Amini A., Wah T., Saybani M and Yazdi S., "A study of density-grid based clustering algorithms on data streams." *IEEE Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),* 2011, vol. 3, pp. 1652-1656.

[33] Wani M., "Introducing Subspace Grids to Recognise Patterns in Multidimensinal Data." *IEEE 11th International Conference on Machine Learning and Applications (ICMLA),* 2012, vol. 1, pp. 33-39.

[34] Murakami K. and Taguchi H., "Gesture recognition using recurrent neural networks." *ACM conference on Human factors in computing systems*, 1991, pp. 237-242.

[35] Shi Y., Chen X., Wang K., Fang Y. and Xu L., "Real-time hand posture analysis based on neural network.", *IEEE 10th International Conference on Signal Processing (ICSP)*, 2010, pp. 893-896.

[36] Htike K. and Khalifa O., "Comparison of supervised and unsupervised learning classifiers for human posture recognition.", *IEEE International Conference on Computer and Communication Engineering (ICCCE),* 2010, pp. 1-6.

[37] Oniga S., Tisan A., Mic D., Buchman A. and Vida-Ratiu A., "Hand postures recognition system using artificial neural networks implemented in FPGA.", *IEEE 30th International Spring Seminar on Electronics Technology*, 2007, pp. 507-512.

[38] Naidoo S., Omlin C. and Glaser M., "Vision-based static hand gesture recognition using support vector machines." *Department of Computer Science, University of the Western Cape, South Africa*, 1999.

[39] Dardas, Nasser H., and Nicolas D. Georganas. "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques." *Instrumentation and Measurement, IEEE Transactions on* 60, no. 11 (2011): 3592-3607.

[40] Cernekova Z., Nikolaidis N. and Pitas I., "Single camera pointing gesture recognition using spatial features and support vector machines.", *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*, 2007, pp. 130-134.

[41] Weimin H., Wai A., Foo S., Biswas J., Hsia C. and Liou K., "Multimodal sleeping posture classification." *IEEE International Conference on Pattern Recognition (ICPR)*, 2010, pp. 4336-4339.

[42] Stainvas I. and Intrator N., "Blurred face recognition via a hybrid network architecture.", *IEEE 15th International Conference on Pattern Recognition*, 2000, vol. 2, pp. 805-808.

[43] Zhaoxue C., Shengdong N., Lijun Q., Zeng C. and Jianrong X., "Automatic liver segmentation method based on a gaussian blurring technique for CT images.", *IEEE 2nd International Conference on Bioinformatics and Biomedical Engineering ICBBE* , 2008, pp. 2516-2518.

[44] Moeslund T., Hilton A. and Krüger V., "A survey of advances in vision-based human motion capture and analysis.", *Computer vision and image understanding*, 2006, 104(2), pp. 90-126.

[45] Wang R. and Popović J., "Real-time hand-tracking with a color glove." *ACM Transactions on Graphics*, 2009, 28(3), pp. 63-69.

[46] Athitsos V. and Sclaroff S., "Estimating 3D hand pose from a cluttered image." *IEEE Computer Vision and Pattern Recognition Proceedings*, 2003, vol. 2, pp. 424-432.

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

[48] Patil M. and Subbaraman S., "A Review On Vision Based Hand Gesture Recognition Approach Using Support Vector Machines." *Journal of Electronicsand Communication Engineering*, 2012, pp. 07-12.

[49] Wang J., Wang C. and Lee J., "Genetic eigenhand selection for handshape classification based on compact hand extraction." *Engineering Applications of Artificial Intelligence*, 2013, 26(9), pp. 2215-2226.

[50] http://web.mit.edu/6.454/www/www_fall_2003/ihler/slides.pdf.

[51] Hassanein S., Khalifa, Al-Atabany W. and El-Wakad M., "Performance of Optical Flow tracking approaches for cardiac motion analysis.", *Middle East Conference on Biomedical Engineering*, 2014, pp. 143-146.

