

# ***Dynamic Gesture Recognition using Transformation Invariant Hand Shape Recognition***

*Thomas A. Coogan*

*Submitted in fulfilment of the requirements for M.Sc.  
Degree*

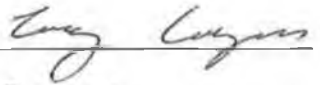
School of Computer Applications  
Dublin City University  
Ireland

Supervisor: Dr. Alistair Sutherland

*2007*

## ***DECLARATION***

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Msc, is entirely my own work 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:  (Candidate)

ID No.: 54147867

Date: 17-09-07

# **TABLE OF CONTENTS**

<i>CHAPTER 1 - INTRODUCTION</i> .....	2
1.1 Overview .....	2
1.2 Sign Language Recognition .....	2
1.3 Computer Vision in Gesture Recognition .....	3
1.4 Approaches to Sign Language Recognition .....	5
1.5 Hand Gesture Recognition.....	6
1.6 Outline of the Thesis .....	7
<i>CHAPTER 2 - LITERATURE REVIEW</i> .....	8
2.1 Introduction.....	8
2.2 Static Hand Gesture Recognition.....	9
2.2.1 Nearest neighbour and Cluster analysis .....	9
2.2.2 Template Matching and PCA.....	10
2.2.3 Contour and Silhouette.....	14
2.2.4 Elastic Graph Matching.....	16
2.2.5 Texture Based Pattern Recognition Techniques .....	17
2.3 Dynamic Gesture Recognition .....	18
2.4 Using Computer Animation and Poser .....	22
2.7 Summary.....	24
<i>CHAPTER 3</i>	
<i>STATIC GESTURE RECOGNITION – SUBSPACE APPROACH</i> .....	25
3.1 Introduction.....	25
3.2 Hand-shape Transformations .....	26
3.2.1 Translation Transformations .....	26
3.2.2 Rotation Transformations .....	27
3.2.3 Scale Transformations.....	29
3.2.4 Colour Transformations .....	29
3.3 Transformation Invariance .....	30
3.4 Creating Training Database .....	32
3.5 Subspace System Overview .....	34
3.6 Reducing Search Time .....	36
3.7 Experiments.....	38
3.7.1 Translation Transformation Experiments .....	39
3.7.2 Combining Rotation and Translation Transformations Experiments .....	43
3.7.3 Combining Rotation Translation and Shape Transformations Experiments ...	44
3.8 Summary.....	46

<i>CHAPTER 4</i>	
<i>REAL HAND IMAGE CLASSIFICATION</i> .....	48
4.1 <i>Introduction</i> .....	48
4.2 <i>Hand Image Pre-processing</i> .....	48
4.2.1 <i>Hand Segmentation</i> .....	50
4.2.2 <i>Hand Scaling and Alignment</i> .....	51
4.2.3 <i>Skin Colour and Illumination Variation</i> .....	52
4.2.4 <i>Image Filtering</i> .....	55
4.3 <i>Recognition Experiments</i> .....	55
4.4 <i>Colour Invariance Experiments</i> .....	59
4.5 <i>Noise Reduction Experiments</i> .....	61
4.6 <i>Summary</i> .....	62
<i>CHAPTER 5</i>	
<i>DYNAMIC GESTURE RECOGNITION</i> .....	64
5.1 <i>Introduction</i> .....	64
5.2 <i>Discrete Hidden Markov Models – An Overview</i> .....	65
5.2.1 <i>The evaluation Problem</i> .....	66
5.2.2 <i>The decoding problem</i> .....	67
5.2.3 <i>The Learning Problem</i> .....	67
5.3 <i>Input Observations for DHMM</i> .....	67
5.3.1 <i>Hand-shape Classification</i> .....	68
5.3.2 <i>Hand Position Classification</i> .....	68
5.4 <i>Experiments</i> .....	71
5.5 <i>Summary</i> .....	78
<i>CHAPTER 6</i>	
<i>CONCLUSIONS AND FUTURE WORK</i> .....	79
6.1 <i>Summary</i> .....	79
6.2 <i>Future Work</i> .....	81
<i>APPENDIX A- ISL HANDSHAPES</i> .....	84
<i>APPENDIX B – PERFORMING PCA ON A SET OF IMAGES</i> .....	85
<i>APPENDIX C – DYNAMIC GESTURES</i> .....	87
<i>APPENDIX D – CODE LISTING</i> .....	92
<i>REFERENCES</i> .....	93
<i>Publications Arising From This Thesis</i> .....	97

# Abstract

In this thesis a detailed framework is presented for accurate real time gesture recognition. Our approach to develop a hand-shape classifier, trained using computer animation, along with its application in dynamic gesture recognition is described. The system developed operates in real time and provides accurate gesture recognition. It operates using a single low resolution camera and operates in Matlab on a conventional PC running Windows XP.

The hand shape classifier outlined in this thesis uses transformation invariant subspaces created using Principal Component Analysis (PCA). These subspaces are created from a large vocabulary created in a systematic manner using computer animation. In recognising dynamic gestures we utilise both hand shape and hand position information; these are two of the main features used by humans in distinguishing gestures. Hidden Markov Models (HMMs) are trained and employed to recognise this combination of hand shape and hand position features.

During the course of this thesis we have described in detail the inspiration and motivation behind our research and its possible applications. In this work our emphasis is on achieving a high speed system that works in real time with high accuracy.

# Glossary of Acronyms

<i>ASL</i>	<i>American Sign Language</i>
<i>BSL</i>	<i>British Sign Language</i>
<i>CHMM</i>	<i>Continuous Hidden Markov Models</i>
<i>CSL</i>	<i>Chinese Sign Language</i>
<i>DHMM</i>	<i>Dynamic Hidden Markov Models</i>
<i>EM</i>	<i>Expectation-Maximization</i>
<i>HCI</i>	<i>Human Computer Interaction</i>
<i>HMM</i>	<i>Hidden Markov Models</i>
<i>ISL</i>	<i>Irish Sign Language</i>
<i>LBP</i>	<i>Linear Binary Patterns</i>
<i>MDA</i>	<i>Multiple Discriminant Analysis</i>
<i>PCA</i>	<i>Principle Component Analysis</i>

# ***CHAPTER 1***

## ***INTRODUCTION***

### **1.1 Overview**

The primary goal of any automated gesture recognition system is to create an interface that is natural for humans to operate or communicate with a computerised device. In the real world gesture occupies a major role in human interaction. We use gesture to point and direct, when speaking gesture is used to emphasise emotion, attitude, purpose and association. This routine use of gesture in communication and direction by humans suggests that any Human Computer Interaction (HCI) should ideally involve gesture. Some notable tasks that could be substantially improved by incorporating gesture would be virtual reality, robot manipulation and gaming. However, gesture recognition could be used to improve the intuitiveness of any HCI system. In most cases HCI is achieved using unnatural low dimensional dedicated devices such as mouse, keyboard and joysticks. Over a period of time we have trained ourselves to use these devices. Instead of forcing humans to adapt and use these interfacing devices traditionally offered by computers, it would be advantageous if the computer could learn human natural interfacing techniques. An incorporation of gestures with HCI could be an extremely beneficial development towards improving the intuitiveness of HCI.

## 1.2 Sign Language Recognition

One of the long term goals of gesture recognition is to develop a computer-based sign language translation system that can recognise a subset of an existing sign language and translate it to text format. Sign languages are the native languages of Deaf communities throughout the world. Sign languages are distinct languages in their own right with their own vocabularies and grammars. Up to now the Deaf have had to communicate with the Hearing either through an interpreter or through written forms of spoken languages, which are not the native languages of the Deaf community. This limits their access to information, education, employment, culture, participation in the community and legal and political representation. Another important point to consider is that many different countries have their own independent sign language, such as Irish Sign Language (ISL), British Sign Language (BSL), American Sign Language (ASL) and Chinese Sign Language (CSL). This means that communication between an Irish Deaf person and a British Deaf person is just as difficult as a native English speaker communicating with a non-native English speaker. A computer-based sign language translation system would increase the opportunities open to the Deaf community.

In order to make such a system available and acceptable it needs to run efficiently in real-time. Instead of using high-end processing or servers to perform this compute-intensive task, any gesture recognition system should be developed and implemented on a standard personal computer (PC) connected to a low-tech colour video camera.

Many existing gesture-recognition systems use sensor-based technologies. However, these techniques have many disadvantages. Data-gloves are used to measure the shape and position of the hands and such systems can recognise thousands of gestures. However, data-gloves are expensive and uncomfortable to wear. They are intrusive and



limit the natural motion of the hand. These gloves must be connected to the computer by wires or via wireless networks. Such restrictions mean they are difficult to operate, fragile, non-portable and not really an acceptable option for practical gesture recognition applications.

A more practical approach is to use computer vision techniques. This involves a user simply performing Sign Language in front of a camera. The captured images are processed and appropriate details extracted to translate the images to text/speech.

### **1.3 Computer Vision in Gesture Recognition**

Computer vision is an area of research that is currently receiving a lot of attention with worthy results. It has been successfully used in biometrics for face detection and fingerprint matching, in surveillance for human and behaviour detection, in pattern detection for medical imaging, in weather forecasting from satellite images, in intelligent robots, along with a vast amount of other areas. With the upsurge in computer vision many new techniques have been developed and have since been applied to gesture recognition and sign language translation.

Different researchers have explored different approaches and techniques to gesture recognition. Most of these techniques, however, contain a common global procedure.

(1) Identify features of the object in the images; A wide range of features have been utilised that try to help discriminate gestures while allowing gesture classification to be invariant to the local characteristics of the user performing the gesture.

(2) Classification of features into classes; Involving accurately sorting object features into their relevant category. Classification can involve many techniques, statistical or non-statistical, discrete or continuous, Nearest Neighbour or K-Nearest Neighbour along with many other techniques including hybrid techniques.

In order to compute the features of a hand-shape, the hand must first be identified in the image. The range of possible skin colours of the potential users is vast if we consider users from all races and ethnic origins. Some researchers request the user to wear coloured gloves in order to quickly identify the hand region. However, this practice is becoming increasingly unacceptable in the research community. A more satisfactory solution is to identify a predetermined skin colour range that represents skin regions. Alternatively motion cues have been used to locate moving hands, while boosting has been proposed to detect hand objects [18]. Hybrid techniques of these approaches can increase the accuracy of hand detection.

Stereo and multi-camera systems are increasing in popularity in current vision research. More than one camera gives the advantage of a three dimensional view of the scene. However, the increased complexity and computation involved in two or more cameras diminish the prospect of achieving real time recognition.

Thermo cameras and infrared cameras can also provide some advantages by eliminating the need for human segmentation and allow us to identify the relative distance of objects from the camera. These cameras can be expensive, compared to simple web cams, and may be inaccessible to the prospective audience of the gesture recognition system.

## 1.4 Approaches to Sign Language Recognition

While some researchers have concentrated on high level gesture analysis, such as arm waving and orchestra conducting we have based our research on sign language recognition with a view on HCI. Sign language recognition offers a vast array of problems to cope with, these include:

- A large vocabulary of allowable hand-shapes;
- Insignificant differences between hand-shapes;
- Variation of signs from different users;
- Variation in the speed the sign is performed;
- Signers different interpretation of the sign;
- Difference between novice and fluent users.

From discussions with fluent ISL signers it is evident that three major characteristics are required for accurate sign language recognition:

- The configuration of the hand (Hand-shape);
- The relative position of the hand in relation to other body parts;
- The hand motion or trajectory of the hand.

As described above accurate hand-shape is crucial for sign language recognition because different signs exist that contain the same motion and position information but only differ in the hand-shape. With this in mind we have based a significant part of our research on classifying the hand-shape.

## 1.5 Hand Gesture Recognition

Vision-based hand gesture recognition systems fall into two types, model-based and appearance-based. Model-based systems use 3D-models of the hands and arms that are compared with the incoming image in real-time. The parameters of the model are varied to find the best match with the incoming data. Tracking algorithms such as Kalman filters [37] or Condensation [38] are used to predict the next set of features in future frames. Some examples of techniques employed by model-based systems are tracking each finger separately, or tracking the contour of the hand. The problem with these systems is the hand is a highly deformable articulate object with up to 28 degrees of freedom. Modelling the hand involves high complexity and performing matching in real-time can be difficult and computationally expensive. The model also tends to lose track if the hand-shape changes sharply or becomes occluded.

As the name suggests, appearance-based systems classify the image based on the physical impression of the 2D image. Usually a large database of 2D images, or templates, is constructed containing a number of different hand poses. When a new image is inputted, the system searches through the database for the nearest matching template. If the database is large enough, and contains very many possible poses, a very high accuracy can be obtained. However, the larger the database, the longer it takes to search, which makes real-time implementation difficult. It is the problem of creating an efficient search algorithm that our work is designed to address.

To achieve accurate gesture recognition over a large vocabulary we need to extract information about the hand-shape. This usually involves detecting the hands, isolating them, and classifying them. In hand-shape recognition, transformation invariance is key

for successful recognition. We propose a system that is invariant to small scale, translation and shape variations. This is achieved by using a-priori knowledge to create a transformation subspace for each hand-shape. Transformation subspaces are created by performing Principal Component Analysis (PCA) on images produced using computer animation. A method to increase the efficiency of the system is outlined. This is achieved using a technique of grouping subspaces based on their origin and then organising them into a hierarchical decision tree. We compare the accuracy of this technique with that of the Tangent Distance technique and display the results.

We introduce a technique that enables us to train this appearance-based method using computer animation images and test using images of real human hands. Also presented is the incorporation of this hand-shape classifier into a dynamic gesture recognition system.

## **1.6 Outline of the Thesis**

The remainder of this thesis is divided into five main parts. Chapter 2 gives a comprehensive literature review of current and previous research in the area of gesture recognition. An introduction to the animation software used is also provided. Chapter 3 introduces our subspace classifier. Here we provided some experiments using animation hand images. In Chapter 4 this technique is extended to allow for classification of images of real hands. Our dynamic gesture recognition system is then outlined in Chapter 5. Once again we offer a broad set of experiments to evaluate the proposed technique. Finally some Conclusions and Future work are described in Chapter 6.

## *CHAPTER 2*

### *LITERATURE REVIEW*

#### **2.1 Introduction**

In recent years hand gesture recognition has become a popular research topic. Many novel and interesting applications of hand gesture recognition have been introduced in recent times. These include, music synthesis [14], television control [26], robot control and as a surgeon's aid [27]. Gesture analysis has been used to identify motion patterns in human joints in order to produce life-like animation and graphics. Gesture Analysis has also been used to identify swimming style [28], dance posture recognition [29] and gait recognition [30].

As stated we are interested in gesture for HCI. We will now try to summarise the techniques used in existing HCI gesture recognitions systems. While some of the systems described below clearly outperform others it is difficult to rate individual systems because their virtues may be focused towards a particular task. Sign language recognition requires user independence over a large vocabulary, while recognition accuracy is important, 100% accuracy is not essential. However, some HCI systems may require a total 100% recognition rate over a smaller vocabulary. Some important entities that could be used to evaluate a system are:

- Recognition Accuracy;
- Size and range of the vocabulary;
- Variation of the gesture being tested;

- Number of users involved in testing;
- Environmental conditions and prerequisites of the image scene;
- Use of coloured gloves;
- Computation time.

## **2.2 Static Hand Gesture Recognition**

Many solutions to the problem of the static gesture recognition problem have evolved from pattern recognition techniques. These involve gathering some set of features that can robustly distinguish an individual hand gesture from all other gesture classes. While having discriminating features is important, these features also have to be invariant to local hand characteristic such as hand-shape, hand size, skin colour, illumination and user interpretation of the gesture. Some of these user-dependent characteristics can be removed using pre-processing techniques depending on the nature of the given system. Another important aspect of feature selection is coping with background clutter, this is particularly important if the hand needs to be segmented from the image.

### **2.2.1 Nearest neighbour and Cluster analysis**

Nearest neighbour is an uncomplicated way to classify images of hand-shapes. Test images are simply compared to a trained database of images and classified by finding their nearest neighbour. A number of techniques can be used to calculate the distance to the nearest neighbour, these include Euclidean Distance, Mahalanobis distance and Bayes' Theorem.

Due to the aforementioned variances in static hand gestures, this database needs to be quite large in order to provide accurate recognition. This technique quickly becomes unfeasible in a reasonable amount of time. Cluster Analysis can be used to speed up this process. The training data will contain samples of images that are quite similar and belonging to the one class. These images can be clustered and represented by one single value, possibly the centre of gravity of the image cluster assuming the cluster approximately adopts a spherical shape.

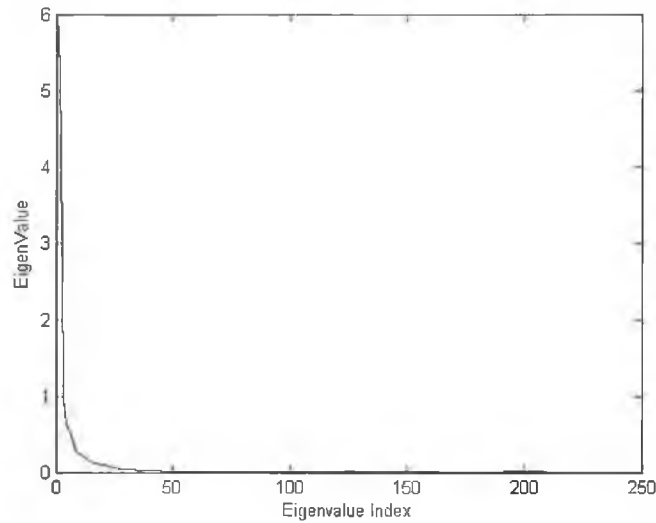
### **2.2.2 Template Matching and PCA**

In order to speed up this style of recognition, Principal Component Analysis (PCA) is often used to reduce the dimensionality of the data. Even when working with relatively small images of 32x32, this equates to 1024 pixels. Working with a 1024 dimensional space requires intensive computational power. Usually the discriminative features of the images will lie on a low dimensional subspace because of the correlation between the features. PCA is a statistical tool that allows us to reduce the dimensionality of data that contains many interrelated variables. It works by projecting the high dimensional data into a lower dimensional subspace while retaining the features that contain most of the variation present in the original data.

PCA is achieved by first finding the covariance matrix of the set of images. The eigenvectors of the covariance Matrix then form the new feature space known as the eigenspace. This eigenspace contains the same number of dimensions as the original feature space, in our case 1024. To reduce the dimensionality a subset of eigenvectors are selected and retained. Usually only a small percentage of eigenvectors are required to represent the variation in the data. By arranging the eigenvalues in decreasing order



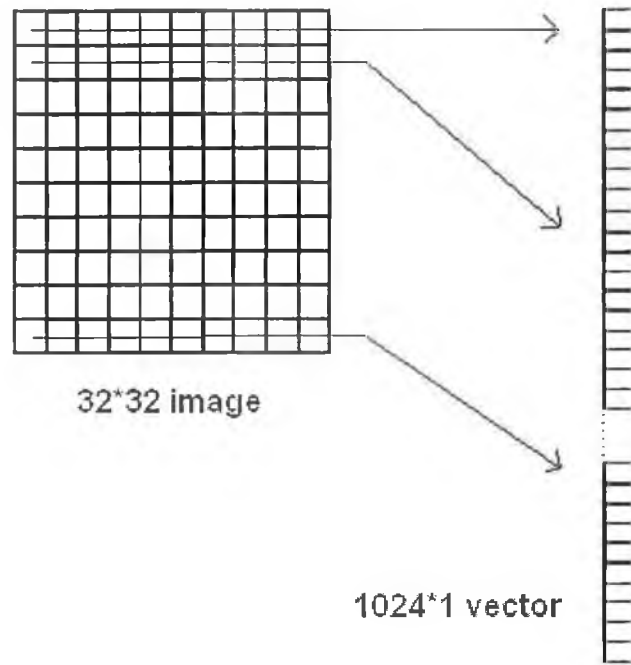
we can select any number of the corresponding eigenvectors in the direction of the greatest variation. Figure 2.1 shows that the majority of data can be preserved by retaining the first 20 eigenvectors. Beyond this point the eigenvalues are close to zero meaning the eigenvectors are virtually ineffective. The number of eigenvectors can be fixed, but is usually variable so that a certain proportion of the information is retained



**Figure 2.1** Plot of eigenvalues for the ordered set of eigenvectors produced from a set of 239 images of the ISL finger spelling hand-shape for A. Note for visualisation purposes only the first 200 of a total of 1024 eigenvalues are displayed

Test images can then be projected into the subspace by multiplying them by the set of retained eigenvectors of the subspace. A complete mathematical description of PCA can be found in Appendix B.

Conventional PCA is performed on vectors. Conversely our images take the format of a 2 dimensional matrix size 32x32. These images need to be reshaped before PCA can occur. This can be achieved by concatenating the rows of the matrix into a 1024x1 vector as shown in **Figure 2.2**.



**Figure 2.2** Reshaping a matrix to a vector

Shamaie [21] introduced a PCA based approach for static hand-shape recognition. In this work Vector Quantisation is performed on the reduced dimensional data to produce a codeword for each hand-shape. Test images are classified by projecting them into the subspace and finding the nearest codeword.

Another method contributed by Wu [1] contains a multi-scale hierarchical tree search using the PC space. He uses a Gaussian kernel to blur the images in order to reduce their differences. PCA is performed on these images to reduce their dimensionality. The subspace of images is then divided into several clusters using the k-means algorithm. The level of blurring is reduced and PCA is performed on each of the individual clusters. The procedure is recursively called to each of the clusters until a stopping condition is reached. Test images are then categorized by traversing the tree, choosing each path by projecting images into each of the subspaces, and finding the

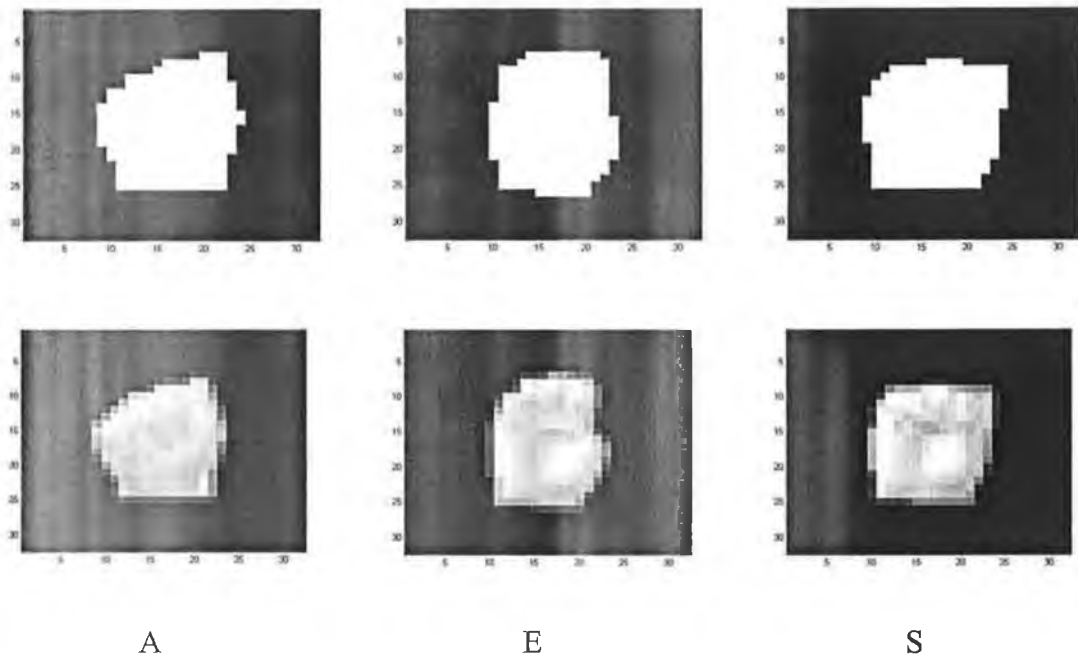
nearest using the perpendicular distance of the image to the eigenvectors. In this work he uses colour gloves to achieve accurate segmentation. An average recognition rate of 98% is achieved for the 23 static ISL finger spelling hand-shapes.

Wu et al. [21] elaborated the simple PCA approach by using Multiple Discriminant Analysis (MDA) to select the most discriminating features. In order to construct such a system they need a large labelled data set. To solve this problem they use an Expectation-Maximization (EM) technique to help automatically label the data set once it has been bootstrapped with some examples. Using MDA allows them to retain features that give disparity between classes while discarding features not required for classification. They articulate the difference between the dimensions of the data retained by PCA and the dimensions of the data after MDA is articulated. In their research they have shown that these mathematical features outperform physical features such as boundary information and texture features. They present a system that achieves a 92.4% recognition rate for fourteen defined hand-shapes from a range of viewpoints. These hand-shapes were chosen to maximise disparity between classes. A hand localization system is used to detect hand regions, while skin segmentation is used to remove background noise.

Overall PCA can be utilized to obtain quick classification. However, due to the nature of the algorithm it depends greatly on the appearance of the object being classified. It is necessary to isolate and segment objects from the background. Also any system based on PCA needs to take into consideration the fact that PCA is particularly susceptible to, translations, rotation, scale, illumination and skin colour.

### 2.2.3 Contour and Silhouette.

Though contour-based representations use invariant features, they may generally suffer from ambiguities resulting from different hand-shapes with similar contours [16]. Some contour images along with their greyscale images are shown in **Figure 2.3**. Here three different hand-shapes from ISL, A, E and S, are illustrated. From these images it is difficult to distinguish what hand-shape is present especially when compared to the greyscale images.



**Figure 2.3** Some sample ISL finger spelling images along with their silhouette images.

Some researchers have utilised hand contour and silhouette for hand-shape recognition. Typically these systems are limited to a small vocabulary of distinguishable hand-shapes. However, frequently these techniques are able to incorporate some rotational invariance which is an attractive benefit. In all of the following described techniques the hand needs to be identified and segmented from the image background.

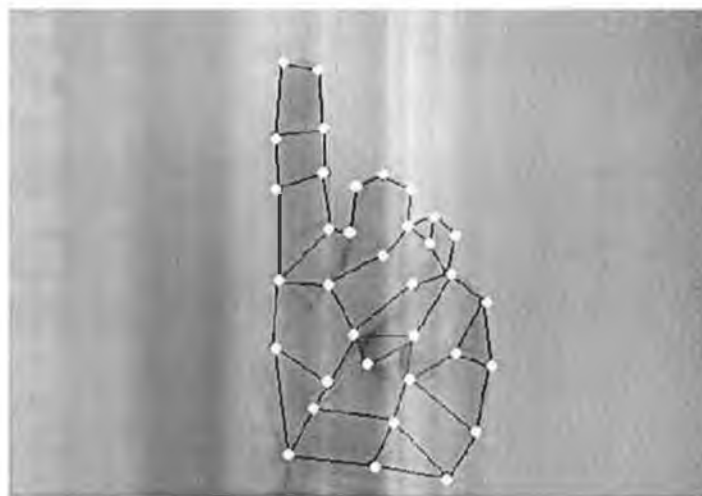
Chen et al. [4] present a method of classifying the static hand poses by using the fourier descriptor to characterise the spatial features of the hands boundary. Skin colour detection and motion information are used to segment the hand from the background. The advantage of this technique is its invariance to the following transformations: scale, skin colour, translation and 2D rotations in the yaw direction.

Carreira et al. [14] outline a procedure for static hand-shape recognition by performing some normalisation on the polar coordinates of the hand-shapes contour. Hands are detected and isolated using a combination of Harr-like features object detection along with skin colour segmentation. They can achieve reasonably high recognition rates in real time using a relatively small vocabulary of 7 gestures. Once again this vocabulary has been chosen to maximise disparity between individual classes. The system presents invariance to scale, translation, symmetry, and 2D rotations in the yaw direction.

Yuan et al. [15] developed their system by determining a new Active Shape Model (ASM) kernel based on shape contours. Classification is improved by incorporating Support Vector Machines (SVM) with the ASM Kernel that they claim allows them to have significant variability between individual hand poses. Once again invariance is achieved for scale, translation and 2D rotations in the yaw direction. They display their superior performance when compared to a simple template matching technique. Using a vocabulary of 6 gestures they present a recognition rate of 95.7% compared to 78.9% achieved by template matching for the same data set.

### 2.2.4 Elastic Graph Matching

A method to classify hand postures against complex cluttered background was proposed by Triesch & von der Malsburg [17] using elastic graph matching. This technique involves overlaying a graph over the relevant object in the 2D image. The nodes of the graph fall along the boundary and on highly textured positions within the hand. The graph is then compared to a trained graph for each hand pose. An example of how an elastic graph is used to represent a hand image is shown in **Figure 2.4**. Training is initialised manually and then fine-tuned using a semi-automated process. Advantages of this approach are that it is invariant to scale, translation, cluttered background, skin colour, and illumination. Test images have been cropped to the area containing the hand image. Classification is achieved by finding the graph that best fits the image. No consideration is taken for test images that contain no hand objects. Tests were performed on a vocabulary of 10 hand postures and contain samples on both complex and uniform backgrounds and using 24 different people. A rotation restriction of 20 degrees is placed on hand postures. The average recognition rate attained was 91%. However, this technique requires high computationally complexity taking several seconds to process each image.



**Figure 2.4** Hand postures represented by labelled graphs

### **2.2.5 Texture Based Pattern Recognition Techniques**

Using texture-based features is another common practice for static gesture recognition. Ong et al. [18] proposed a system to detect and classify hand-shapes by creating a strong classifier consisting of a number of weak classifiers. The weak classifiers used are based on Haar wavelet-like features [19]. The weak classifiers to be combined to form the strong classifiers are then learned through the boosting technique. Such a system has the advantage of not requiring segmentation and can cope with cluttered background and variation in skin colour. Nevertheless it is limited to a small number of hand-shapes at constrained postures.

Recently using local spatial texture information has become popular in face detection using the Modified Census Transform (MCT) and Local Binary Patterns (LBP). The methodologies of these two practices are quite similar and only differ in the way spatial texture information is ordered. Just et al. [20] introduced a hand-shape system based on the MCT. Boosting is used on these MCT features to train a strong classifier. The main benefit of the MCT is that it is invariant to illumination. However, the classification results are modest, especially when complex backgrounds are present. With a vocabulary of 10 hand-shapes a recognition rate of 92.79% is achieved on images with a uniform background, while 81.25% is achieved when the background is cluttered. In the data set all images have been cropped to contain only the hand object and contain only small evident rotational variance. In each image the hand is perfectly centred and all images are the same size.

Another texture-based feature that has been used is wavelet filters. Wu et al. [21] presented a technique that combined Gabor Wavelet filters for texture information with Fourier descriptor for shape information along with that of other physical features such as hand area and contour length. Similarly to their technique described in 2.2.2 where mathematical features were employed, EM was used to semi-automatically produce a large labelled training set. MDA was then used to select the discriminative features. This technique proved successful for a vocabulary of fourteen hand-shapes at many orientations and realised a recognition rate of 90.8%. However, it failed to match the accuracy of their other previously described method based on PCA/MDA which achieved 92.4%.

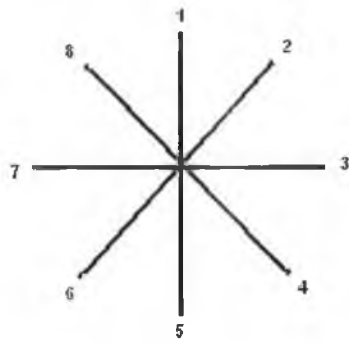
### **2.3 Dynamic Gesture Recognition**

Gupta et al. [12] present a method of performing gesture recognition by tracking the sequence of contours of the hand using localised contour sequences. Their algorithm requires the hand to be segmented from the background and is done using a histogram threshold on greyscale images. This approach achieved high classification accuracy for a vocabulary of 10 gestures taken from American Sign Language (ASL). However, it does not run in real time using conventional hardware. Using contours alone will limit the scalability of this technique.

Wu [1] developed a dynamic gesture recognition system using Discrete Hidden Markov Models (DHMMs). DHMMs were trained on a sequence of tuples that represent each gesture. A tuple consisted of two integers that symbolised the shape of the hand and a directional parameter. The shape of the hand was classified using a PCA multi-scale hierarchical tree search as described in **Section 2.2.2**. Here a relatively straightforward



approach was used to classifying the movement of the hand, by noting the local direction the hand has moved. This direction code is shown in **Figure 2.5** and is calculated by finding the direction of hand object **I** in relation to hand object **I**<sup>-1</sup>. A DHMM was trained for each gesture using 60 example recorded continuously. Likewise each gesture was tested with 60 examples that we not used at any time during training. A recognition rate of 92.88% was achieved for a vocabulary of 35 dynamic gestures taken from ISL under controlled environments, using coloured gloves to accurately distinguish and segment hands.