[52] Campos M., Ferreira M., Martins T. and Santos C., "Inspection of bottles crates in the beer industry through computer vision." *IEEE 36th Annual Conference on Industrial Electronics Society*, 2010, pp. 1138-1143.

[53] Du X. and Tan K., "Autonomous Reverse Parking System Based on Robust Path Generation and Improved Sliding Mode Control.", *IEEE Transactions on Intelligent Transportation Systems*, 2014, pp.1-13.

[54] Martinetz T. and Schulten K., “A neural-gas network learns topologies.”, *University of Illinois at Urbana-Champaign*, 1991.

[55] Cao J., Chua K., Chong W., Lee H. and Gu Q., "A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine." *Neurocomputing*, 2003, 55(1), pp. 321-336.

[56] Witkin A., "Scale-space filtering: A new approach to multi-scale description.", *IEEE International Conference on Acoustics, Speech, and Signal Processing,* 1984, vol. 9, pp. 150-153.

[57] Guttman A.,"R-trees: a dynamic index structure for spatial searching.”, *ACM SIGMOD international conference on Management of data*, 1984, 14(2), pp. 47 – 57.

[58] Barrow H., Tenenbaum J., Bolles R. and Wolf H., “Parametric correspondence and chamfer matching: Two new techniques for image matching.”, *SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER*, 1977.

[59] Scholz M., Fraunholz M. and Selbig J., "Nonlinear principal component analysis: neural network models and applications." *Principal manifolds for data visualization and dimension reduction*, *Springer Berlin Heidelberg*, 2008, pp. 44-67.

[60] Kruger U., Zhang J. and Xie L., "Developments and applications of nonlinear principal component analysis a review." *Principal manifolds for data visualization and dimension reduction*, *Springer Berlin Heidelberg*, 2008, pp. 1-43.

[61] Schölkopf, B., Smola, A., and Müller K., "Kernel principal component analysis.", *ICANN'97 Lecture Notes in Computer Science*, *Springer Berlin Heidelberg*, 1997, vol. 1327, pp. 583-588.

[62] Ghodsi A., "Dimensionality reduction a short tutorial.", *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada*, 2006.

[63] Roweis, S. and Saul L., "Nonlinear dimensionality reduction by locally linear embedding. ", *Science Journal*, 2000, 290(5500), pp. 2323-2326.

[64] Tenenbaum J., Silva V. and Langford J.," A global geometric framework for nonlinear dimensionality reduction.", *Science Journal*, 2000, 290(5500), pp. 2319-2323.

[65] Skiena, S.,"Dijkstra's Algorithm.", *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*,1990, pp. 225-227.

[66] Khoshelham K.,"Accuracy analysis of kinect depth data.", *ISPRS workshop laser scanning*, 2011, 38(5).

[67] Gedraite E. and Hadad M., "Investigation on the effect of a Gaussian Blur in image filtering and segmentation.", *IEEE 53rd International Symposium ELMAR, Zadar Croatia*, 2011, pp. 393-396.

[68] http://my.smithmicro.com/poser-3d-animation-software.html.

[69] Raymond T. and Jiawei Han., "CLARANS: A method for clustering objects for spatial data mining." *IEEE Transactions on Knowledge and Data Engineering*, 2002, 14(5), pp. 1003-1016.

[70] Farouk M., Sutherland A. and Shokry A. "A multistage hierarchical algorithm for hand shape recognition." *IEEE 13th International n Machine Vision and Image Processing Conference IMVIP'09.*, 2009, pp. 105-110.

[71] Farouk M., Sutherland A. and Shokry A., "Nonlinearity Reduction of Manifolds using Gaussian Blur for Handshape Recognition based on Multi-Dimensional Grids" , *2nd International Conference on Pattern Recognition Applications and Methods ICPRAM 13*, 2013, pp.303-307.