**Figure 2.5** A simple directional code used in dynamic gesture recognition.

Shamaie [21] outlined and compared two frameworks for recognising dynamic gestures. Both a graph matching technique and HMM technique were tested and compared for speed and accuracy. First a global PCA is performed on a data set of training images. As a gesture is performed the sequence of hand-shapes can be considered as creating a graph in the subspace. The graph matching technique is trained by learning the trajectory of a gesture in the PC space using many different samples and producing a representative graph which is a series of nodes and vertices. Likewise a test gesture is projected into the PC space and the resulting graph is compared to the training graphs for each gesture to find the nearest. The HMM based technique involves dividing the PC Space into a number of codewords. Hand-shapes are projected in the PC Space and

the relating codeword is found. The sequence of hand-shapes produced from a gesture form a sequence of codewords which are used to train a HMM for each gesture. A test gesture is projected in the same PC space and the sequence of codewords produced is passed into each of the HMMs to find the best match. Both techniques were tested and trained using the same data set consisting of 100 gestures. A selection of 5 samples were used for training while a different 5 were used for testing. The HMM based techniques proved superior with a recognition rate of 95.4% with the graph matching algorithm reaching 95% accuracy. However, the HMM based approach proved to be more computationally expensive and was 6 times slower than the graph matching algorithm.

Patwardhan et al. [13] recently introduced a system based on a predictive eigentracker to track the changing appearance of a moving hand. The initial eigenspace is constructed from a limited number of samples. Hand images are isolated using skin segmentation and motions cues. Both affine coefficients and eigenspace coefficients are considered when classifying test images. This introduces some rotation, scale and shear invariance. The offline trained eigenspace is updated on the fly with an efficient on-line eigenspace update mechanism used to refine the eigenspace. The eigentracker is used to give information on both the appearance and motion. Gestures are considered as vectors of shape and trajectory coefficients and are compared using the Mahalanobis distance. Particular attention is taken to ensure that a gesture vocabulary is formulated by choosing gestures that are well separated in gesture-space. The system is trained with 64 gestures, 8 occurrences of 8 different gestures. Testing is then performed with these 64 gestures used for training along with a further 16 unseen gestures, 2 occurrences of each of the 8 different gestures. The gestures are captured under reasonably controlled

conditions. This system gives 100% accuracy for this vocabulary of 8 gestures using this data set.

Kadir et al. [25] describe a technique to recognise sign language gestures from British Sign Language (BSL). They use a set of four discrete features to describe each of the following:

- Position of the hands relative to each other;
- Position of the hands relative to other body locations;
- Movement of the hand;
- Shape of the hand.

Head and hand detection is achieved by using a boosting technique where a strong classifier is constructed from a number of weak Haar-like classifiers. This approach is extended to perform hand-shape classification where a strong classifier is learned for each of the 12 allowable hand-shapes. The relative position of the hand can be easily determined once the hand has been detected. Body locations are estimated using the head position to give a relative location of other body parts on the contour of the person. A total of 10 different motion patterns are used to classify the motion of the hands, they are calculated by examining the hand positions in subsequent frames throughout the gesture. A Markov chain is used to represent gestures, which are considered as a sequence of states based on the aforementioned features. They have achieved an average recognition accuracy of 89% for a vocabulary of 164 gestures for a single user system under a controlled environment. In these videos the signer wears colour gloves. The person remains as still as possible while the hands and arms are the only moving objects in the frame.

## 2.4 Using Computer Animation and Poser

With the recent surge in the realistic quality of computer animation we decided to investigate the notion of acquiring synthetic images of hands and hand gestures. In particular we found the Poser animation package to be particular appealing. Poser is a rendering and animation software program used to model the human figure in three dimensional format. As the name suggests Poser is designed to accurately copy and imitate the postures of both humans and animals. It offers an articulated model that accurately represents the degrees of freedom of the human body.

It comes with a few sample human figures of both sexes and of different ages. It provides functions to move and reposition each joint of the figures. Limits can be set on each of these joints to ensure that no un-human joint positions are reached. A large library of poses are offered and can be utilised on any of the supplied figures. Poser also provides the functionality to allow Python, a scripting language, to interact with the on screen models. Functions are offered that allow all joints to be manipulated in their allowable degrees of freedom. This means we can write batch files that manipulate figures in order to quickly produce high quality images that can be used to train and test our gesture recognition techniques.

The advantage of using Poser is that, given a theoretical notion, we can instantly create a tailored database of suitable images in order to test the correctness of the given hypothesis. This saves time and effort in manually recording and labeling sequences of images produced from a camera, and can reduce the toil and expense involved in recruiting actors to perform gestures in a number of different circumstances, especially when testing an adhoc notion. While we can introduce unsystematic variations into the

joint movements, one particular advantage of using Poser was in generating images at precise angles to test the accuracy of certain systems. This is illustrated in **Chapter 3** when testing the bounds of our subspace technique.

A full list of the advantages of using Poser to create gesture based images is described below:

- Manipulate all the joints of the body, including the fingers and hand;
- Control the angles of each joint;
- Change the skin colour;
- Modify the lighting conditions;
- Control the size of the hand and fingers;
- Set the orientation of the figure;
- Specify the distance of the figure from the camera;
- Move both the camera and the figure;
- Use male/female figures of different ages;
- Change the figures clothing;
- Modify the figures hair and facial expressions;
- Set a uniform background in order to speed up segmentation of training images.

**Figure 2.6** shows some examples of Poser images. These contain different actors of different sexes and different ages, under varying light conditions with different camera angles, along with a fluctuating distance from the camera. A more detailed description of how Poser was used in this research is outlined in future chapters.



**Figure 2.6.** Some sample figure images produced from Poser

## **2.7 Summary**

In this chapter we outline the technologies currently used in both static and dynamic gesture recognition. An exploration of model and appearance based features is provided for static gesture recognition while an in-depth examination of dynamic gesture recognition is offered. In addition an introduction to the merits and the potential use of Poser software is described.

# *CHAPTER 3*

## *STATIC GESTURE RECOGNITION – SUBSPACE APPROACH*

### **3.1 Introduction**

Many of the approaches to hand-shape recognition described in the literature review display sub-optimal results due to the highly deformable nature of the hand. Usually a compromise is obtained between small vocabulary and accurate recognition. Any hand-shape recognition system needs to be able to cope with slight distortions along the 28 degrees of freedom of the hand.

Rotation, translation scale and colour are the four most significant transformations that our invariant system needs to tolerate. In this chapter we concentrate on rotation and translation. Scale and colour are described in more detail in **Chapter 4**. Clearly any hand-shape recognition system needs to be able to identify similar hand-shapes across different rotations. It is inconceivable to require the user to perform the hand-shape in exactly the same orientation each time. With these issues in mind one important question is how to align these object images. A commonly used approach is to align hand objects based on the centroid of the objects bounding box. However, with the aforementioned variances of the hand in mind, combined with slight hand-shape mutations from occurrence to occurrence and from user to user, this centroid is

inconsistent. Therefore the system should also be capable of handling translated object images.

Another significant problem of hand-shape recognition is accurate segmentation of the hand from the images. We have found that accurate segmentation is extremely difficult due to shadows in the image, background noise, motion blur. With this in mind we should try to ensure that our classification system is able to manage incomplete segmentation. However, in some cases, if the segmented image of the hand is poor, naturally a drop in accuracy is expected.

We have found that our proposed method of using an invariant subspace approach offers reasonable recognition over a reasonably large vocabulary.

## **3.2 Hand-shape Transformations**

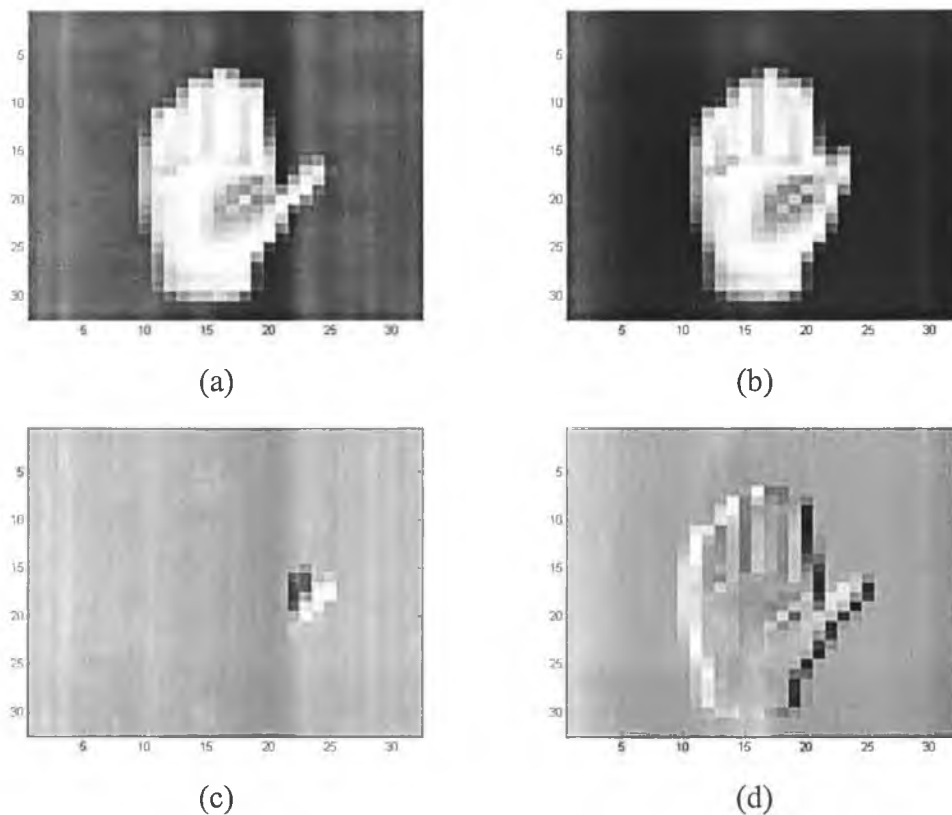
A brief description of the transformations confronting a hand-shape recognition system were described above. We will now give a more concise explanation and illustrate how these transformations affect accuracy.

### **3.2.1 Translation Transformations**

The need for a hand-shape recognition system to be invariant to translation is illustrated in **Figure 3.1**. Here we see two images, (a) and (b), that represent two different occurrences of the same hand-shape. However, the thumb in (a) is more outstretched than that of (b). This problem of user interpretation of hand-shape is common in Sign language and gesture recognition. If correctly aligned the actual difference between the two images is shown in (c). Note that two images were created in Poser so we can find



this correct alignment quite easily. However, in practice it is difficult and time consuming to find the exact and correct alignment of articulate objects. A commonly used approach is to align objects by the centres of their bounding boxes. However, this technique is susceptible to small variations as shown in (d). Here the actual distance between (a) and (b) is shown once they have been aligned using the bounding box technique. This difference is significantly larger than (c) and such an amount of noise can cause misclassification.



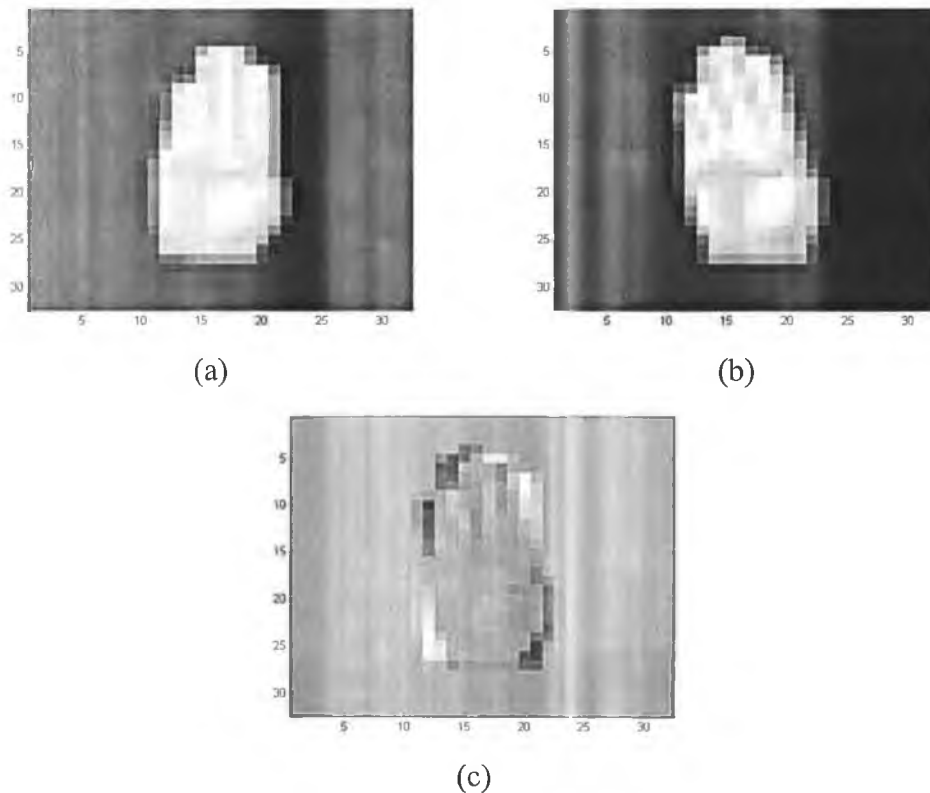
**Figure 3.1.** The alignment problem in hand-shape recognition.

### 3.2.2 Rotation Transformations

Another transformation needing consideration is that of rotation. **Figure 3.2** demonstrates how any hand-shape classification system needs to cope with rotation

transformations. (a) and (b) show two different examples of one particular hand-shape. They only differ in the rotation at which they are performed. Simple differences in rotation can occur due to user interpretation of the hand-shape, along with having the user at a differing angles to the camera. If (a) and (b) are aligned using the bounding box technique and we compare the resulting images in a pixel by pixel manner, a large variation is observed as in (c). Such a large variation in equivalent hand-shape at altered rotations is an important consideration in hand-shape recognition.

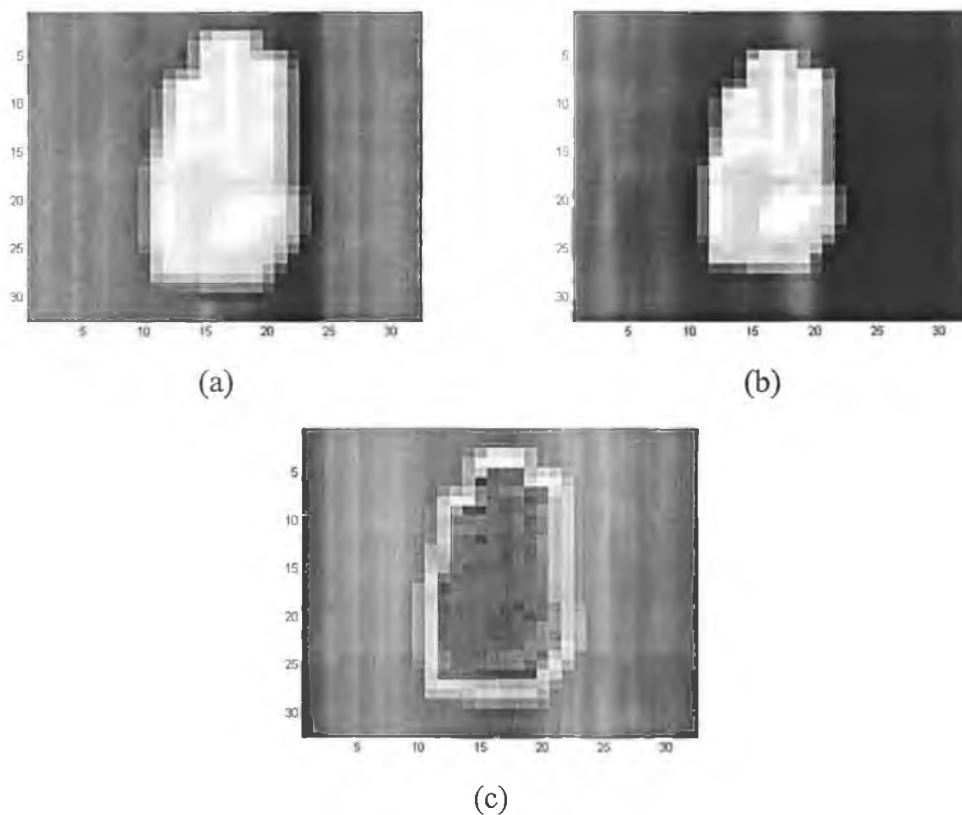
It is worth considering that in this case we have only contemplated rotations in the yaw direction, similar variations are also evident in the pitch and roll directions. The problem is further deteriorated when a combination of the three rotation directions are present.



**Figure 3.2.** Rotation invariance problem in hand-shape recognition

### 3.2.3 Scale Transformations

A similar predicament is apparent when dealing with scale transformations and is represented in **Figure 3.3**. Different scales of comparable hand-shapes occur when the users are at various distances from the camera or when different users have inconsistent hand sizes. An example of two identical hand-shapes that only differ in their scale is shown in (a) and (b). Once again a pixel by pixel comparison shows a vast discrepancy that will induce classification error.

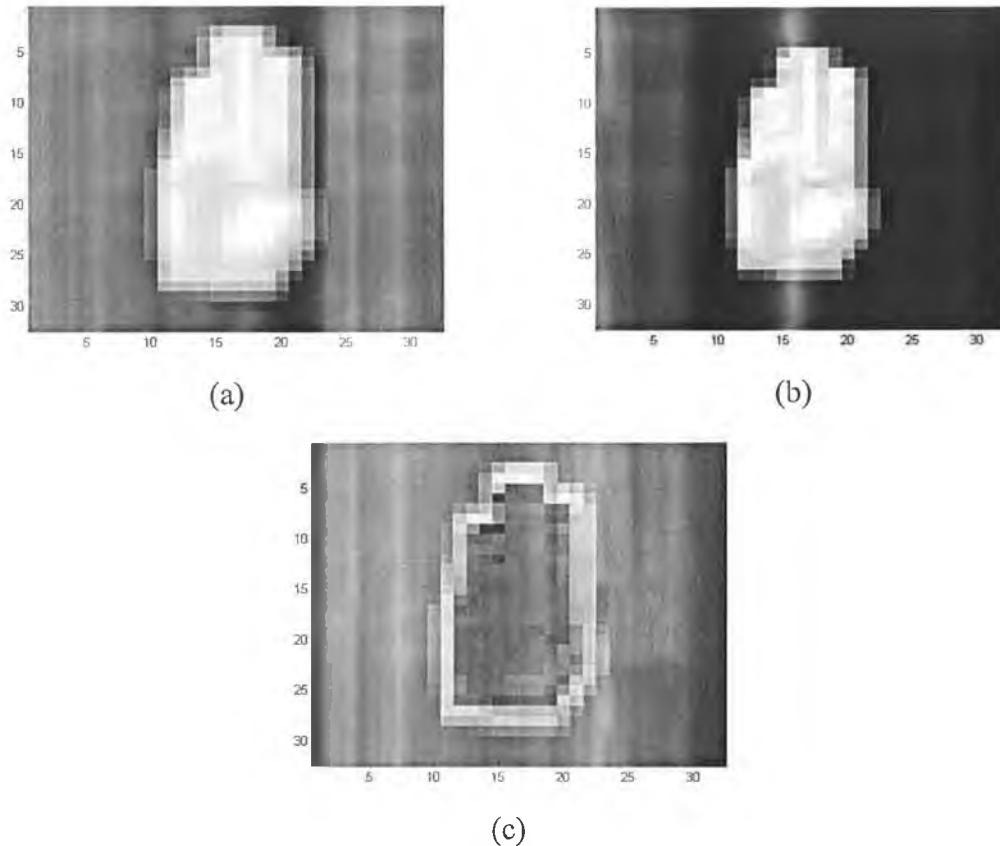


**Figure 3.3.** Scale invariance problem in hand-shape recognition

### 3.2.4 Colour Transformations

Similarly colour invariance offers a comparable recognition problem. Once again we display two hand-shapes, in **Figure 3.4**, that are identical in structure and orientation, but differ in their colour. The considerable variation between these images is portrayed

in (c). In reality dissimilarity in colour is a regular phenomenon due the fact that the range of humans skin colour is relatively large. Irregular lighting conditions can also alter the colour of hands. Eliminating this noise introduced by hand colour is another issue that needs to be solved in order to achieve accurate hand-shape classification.



**Figure 3.4.** Scale invariance problem in hand-shape recognition

### 3.3 Transformation Invariance

We have outlined some of the individual image transformations that we wish our system to be invariant to. However, the problem becomes much more complex when more than one transformation is in existence. This in turn makes classification much more difficult.

Simard et al.. [9] proposed the tangent distance technique to achieve transformation invariance. This method approximates the high dimensional transformation hyper-plane with its tangent plane. Now two images can be compared by finding the distance between their transformation tangent planes. Later we compare results from our subspace method with that of the Tangent Distance technique.

We propose a transformation subspace technique to combat these issues. Multiple subspace approaches have been employed previously by Wu [1], where the training data was produced in an ad-hoc manner and sectioned into subspaces. This method proposed an exhaustive search along subspaces for a given test image.

Zhao [2] used an approach to calculate transformation subspaces from original subspaces created from ‘perfect’ training images for face recognition. They offer a multi-resolution search to speed up the exhaustive search of the test image to the original subspaces, along with each transformation subspace. It is important to note that this method only allows for 2D image transformations.

Our proposed method creates the invariant subspace from a sampled subset of all possible transformation images. These images are produced systematically using the commercially available Poser modelling software [10] and includes 3D hand transformation. Using this technique to obtain the training data means we can produce a large, and complete, labelled training set. Performing PCA on the set of images for each hand-shape will generate a subspace that accurately represents the complex transformation hyper-plane of the given hand-shape. Instead of performing an exhaustive search on each subspace, we propose a hierarchical tree search that groups

similar eigenspaces together. This is achieved by performing a fuzzy k-means algorithm on the origins of the eigenspaces. This allows us to reduce the search time involved while retaining accurate search results.

### **3.4 Creating Training Database**

One crucial factor in this system is how competently the training images can be produced to accurately represent the transformation hyper-plane. Indeed with all appearance based recognition systems the problem of producing a large labeled training set needs to be considered. We have found that by using computer animation we can accurately create a model of the hand in any orientation. It is also possible to change all of the physical characteristics of the hand, for example hand size, direction, orientation and skin colour along with lighting and scene conditions. It is also possible to modify the distance of the user from the camera, the angle of the camera and the background.

Once the original pose for each hand-shape is manually initialised, all subsequent transformations can be generated automatically by manipulating the hand model using the Python scripting tool provided by Poser.

The origin pose is defined as the perfect instance of a static hand gesture. This involves having the wrist, palm and each finger at the correct position and the hand at the correct orientation. An example of an origin pose for the letter A is shown in **Figure 3.5**.



**Figure 3.5.** Origin pose for the letter A

Capturing an image of the hand model at each stance establishes the complete set of hand-shape transformations needed to construct its transformation subspace. Using this method of a-priori knowledge to construct the subspaces means we can eliminate the process of automatic subspace segmentation as proposed by [7, 8]. This involves complex and usually time-consuming calculations, used to accurately identify images that should be contained in the same subspace, in order to achieve accurate recognition. It also allows us to dismiss the need for managing outliers or missing data in our subspaces [8]. This means we can create more accurate transformation subspaces than was previously possible using the simple PCA method.

We now introduce a system that has been trained and tested with images created using computer animation. The purpose of this system is to investigate how accurate this subspace system can deal with translation, rotation and small random hand configuration transformations. This system is tested with images created from Poser images to ensure the accuracy is impartial with respect to environment and user dependent issues. Testing with images of real hands is dealt with in **Chapter 4**.

### 3.5 Subspace System Overview

In order to create a given subspace, PCA is performed on the set of processed images for each hand-shape. Processing involves segmenting hand objects from each image and scaling them to 32x32. Segmentation is quite a simple process because we can control the environment of the Poser images. In practice we set the background to a particular colour. This colour can vary slightly when lighting is introduced. Segmentation now simply involves identifying pixels of the image that do not lie in the colour range of the background. Morphological operators are used to fill holes and smooth the edge of the object.

Similarly scaling is straightforward because the animated user is at a constant distance from the camera. Scaling is achieved by resizing the pixels contained within the objects bounding box so that the object pixels occupy a certain portion of the total image pixels. Considering the fact that subspaces are created using different translations of the origin image, we can use the centre of the bounding box to give a rough alignment of the images.

Once all pre-processing is complete we then perform PCA on a set of images for each hand-shape. Performing PCA provides  $M$  orthogonal eigenvectors  $\{u_1, \dots, u_M\}$  of the covariance matrix, that correspond to the first  $M$  largest eigenvalues, in order to maintain a minimum energy of the dataset. In our experiments we have found that retaining 95% of energy is sufficient to accurately differentiate hand-shape subspaces. Similar experiments as to how this quantity is calculated are detailed in **Section 4.3**. It is important to note that this retention of 95% of the energy only applies when testing and training with images that were obtained under these controlled Poser circumstances.

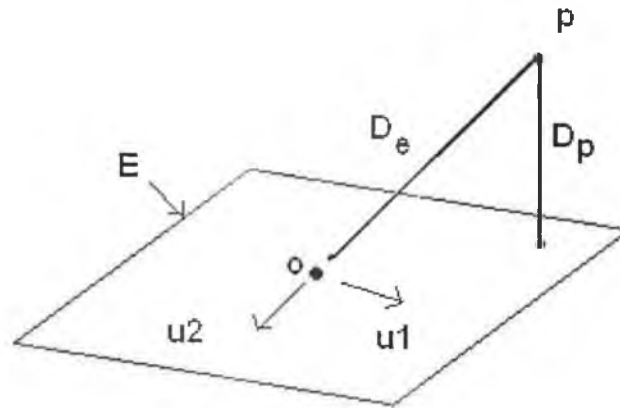


A different value is used in the ensuing chapter when testing with images of real hands. Nevertheless preserving 95% of the energy means we can accurately represent an eigenspace with 10-15 eigenvectors.

In order to classify test images a distance metric needs to be introduced. We project the test image into the subspace and find the perpendicular distance of the projected point to the eigenvectors representing the subspace. The perpendicular distance ( $D_p$ ) of a point  $\mathbf{p}$  to a given eigenspace  $\mathbf{E}$  is illustrated diagrammatically in **Figure 3.6** and is mathematically calculated using **Equation 3.1**.

$$D_p^2 = D_e^2 - \sum_{i=1}^M [(\underline{p} - \underline{o}) \bullet \underline{u}_i]^2 \quad (3.1)$$

Where  $D_e$  = Euclidean distance between  $\mathbf{p}$  and the origin  $\mathbf{o}$  of  $\mathbf{E}$ .



**Figure 3.6.** Perpendicular Distance of a Point to an Eigenspace

We now have the backbone for a simple finger spelling recognition system. American Sign Language contains 24 static finger spelling gestures. A sample system can be developed as shown in **Figure 3.7**. It is constructed as follows:

**Training** - Generate a transformation subspace for each hand-shape.

**Testing** – Project the test image into each of the subspaces to find the subspace with the nearest perpendicular distance. This subspace will be representative of one particular hand-shape.

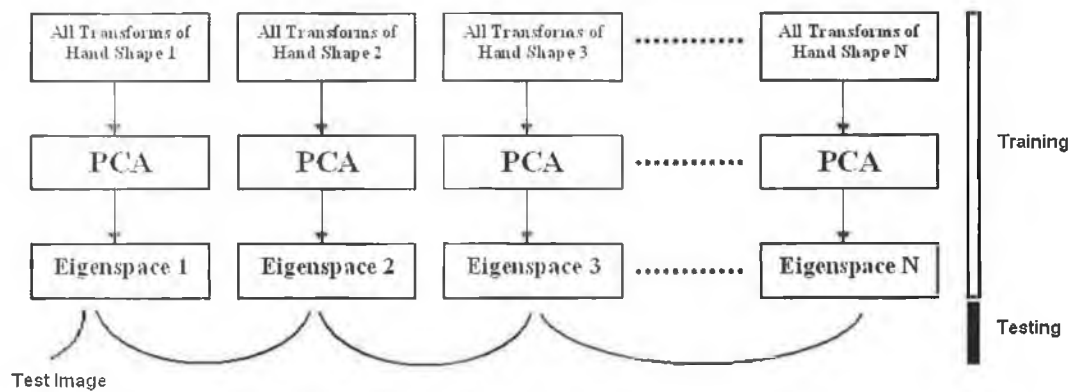
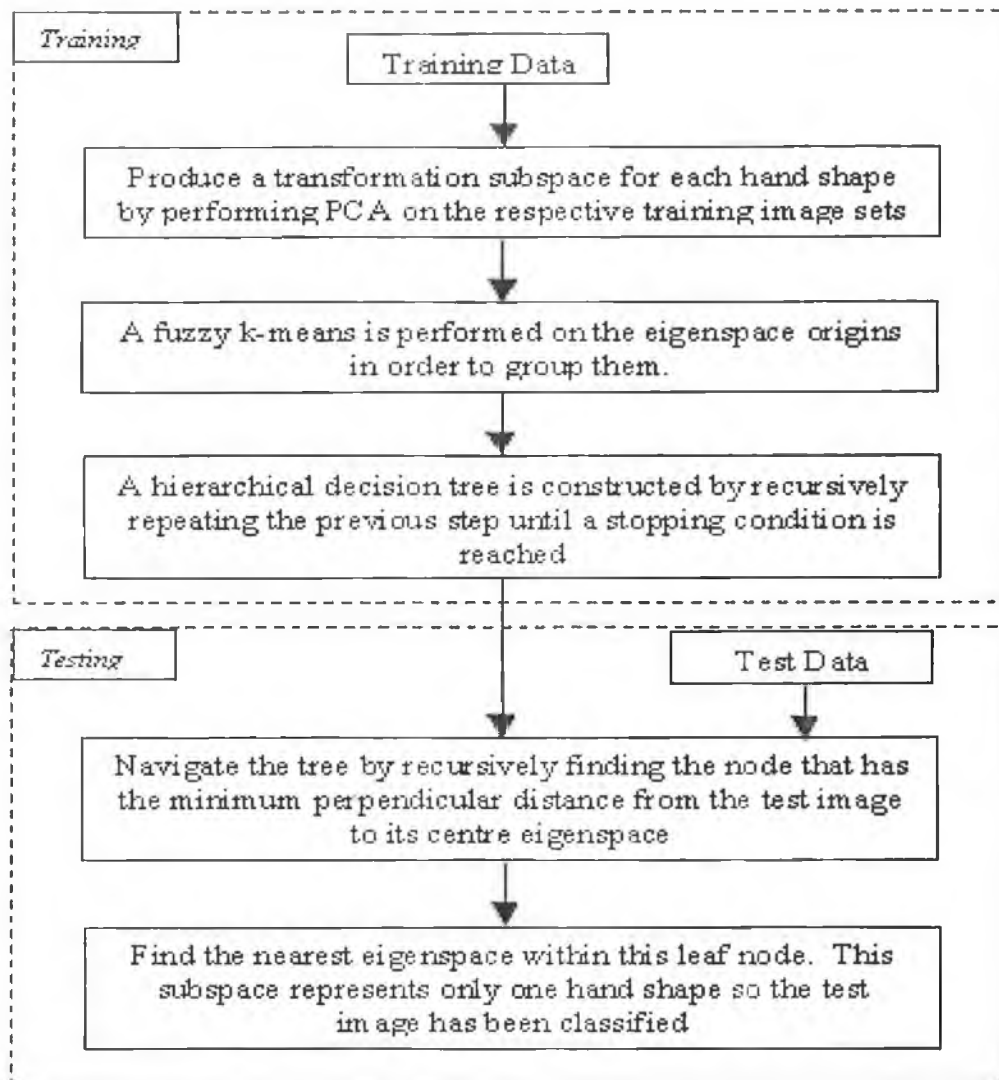


Figure 3.7 Simple Finger Spelling Recognition System Overview

### 3.6 Reducing Search Time

Initial experiments have shown that it is possible to reduce the search time from that of an exhaustive search. This can be achieved by organising the PCA reduced subspaces into a hierarchical decision tree. The decision tree is constructed using a fuzzy k-means algorithm that divides the dataset into two groups. This process is recursively executed until the stopping condition of the data in a node reaching a variance threshold is reached.

We have found that the origins of the eigenspaces are appropriate constituents to estimate comparability of eigenspaces. Consequently k-means is performed on the origins in order to group eigenspaces and create the search tree.



**Figure 3.8.** Reducing Search Time Overview

In the resulting binary tree, at each stage the test image will have the option of following either path. The test image chooses the path by which it has a smaller perpendicular distance to the centre eigenspace of that node. The centre eigenspace of a

node is estimated by finding both the average origin and average eigenvectors of the eigenspaces contained in that node. The approximation technique was chosen to speed up the training phase. We have found this approximation of the centre image is sufficient for the test image to successfully navigate the tree. Some experimental results are shown in the next section.

Once a leaf node is obtained, the nearest subspace in that node to the test image can be found by exhaustively searching through the reduced number of subspaces. This technique is summarised in **Figure 3.8**. While this tree search can reduce the search time involved in classifying each test image, some accuracy is compromised. This tree search should therefore be used then when high speed is necessary and precise accuracy is not essential. In the case when accuracy is the dominant requirement of the system, the exhaustive subspace search should be implemented.

### **3.7 Experiments**

We have performed a series of tests to assess the transformation subspace technique. We compare it with the Tangent Distance with relation to speed and accuracy. Code used for Tangent Distance is as found at [11].

During the course of these experiments we concentrate on achieving accuracy over small transformations. In our overall gesture recognition system, the hand object is extracted from the image and preprocessing steps are used to align and scale the hand image. This removes the need to recognise larger transformation at this stage.

### 3.7.1 Translation Transformation Experiments

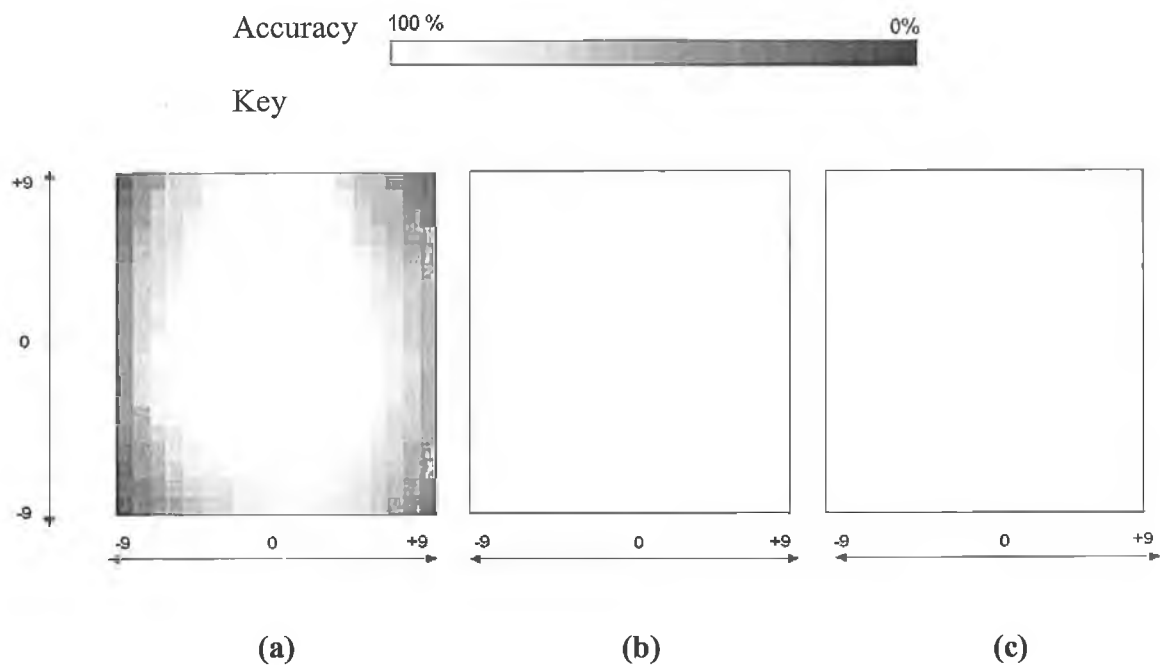
All test and training images used in these experiments are obtained using computer animation. Primarily we sought to test whether our invariant subspaces technique is robust to translation transformation. This involved developing a transformation subspace that was trained solely to achieve translation invariance. The transformation subspace has been trained using only the origin hand-shape and origin hand-shapes translated in all directions using combinations of 2, 4 and 6 pixels. Each subspace is therefore created from 49 training images (7x7 manifold of translations in all directions). Translation occurs in the 380x380 image, produced by Poser, before scaling to 32x32. Because of the small number of transformation images used, each subspace can be sufficiently represented using 7 eigenvectors.

In order to find the tangent distance, the test image is compared to the origin hand image, using the tangent distance technique, for each hand-shape. Note hand images are scaled and aligned as in the subspace technique.

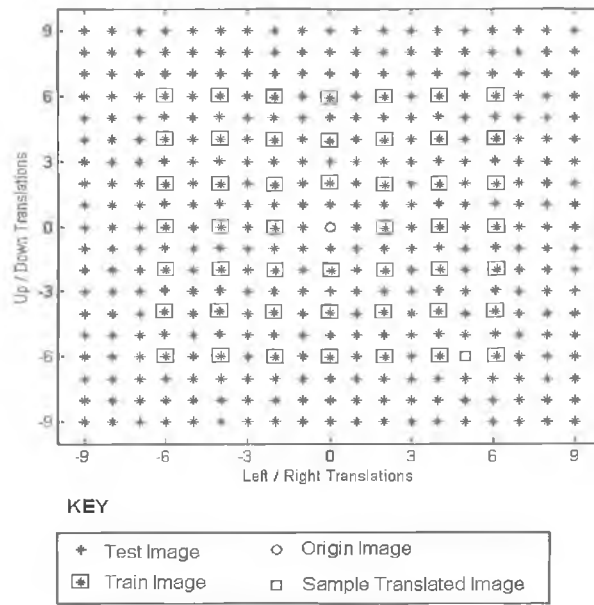
**Figure 3.9** gives a graphical representation of the performance of the various algorithms. These images illustrate the results of testing with the 19x19 translation manifold of the 24 origin hand-shapes images. This 19x19 translation manifold is produced by translating the origin by up to 9 pixels in all directions translated in all directions. A description of the type of the translation can be observed in **Figure 3.10**. The origin image is marked by a circle at point [0,0] in the manifold. As can be seen from this manifold description diagram combinations of translations in different directions are tested. An example of which is marked by the square at point [5,-6]

which denotes a 5 pixel translation to the right, along with 6 translations in a downward direction.

From **Figure 3.9** we can clearly see that both the Subspace Distance and the Subspace Tree Distance outperform the Tangent Distance. It is also evident that, as expected, some accuracy has been compensated using the quicker Subspace Tree Distance over the more precise Subspace Distance.

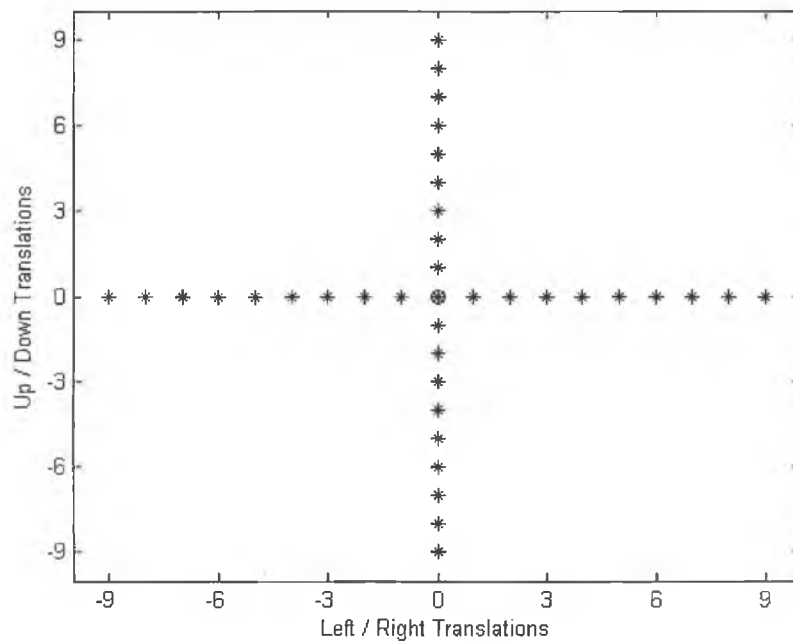


**Figure 3.9.** Performance images for (a) Tangent Distance, (b) Subspace Distance and (c) Subspace Tree Distance.



**Figure 3.10** A description of the 19x19 manifold used to test translation invariance

**Figure 3.11** presents a subset of the above test set. Here the test set is 96 (24 origin hand-shapes, each translated up, down, left and right). In this experiment we aimed to identify the point at which accuracy deteriorated when a translation transformation exists in the image in only one solitary direction; no translation combinations are used.



**Figure 3.11** A description of the Testing data used to test accuracy of solitary translations

The results of these experiments are as revealed in **Table 3.1**. Undoubtedly both the Subspace Distance and the Subspace Tree Distance significantly outperform the Tangent distance. It is evident that the tangent distance is only beneficial for small image translation transformations. Conversely the transformation subspace method presents 100% accuracy for the range of images it has been trained on, combinations of 2,4 and 6 pixel translations, and provides worthy accuracy on the remainder of test image translations. The Subspace Tree Search compares quite well to the regular subspace search, with some minor decrease in precision. Also illustrated is the favorable speed of the Tangent Distance. However, the subspace technique offers ample efficiency for the task of hand-shape recognition in real time. The Subspace Tree Search goes some way to addressing the speed issue but in doing so we compromise, slightly, on accuracy. Note all experiments are run on a standard PC using the Matlab interpreter with non-optimised code.

**Table 3.1.** Comparison of the performance of distance metrics for translated images.

Distance algorithm	Recognition Rate (%) for test images translated by the following number of pixels									Average Speed Per Image (seconds)
	1	2	3	4	5	6	7	8	9	
<b>Tangent</b>	100	100	97	85	74	60	40	28	21	0.0042
<b>Subspace</b>	100	100	100	100	100	100	100	97	94	0.0084
<b>Subspace Tree Search</b>	100	100	100	100	100	100	99	96	92	0.0068



### 3.7.2 Combining Rotation and Translation Transformations Experiments

Now that we are satisfied that we can recognise images that contain small translations, we now want to introduce rotation transformation. In particular we endeavor to test accuracy when both a translation and a rotation transformation occur in an image. **Table 3.2** shows the results of both translated and rotated object images when they are included in training and testing.

For the training phase, each hand-shape subspace has been created on a set of training images that contains translated origin images, translated origin images rotated 6° left and translated origin images rotated 6° right. As before only translations of 2,4 and 6 pixels are used. This means the total training set is 3,528, as shown in **Equation 3.2**, 147 images for each of the 24 training subspaces.

$$24 \text{ Origin image} \times 49 \text{ translations} \times 3 \text{ rotations} = 3528 \text{ images} \quad (3.2)$$

Test images contain the rotations in the ranges -12° to 12°, at intervals of 3°, in the yaw direction as described in **Table 3.2**. Translation combinations of 1,3 and 5 pixels are also included. The total test set then contains 10,584 images, as described in **Equation 3.3**, 441 images for each of the 24 subspaces created in training.

$$24 \text{ Origin image} \times 49 \text{ translations} \times 9 \text{ rotations} = 10584 \text{ images} \quad (3.3)$$

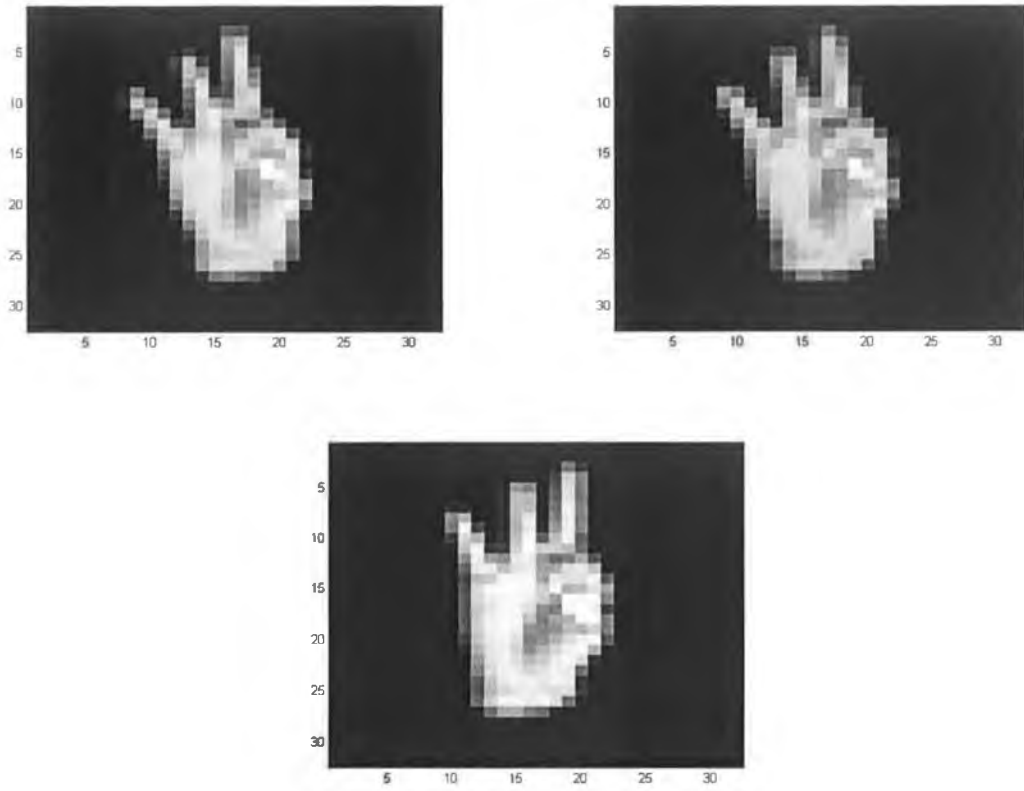
Once again the subspace distance demonstrates a superior performance. The Subspace Tree Search offers similar results, while the Tangent Distance technique deteriorates when non-trivial rotations are tested. A reason for the inferior performance is that the object image rotations are acquired in 3D space whereas Tangent Distance aims to approximate 2D image rotation transformations.

**Table 3.2.** Performance of Tangent Distance and Subspace Distance for translated and rotated object images.

Distance algorithm	Recognition Rate (%) for test images rotated by the following number of degrees and using different translations.									Average Speed Per Image (seconds)
	-12°	-9°	-6°	-3°	0	3°	6°	9°	12°	
<b>Tangent</b>	54	70	84	81	86	83	77	62	46	0.0042
<b>Subspace</b>	96	99	100	100	100	100	100	99	93	0.0098
<b>Subspace Tree Search</b>	95	99	98	97	96	95	95	93	90	0.0072

### 3.7.3 Combining Rotation Translation and Shape Transformations Experiments

In the ensuing experiment we introduce a series of random shape transformation to the test set in order to determine if our Subspace technique can effectively handle the real world situation where many different variations of hand-shape are present and where gestures are performed inconsistently. The random shape variances can easily be obtained using Poser. Random variations are achieved by slightly deviating each joint in the hand in an unsystematic manner in all allowable directions. An example of three different random variations to the static gesture representing the letter ‘g’ is shown in **Figure 3.12**. Here the difference between the posture of the hand-shape, the spacing between fingers and the position of the fingers is evident.



**Figure 3.12** Images of the ‘g’ hand-shape that contain some random shape variations

In this experiment we used the same training set as used in the previous experiment, **Section 3.7.2**. Therefore we are simply testing the recognition accuracy of our technique when shape differences have not been included in the training phase. The test set contains 31,752 images, as described in **Equation 3.4**, 1,323 images for each of the 24 subspaces.

$$\begin{aligned}
 &24 \text{ Origin image} \\
 &\times 49 \text{ translations} \\
 &\times 9 \text{ rotations} \\
 &\times 3 \text{ random shape variations} \\
 &= 31752 \text{ images}
 \end{aligned}
 \tag{3.4}$$

In **Table 3.3**, the result of introducing random shape variations to the test images are displayed. Once again the subspace technique outperforms the other two techniques.

However, the recognition accuracy has decreased somewhat because of the shape variations introduced. Clearly the subspace technique and the Subspace Tree Search provide a better invariance to random hand-shape variations compared to the tangent distance.

**Table 3.3.** Performance of Tangent Distance and Subspace Distance for translated and rotated object images that contain random shape distortions.

Distance algorithm	Recognition Rate (%) for test images rotated by the following number of degrees and using different translations along with ad-hoc shape distortion.									Average Speed Per Image (seconds)
	-12°	-9°	-6°	-3°	0	3°	6°	9°	12°	
<b>Tangent</b>	47	63	70	77	78	72	66	53	40	0.0044
<b>Subspace</b>	90	93	99	99	98	96	89	86	85	0.0104
<b>Subspace Tree Search</b>	87	87	94	95	96	94	88	85	85	0.0078

### 3.8 Summary

In this chapter we presented our novel technique of using subspace classifiers, constructed using images created from computer animation, to classify hand-shape images. This basic Subspace classification runs quite fast and is sufficient to classify a reasonable number of static hand gestures in real time. However, if superior speed is necessary we present a Subspace Tree search that can be utilized. As shown in the experimental results some accuracy will be compromised by using the faster tree search.

In the series of tests performed we sought to test whether this subspace technique could cope with translation and rotation transformations, which are common place in static hand gesture recognition. We compared the results of the Subspace Distance and the

Subspace Tree search with the Tangent Distance. It is worth noting that the point where both these techniques failed was when similar images were misclassified due to rotation or random variations in a single hand shape. This is a problem in real world sign language recognition where recognition ambiguity can occur. Humans usually solve this by classifying shapes based on contextual information. This problem is addressed somewhat in Chapter 5 when dealing with dynamic gestures.

These experiments show that while the Tangent Distance is an effective technique for small 2D image transformations, its usefulness does not compare well with Transformation Subspace Distance for robust hand-shape classification. The Tangent Distance is not able to represent the complex manifold of transformations as well as the Transformation Subspace Distance can.

One point to note with these experiments is that all the test data was created using computer animation. The advantage of using computer animation is, we can accurately and systematically extract data at predefined positions, configurations, and angles. However, while every care was taken to include only natural and reasonable configurations of static hand gestures, some unconventional images may be present in the datasets. This is particularly relevant to the experiments illustrated in **Section 3.7.3** where random variations were introduced to the test set. This might even account for the slight accuracy deterioration that was observed. A true reflection of the accuracy and practicality of the Transformation Subspace Distance is presented in the next chapter when we test with images of real static hand-shape gestures rather than those generated from computer vision.

## ***CHAPTER 4***













### ***REAL HAND IMAGE CLASSIFICATION***

#### **4.1 Introduction**

So far, a robust hand-shape recognition system has been proposed based on a subspace classifier. The subspaces are constructed, with a-priori knowledge, from images acquired using Poser modelling software. However, for this technique to be useful we need to be able to classify images of real hands rather than testing with images produced from Poser. This would mean training the system with Poser images and testing with images of a human hand. Such a system would inherently be multi-user as it would be trained and tested by different users. Our system now has to deal with many of the problems mentioned in the literature regarding template-matching techniques; accurate hand segmentation, skin colour, illumination, hand size and distance from the camera.

#### **4.2 Hand Image Pre-processing**

We have developed a detailed pre-processing step to counteract these issues. These steps are performed on the Poser images used to train the system along with the real hand images used to test the system. This process is described in detail below and illustrated in **Figure 4.1** using two different users with different skin colour where the hand is at different distances from the camera.

	Original Image	
	Segmented image of the right hand, produced from the segmentation and tracking stage	
	Hand Image is converted to Grayscale	
	Hand objects are centred using the centre of the bounding box and resized to meet the criteria that they occupy a predefined area within a 32*32 scaled image	
	Hand is colour normalised using a colour histogram equalisation technique.	
	Images are convolved with a gaussian kernel to reduce noise.	

**Figure 4.1.** Hand Image Pre-processing Steps

#### 4.2.1 Hand Segmentation

To segment the hand we use the technique devised by Awad et al. [23]. They present “A Unified System for Segmenting and Tracking the face and Hands” that is specifically designed for Sign Language Recognition. The hands and face are initially segmented by locating skin pixels in the image. Skin Pixels lie in a predetermined range in RGB colour space. Initially we assume only three skin coloured objects exist in the frame, the two hands and the head. The head is identified as being the uppermost skin coloured object while the left and right hands lie either side. These skin objects are then tracked using a Kalman filter based algorithm. Tracking improves the segmentation results. This is achieved by using the assumption that the hand position doesn’t change substantially in successive frames. Therefore we can reduce the search space involved in finding the skin-coloured objects in the succeeding frame based on their position in the current frame. They have shown that this combination of colour, motion and position information can provide accurate segmentation of the hands and face in sign language recognition. **Figure 4.2** shows a working example of this procedure. (a) displays the original input image while (b) shows the results of the segmentation stage. Here we can see that the three skin coloured objects have been detected and isolated.



(a)



(b)

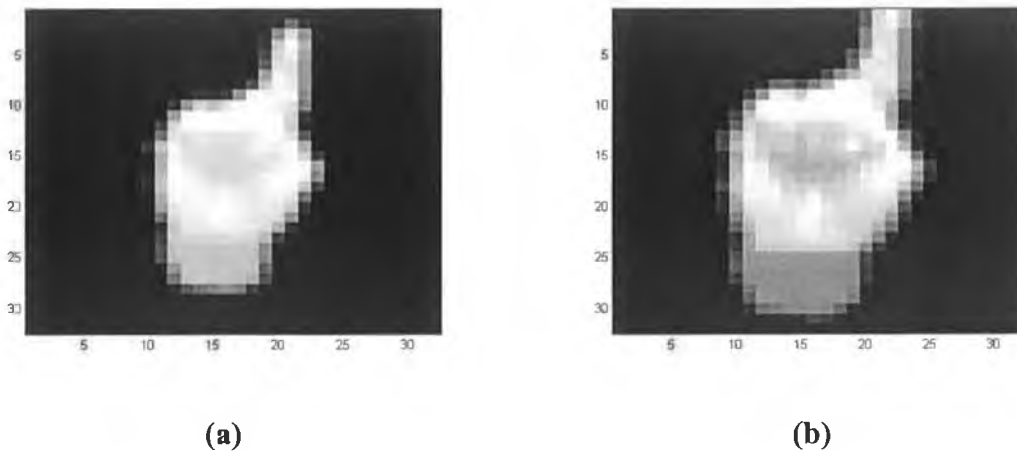


**Figure 4.2** Results of skin segmentation (a) original image, (b) image produced from skin segmentation.

#### 4.2.2 Hand Scaling and Alignment

Once the hand has been identified and segmented it should be scaled and aligned to ensure the system can deal with users with different sized hands and users at differing distances from the camera. We exploit a simple but effective practice of scaling the hand objects so that they occupy a predetermined area in a 32x32 resized image. Our experiments have shown that ensuring the hand object consumes 25% of the image pixels produces optimal results. The goal of this scaling is to ensure the hand objects are as large as possible while remaining totally encapsulated in the image boundary. **Figure 4.3** shows two examples of a hand object that have been scaled differently, (a) hand object scaled to occupy 25% of image pixels, (b) hand object scaled to occupy 30% of image pixels. Looking at (b) it is evident that the object penetrated the image boundary. While having the object only being represented by 25% of the image pixels seems like a small percentage, it is necessary to guarantee the full object is contained within the image. This fact also substantiates the concept of using a dimensionality reduction technique such as PCA. If the relevant information in the image only constitutes 25% of the actual image, then it is inherently possible to reduce the dimensionality of this representation of a hand shape.

Alignment is accomplished by repositioning the hand object so that the centre of the bounding box lies in the centre of the image. In **Section 3.3** we identified how our subspace technique was trained to overcome some of the inadequacies of this simple alignment procedure.



**Figure 4.3** Example of two different scaling factors, **(a)** object scaled to occupy 25% of the image pixels, **(b)** object scaled to occupy 33% of image pixels.

#### 4.2.3 Skin Colour and Illumination Variation

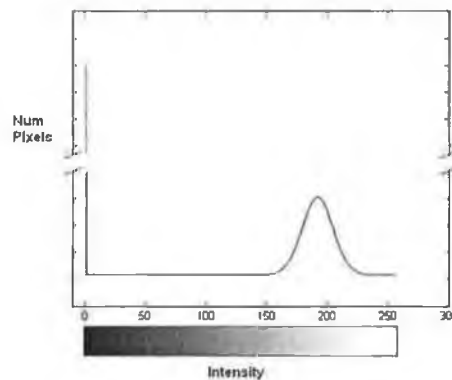
Removing skin colour and colour variance due to illumination is essential in an appearance based multi-user hand-shape recognition system. First the hand image is converted to greyscale, this reduces the space in which colour can be represented. In order to colour normalise each hand image in greyscale space we have incorporated a colour histogram equalisation approach into our system. Colour histograms are graphs that depict the colour distribution of pixels in an image. The histogram can be calculated simply by counting the number of occurrences of each colour value in the image. Histogram Equalisation is the process of redistributing the colour values in the image so that the image histogram takes a predetermined form.

We know from the hand-scaling step that all hand objects are resized to occupy the same area within an image. With this in mind a common histogram can be defined that can represent all hand images. We defined this histogram as per **Figure 4.4**. It contains a large spike that represents the background of the image; this is located at the beginning of the colour scale because the background pixels are set to 0. The Gaussian-

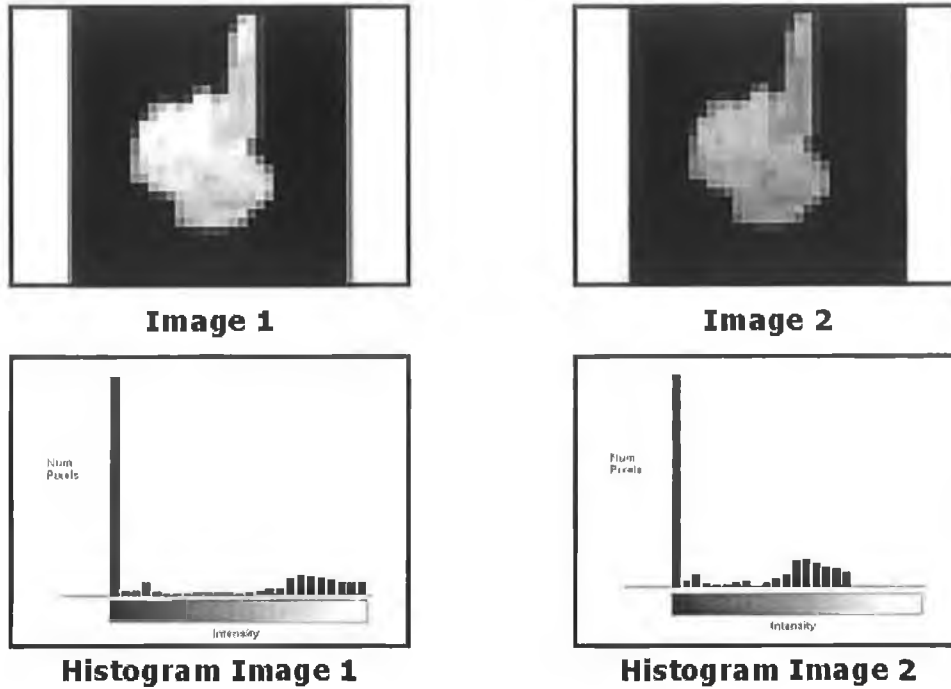
shaped pulse towards the end of the colour scale represents object pixels. This positioning is important to maximise the contrast of the normalised hand image.

**Figure 4.5** illustrates two hand images that differ only in their skin colour. Also shown are their colour histograms calculated prior to histogram equalisation. **Figure 4.6** displays the same two images along with the resulting images produced from the histogram equalisation stage using the baseline histogram from **Figure 4.4**

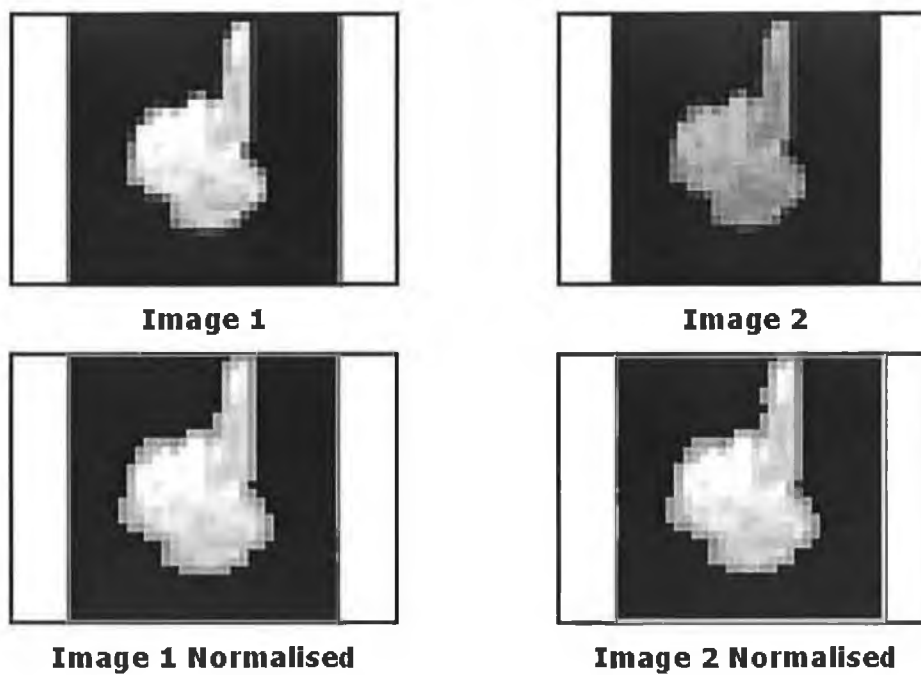
In **Section 4.4** we compare the usefulness of the histogram equalisation technique with other commonly used methods in computer vision such as Local Binary Pattern (LBP) and edges.



**Figure 4.4** Baseline Histogram for Histogram Equalisation



**Figure 4.5.** Two Poser hand images differing in skin colour along with their colour histograms.

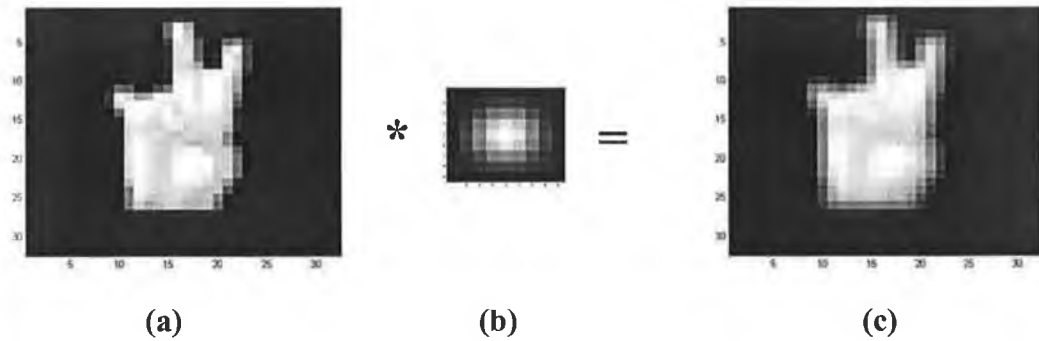


**Figure 4.6.** Two Poser hand images differing in skin colour along with their colour normalised images.

#### 4.2.4 Image Filtering

We have found that it is useful to apply a simple Gaussian filter to an image before classification. This filter can help smooth out noise in an image. Filtering of noise can considerably flatten features that are local to the individual hand of the user. Dispersing this noise improves the invariance of the recognition.

The hand image is convolved with a 9x9 gaussian kernel with a small standard deviation to ensure the filtering does not blur important information in the image. This approach is revealed pictorially in **Figure 4.7** where the input image **(a)** is convolved with a Gaussian filter **(b)** and the resulting image is shown in **(c)**. Some results on how this minimal procedure can enhance recognition result are documented in **Section 4.5**.



**Figure 4.7** Convolution of images with gaussian kernel to filter noise, **(a)** original image, **(b)** gaussian kernel, **(c)** filtered image.

#### 4.3 Recognition Experiments

In order to classify real hand images we create a subspace for each hand-shape as described earlier. However, this time all the training images will be preprocessed using the techniques described above. Similarly all test images will traverse through the same pre-processing steps. In the previous section when we trained and tested with images

produced from Poser we could constrain many parameters. However, now when we test with real hand images the transformation subspaces need to be more robust to user interpretation of static gestures. Even simple finger spelling hand-shapes are open to rotation and arbitrary shape transformation in all directions. Therefore we need to increase the 3D rotations included in the training data.

In this test we use 28 hand-shapes consisting of 23 static finger spelling and 5 static counting gestures from Irish Sign Language. A subspace for each hand-shape is now created by performing PCA on the set of 3,969 images as described in **Equation 4.1**.

$$\begin{aligned}
& 1 \text{ Origin image}^1 \\
& \times 49 \text{ translations}^2 \\
& \times 9 \text{ rotations in yaw direction}^3 \\
& \times 3 \text{ rotations in pitch direction}^4 \\
& \times 3 \text{ pitches in roll direction}^5 \\
& = 3969 \text{ images}
\end{aligned} \tag{4.1}$$

<sup>1</sup> Origin hand image that can be defined as being the perfect orientation of the hand-shape.

<sup>2</sup> Origin hand-shapes translated in all directions using combinations of 2, 4 and 6 pixels.

<sup>3</sup> 9 rotations are used in the yaw direction as this is the direction that contains most significant deviation.

These rotations are 3 degrees apart covering a total pitch of 24 degrees.

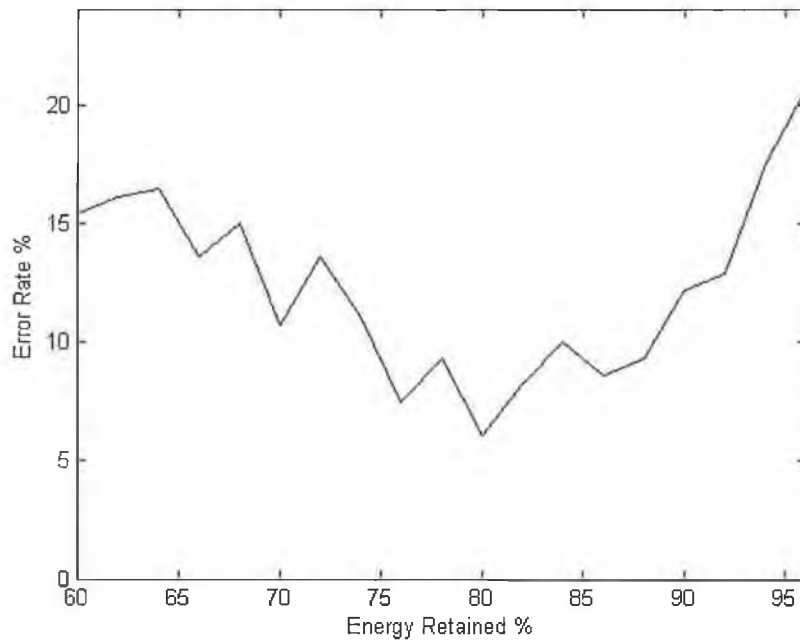
<sup>4</sup> 3 rotations in the roll direction, each at 10 degrees covering a total pitch of 20 degrees.

<sup>5</sup> 3 rotations in the roll direction, each at 10 degrees covering a total pitch of 20 degrees.

We developed a test set in order to test the amount of energy we need to retain in each of these subspaces. This test set contained 560 images, 20 occurrences of each of the 28 hand-shapes that were used. All these images were acquired from one trained user of the system over 4 separate sittings on 2 different days.

The first objective of our experiments was to identify the energy retention value that gives superior recognition. **Figure 4.8** clearly shows that when 80% of the energy is retained the lowest error rate is achieved. One explanation for this is once we go over 80% the subspaces attempt to retain information that is local to that of the individual user, i.e. local characteristics of the computer animation images. It is important to find this balance between retaining as much information as possible without introducing noise into our subspaces.

80% energy retention entails keeping 12-16 of the most significant eigenvectors, depending on the hand-shape. Having a low number of eigenvectors is also important to maintain efficiency. The effect the number of eigenvectors has on complexity can be reviewed in **Equation 3.1**.



**Figure 4.8.** Plot of Error Rates against Energy Retained in Subspaces

Preserving 80% energy in the subspace we achieve 94.5% recognition accuracy for our test set. The performance accuracy of each individual static gesture can be observed in **Table 4.1**. This table exhibits the confusion matrix for the static gesture recognition vocabulary. Most confusion is caused where gestures are very similar. Gestures can be compared by studying the static gesture vocabulary in appendix A. The two gestures that give highest confusion are U and R. These gestures only differ slightly, when performing U, the index and middle finger lay parallel, while performing R the index and middle finger are crossed. These differences become particularly minute once the images are scaled to 32x32.

**Table 4.1** Confusion Matrix for Static Gesture Recognition

	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	Y	1	2	3	4	5
A	18																	1	1									
B		20																										
C			20																									
D				16				2											2									
E					20																							
F						19																			1			
G							18							2														
H								18						1	1													
I									20																			
K										18																	2	
L											20																	
M												20																
N													20															
O														19				1										
P															19				1									
Q										1						17					2							
R																	17			3								
S	1				1													18										
T					1												1		18									
U																				20								
V																					20							
W																						1	19					
Y																								20				
1																									20			
2																										20		
3										1																	19	
4																						1						19
5																										3		17



#### 4.4 Colour Invariance Experiments

In addition to these experiments we also tested other image pre-processing techniques to see if it is possible to improve on our results. In particular we compared the results of the image histogram technique to that of using LBP along with object edges.

Edge Detection is a popular processing technique in image processing. Edge Detection involves detecting discontinuities in the colour intensity values. These discontinuities are observable over any range of colour and object lighting so detecting edges has the advantage that they are inherently invariant to object colour and lighting.

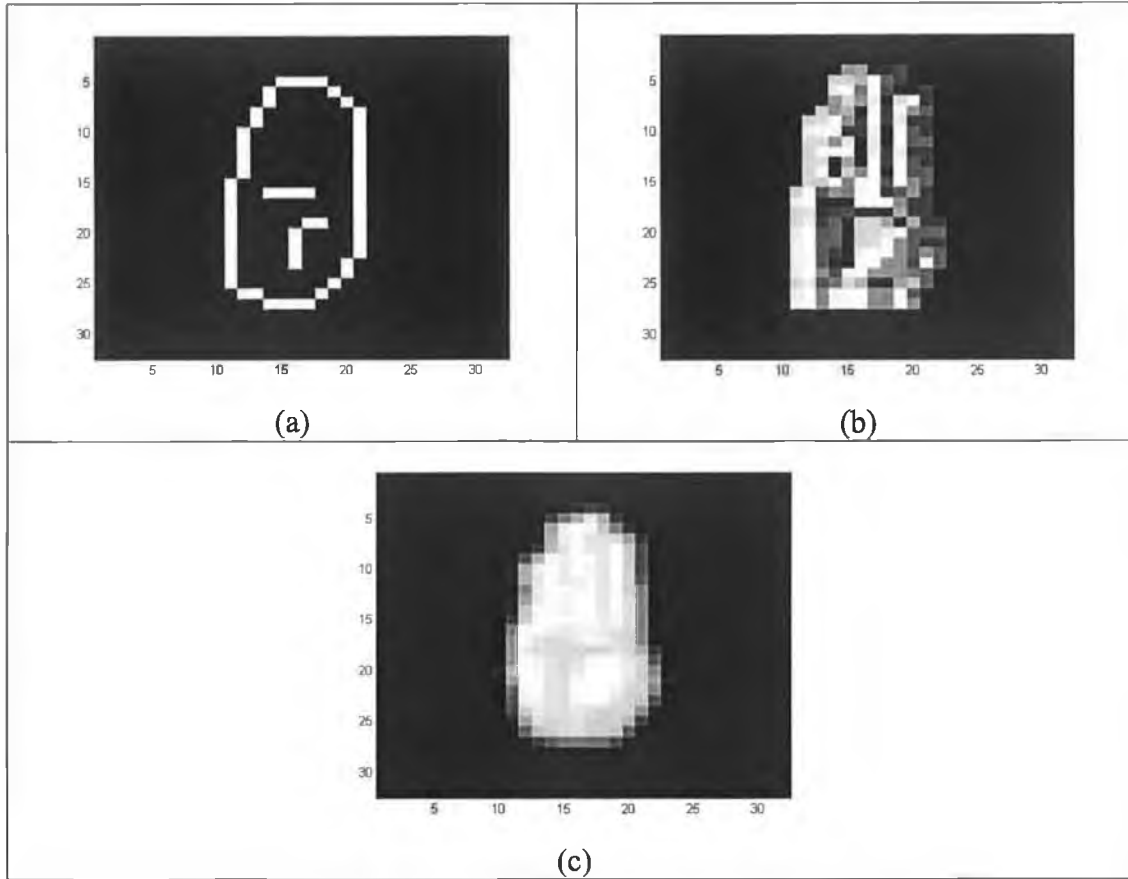
LBP is a texture analysis operator and can also be defined as being colour invariant. LBP works by convolving a 3x3 kernel with the image. This kernel is designed to emphasise the local spatial structure surrounding each pixel. Once again this spatial structure exists despite the colour intensity of the image, making this technique invariant to colour.

In order to test which technique was more accurate and to see if we could improve the precision of our colour invariant system we tested and compared image histogram equalisation, edges and LBP. The experiments were carried out exactly as described earlier in this chapter except edge and LBP were used instead of images histogram equalisation. The results are as shown in **Table 4.2**. Here we see that the image histogram technique offers a far greater accuracy. **Figure 4.9** shows some samples of images after each of the pre-processing techniques. As can be seen from **Figure 4.9 (a)** not much information is preserved using the edge technique. The main reason for this is

that images are scaled to 32x32 so at this resolution it is difficult for an edge detection to identify definite edges without introducing a lot of noise. Therefore a large amount of information that is used to distinguish different classes is discarded, consequently deteriorating recognition accuracy. **Figure 4.9 (b)** Illustrates the same images once the LBP transform has been undertaken. While the intrinsic texture of the hand is evident, it is clearly obvious from this image that a lot of noise is introduced in the resulting image. So when PCA is performed on this set of images, containing this vast amount of noise, the eigenvectors produced will consequently try to represent the noise, as the noise will depict the greatest variation of the data. An example of a histogram-equalised image is described in (c). Although the image has been similarly scaled, it still retains the basic natural structure of the hand object while being relatively free from noise and an element of colour intensity normalisation has been introduced.

**Table 4.2** Results of using different techniques for colour intensity invariance

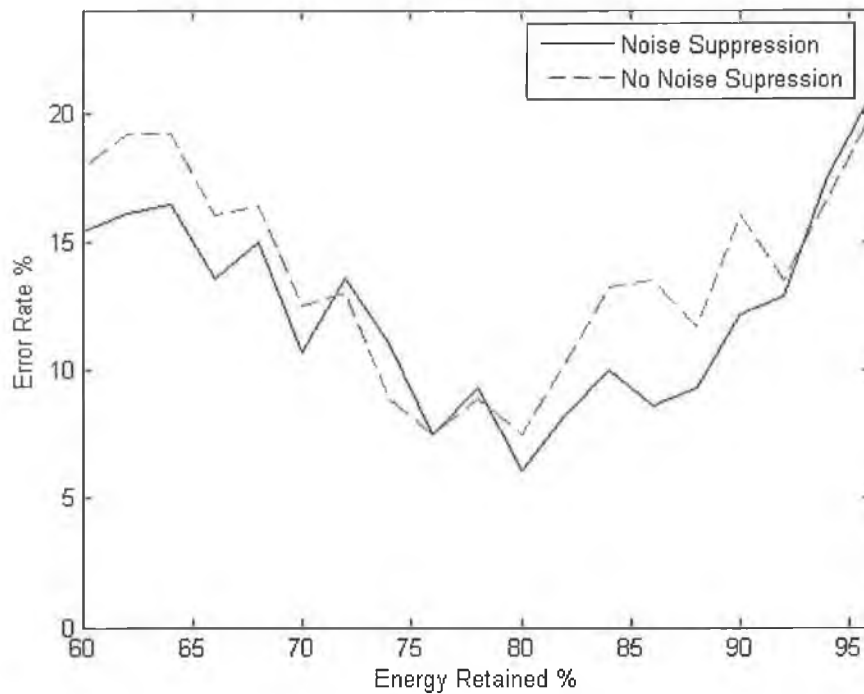
<b>Technique</b>	<b>Image Histogram Equalisation</b>	<b>Edge</b>	<b>Local Binary Patterns</b>
<b>Error Rate</b>	<b>5.8%</b>	<b>19%</b>	<b>33%</b>



**Figure 4.9.** Different pre-processing method to deal with colour invariance (a) object edge, (b) LBP, (c) image histogram.

#### 4.5 Noise Reduction Experiments

A simple noise suppression method was outlined in **Section 3.6** using a Gaussian filter. This filter allows to further smooth noise that arises due to the vast number of local differences in the hand images. **Figure 4.10** illustrates the effectiveness of this straightforward approach. Here the error rates are displayed for the different number of eigenvectors retained. The dashed line shows the results without noise suppression while the full line displays the results once noise suppression has been employed. Here it is apparent that the noise reduction has a beneficial effect on the error rates and an improvement, on average, of 1.6% is achieved.



**Figure 4.10** Error rates before and after noise suppression is employed.

#### 4.6 Summary

In this chapter we have presented a real hand-shape classification system. A novel technique was introduced for training a subspace classifier from images, created from computer animation, which was later tested using real hand images. Using the animation a large labelled training data set was produced and the subspaces created. This technique allowed us to introduce invariances to rotation and translation. To allow the system to successfully recognise images of human hands, from the trained data set of computer animated hand images, we introduced a chain of relatively complex pre-processing steps to remove user dependant features such as colour and scale. Creating a multi-user recognition is imperative for practical gesture recognition system.

A set of test images was created employing a single user at different sittings. A series of tests were then performed to analytically evaluate our technique. During the course

of our experiments we endeavoured to identify particular techniques and parameter values that improved the accuracy of our system. We established the colour histogram to be the most successful method of achieving colour invariance. Simple parameter values such as the amount of energy retained in each subspace proved to be critical and influence the test results significantly. Also we tested the system using a noise reduction technique which once again offered further improvement on our systems accuracy.

It should be noted that all of these experiments in this chapter were performed using the right hand as the signing hand. In reality gesture of identical hand-shape can be performed with the left hand. The recognition of equivalent left handed static gestures can be achieved in a simple manner by considering the left hand as a mirror image of the right. This means the left hand can be classified using the same training data. The pre-processing steps will be as before except a further step is included to find the mirror image of the hand-shape.

# *CHAPTER 5*

## *DYNAMIC GESTURE RECOGNITION*

### **5.1 Introduction**

In the previous chapter we tackled the challenge of recognising static gestures. However, static gestures are rarely used in the real world. Humans performing their everyday activities are more likely to use dynamic gestures. The problem of dynamic gesture recognition is far more complex than that of static gesture recognition. Dynamic gesture recognition requires both temporal and spatial movement recognition of both hand movement and hand-shape.

We have devised a system using Hidden Markov Models (HMMs) to recognise dynamic gestures. Many researchers in the area of gesture recognition have explored HMMs. Some of these techniques have been described in the literature review, Wu et al. [1], Shamie et al. [3], Huang et al. [33] and Chen et al. [4]. However, most of the interest in the use of HMMs in gesture has arisen from Starner et al. [31] along with Vogler et al. [32] who both worked independently on ASL recognition. A HMM is a tool for representing probability distributions over a sequence of observations [24]. Different researchers have chosen many different types of features as the input observations and can explain the vast amount of research in the area. The option of using either Discrete HMMS (DHMMs) or Continuous HMMS (CHMMs) means we have the choice of using discrete or continuous input features. DHMM based systems will partition the feature spaces into a number of distinct sections. Each of these

sections will have a unique index that will act as the input observations for the DHMM. CHHMs use probability density functions that operate on the continuous input observations. It is worthwhile noting that in general DHMMs run faster and require less time to train compared to CHMMs. In working systems the type of input features selected, continuous or discrete, will depend on the techniques used to extract the features. This will then influence the type of HMM employed

In this chapter we briefly describe the basic ideas behind HMMs. In addition we will describe our technique to classify our ISL gestures. It is based on DHMMs that act on both hand-shape and hand position features. DHMMs were chosen because the features extracted from our hand classification system are inherently discrete. Also using DHMMs further decreases computation time allowing real time classification. We present some experimental results that are encouraging for future expansion of the system.

## **5.2 Discrete Hidden Markov Models – An Overview**

We will now present a concise description of HMMs, more detailed tutorials and summaries of HMMs can be found in [34] and [35]. HMMs consist of a number of states, which are linked together in a chain like structure. Each HMM will possess a start and end state, along with a set of state transition probabilities that are used to estimate when to transfer between states. Usually each state will be representative of a portion of the input observations. Furthermore the current output of the model is stochastically based on the existing state of the system. It is this doubly stochastic property that gives HMMs their superiority to regular Markov chains.

Construction of a DHMM involves the following five attributes:

- The number of states  $N$  in the model. Each model will contain both a start and end state, along with a number of intermediate states that will vary in different models;
- The discrete number  $M$  of allowable observations. In our case this will be the discrete hand-shape and hand position combinations;
- The state transition probability distribution  $A$ . This is learned from the training data using the Learning Problem;
- The observational symbol  $B$ . This is also learned from the training data using the Learning Problem;
- The initial state distribution  $\pi$ .

In practice a HMM is categorised by  $\{A, B, \pi\}$  because  $M, N$  are constants. Given this definition of a HMM 3 problems need to be addressed.

### 5.2.1 The evaluation Problem

If we have a HMM  $\lambda$  and a sequence of Observations  $O = \{o_1, o_2, o_3, \dots, o_L\}$ , where  $L$  is the length of the observation sequence, how do we compute  $P(O, \lambda)$ ? This probability can be calculated quite easily using simple probabilistic arguments. However, this involves a number of calculations in the order of  $N^T$ . The Forward/Backward algorithm can significantly lower the complexity of this operation. This technique is used in our system to calculate the likelihood of each HMM for a sequence of observations of a gesture.



### 5.2.2 The decoding problem

If we have a HMM  $\lambda$  and a sequence of Observations  $O = \{ o_1, o_2, o_3, \dots, o_L \}$ , what is the most likely state sequence in the model that produced  $O$ ? The Viterbi algorithm is a well known solution to this problem. In our system we only worry about the actual classification of the gestures and not the individual states traversed by the each HMM. Therefore the decoding problem is superfluous in our system. However, more details can be obtained in [34] and [35].

### 5.2.3 The Learning Problem

Given a HMM  $\lambda$  and a sequence of Observations  $O = \{ o_1, o_2, o_3, \dots, o_L \}$ , how should the model parameters  $\{A, B, \pi\}$  be adjusted in order to maximise  $P(O, \lambda)$ ? Usually some initial values are given for an individual HMM. These values are then learned and updated based on the training samples. This means increased training can achieve a more accurate model. An Expectation-Maximisation method known as Baum-Welch has been proposed to solve this problem.

## 5.3 Input Observations for DHMM

As stated above our dynamic gesture recognition system uses both hand-shape and hand position information to classify gestures. For our dynamic gesture recognition system the sequence of observations are feature vectors containing two elements, both of which are positive integers. The first denotes the group to which the static hand-shape has been classified. The second symbolises the position the hand occupies in the image. **Equation 5.1** describes a typical gesture that is represented by a sequence of 2D discrete observations,  $S$  and  $P$ , where  $S^i$  and  $P^i$  signify the hand-shape and hand position, respectively, at time  $i$ .

$$\{ [S^1, P^1], [S^2, P^2], [S^3, P^3], \dots, [S^n, P^n] \} \quad (5.1)$$

In order to classify the hand-shape and hand position the hand must first be segmented from the image and its centre must be located. To do this we utilised the approach proposed by Awad [23]. They present “A Unified Method for Segmentation and Tracking of Face and Hands in Sign Language Recognition”. Some of the aspects of their technique were described in **Section 4.2.1**. Successful segmentation of the hands and face is achieved by using three key characteristics, colour, motion and position. These skin objects are then tracked using a Kalman filter based algorithm. It is useful to consider both segmentation and tracking in parallel as they are co-dependent; accurate segmentation aids successful tracking while successful tracking further improves the segmentation.

### 5.3.1 Hand-shape Classification

The hand-shape classification method used is the subspace classification technique as described in **Chapter 3**. Instead of merely using the 28 static gestures used in finger-spelling and counting, we extend the system by adding further hand-shapes that occur in our dynamic gesture vocabulary. These new hand-shapes include some instances where the original static gestures have been substantially rotated into an unrecognisable state and therefore need to be considered as a different hand-shape. In all, 40 autonomous hand-shapes are used, all of which are given a unique index. This means the output of the hand-shape classifier will be an integer in the range 1-40.

### 5.3.2 Hand Position Classification

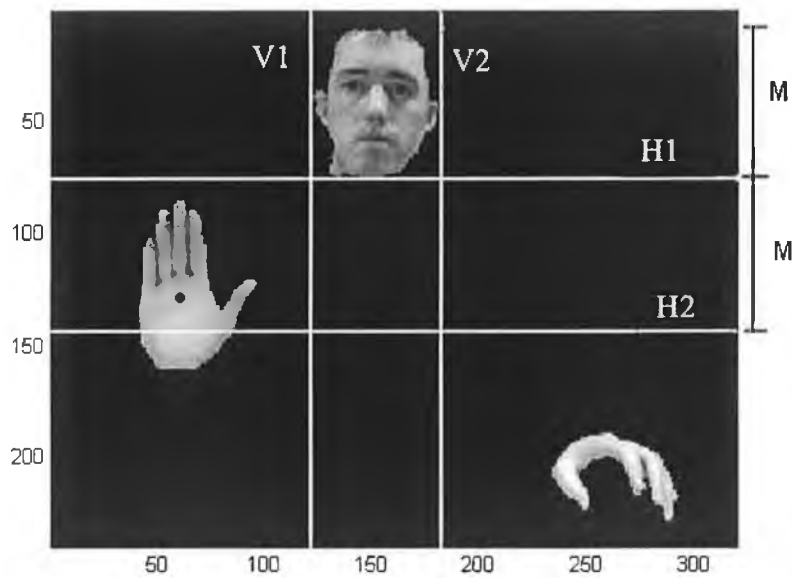
The second feature extracted is used to describe the position of the hand in the image with relation to the head. The position of the hand is classified by determining the

section of the image that the centre of the hand lies in. This centre is calculated by simply finding the centre of the hands bounding box. Care needs to be taken when dividing the image into sections to ensure we can cope with situations when the user is at different distances from the camera and when the user is at different positions in the image.

This image is divided into 9 sections as shown in **Figure 5.2**. Each of these sections are given a unique label as illustrated in **Figure 5.1**. These sections are created by dividing the image vertically by drawing two lines V1 and V2 either side of the head. The first of the horizontal lines, H1, is located directly under the head. The second, H2, is placed M pixels below H1, where M is the length of the head object. Classifying the position in this manner ensures that location information is calculated invariantly to the position of the user in the image.

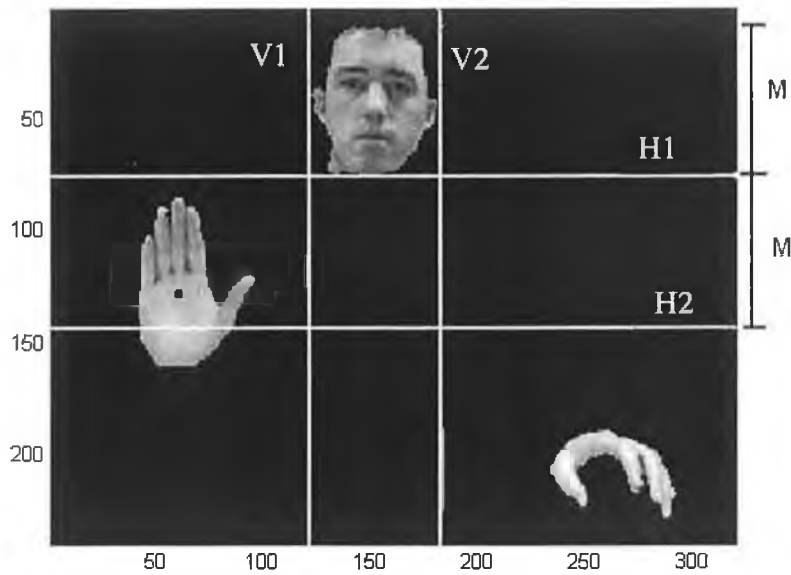
1	2	3
4	5	6
7	8	9

**Figure 5.1** Section labelling for a divided image



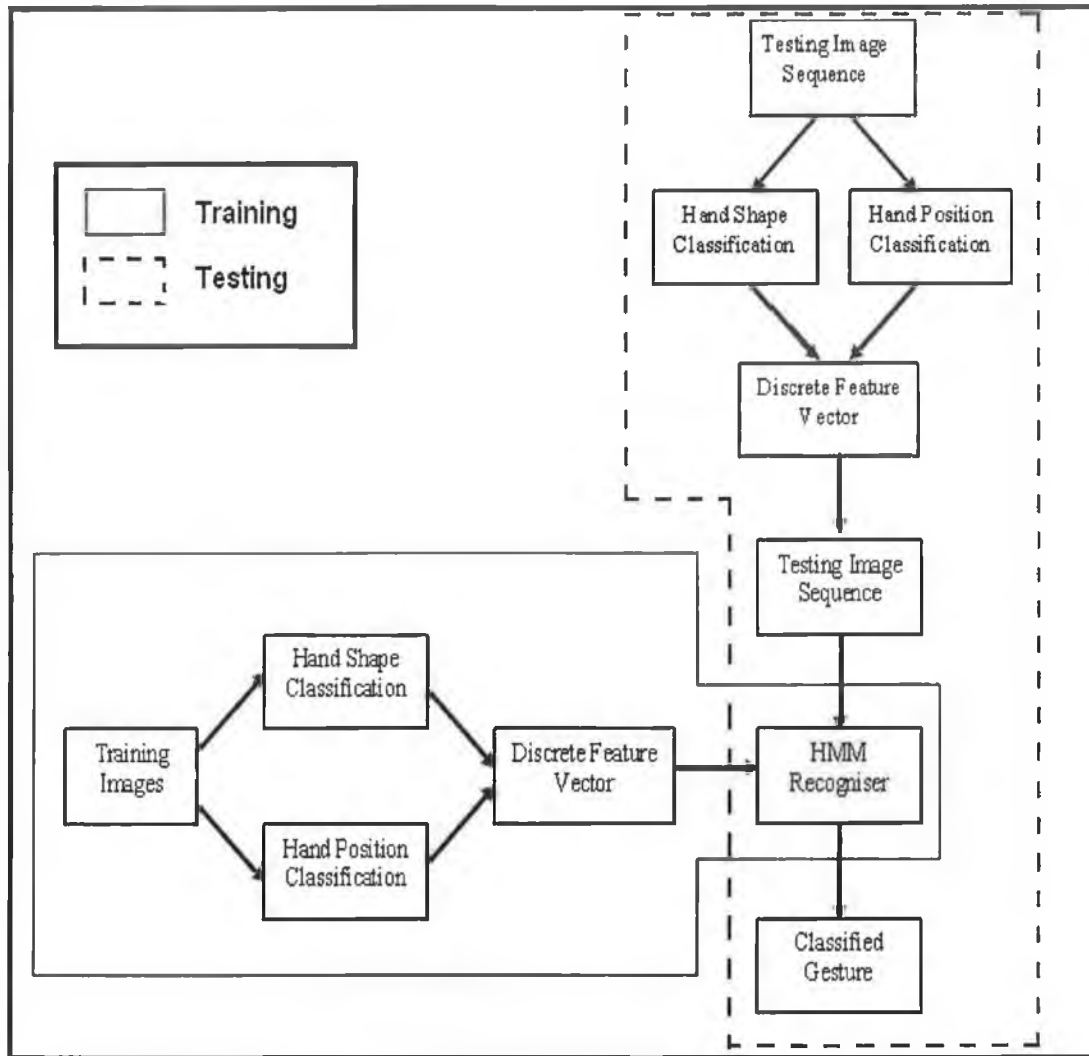
**Figure 5.2 – Hand Position Classification**

A dynamic gesture can now be represented as a sequence of these feature vectors, containing both shape and position information. A HMM is trained for each possible gesture using many different examples. A gesture is classified online, by manually identifying its start and stop points, then finding the HMM with the highest probability for the feature vector of the test sequence. **Figure 5.3** shows the flow diagram for both training and testing in our DHMM system. As illustrated the hand-shape and position information are calculated independently, then combined and input in to the HMM recogniser.



**Figure 5.2 – Hand Position Classification**

A dynamic gesture can now be represented as a sequence of these feature vectors, containing both shape and position information. A HMM is trained for each possible gesture using many different examples. A gesture is classified online, by manually identifying its start and stop points, then finding the HMM with the highest probability for the feature vector of the test sequence. **Figure 5.3** shows the flow diagram for both training and testing in our DHMM system. As illustrated the hand-shape and position information are calculated independently, then combined and input in to the HMM recogniser.



**Figure 5.3** Flow diagram for both training and testing in our DHMM system.

## 5.4 Experiments

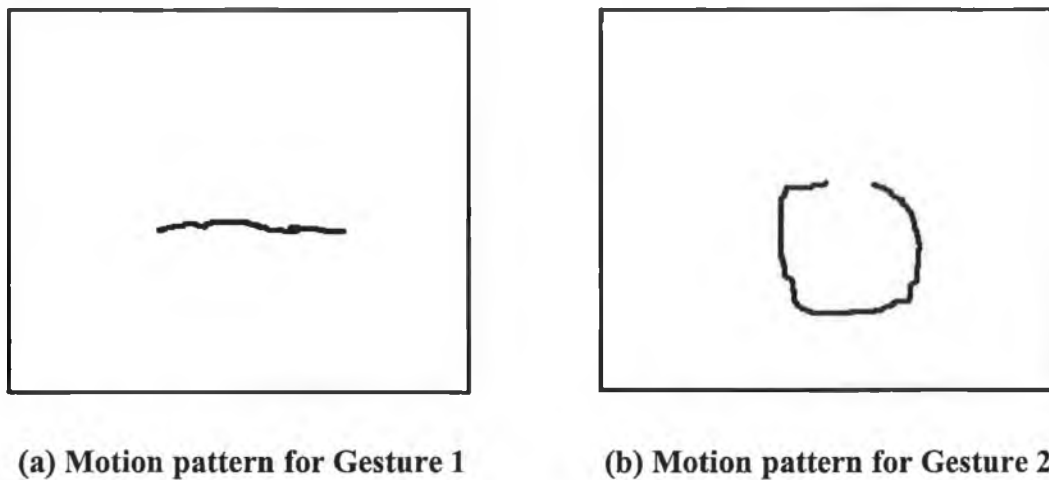
In order to test the accuracy of our dynamic gesture recognition system we have generated a vocabulary of 17 dynamic gestures, these gestures are outlined in **Appendix C**. In ISL many of the gestures are two handed involving movement and interaction of both hands. We, however, concentrate on one-handed gestures. When deciding this lexicon care was taken to ensure similarities between gestures are present to test the strength of the system. Some gestures exist that contain the same hand-shape but are

performed at different positions while others contain different hand-shapes performed in the same position in the image.

20 samples of each isolated gesture were recorded employing 2 different users, of different racial origins, over 4 different days. One of the users is a trained expert of the system while the second is a novice who performs the gestures as instructed by the trained expert. The importance of capturing videos using multiple people with a different familiarity of the system, at altered sittings, is crucial to ensure a large variation is present in the gestures, in both the training and testing sequences. The videos are captured in an office environment with additional lighting to the front of the user. Each of the videos are captured at 25 frames per second (fps). In our experiments the samples are divided into test and training sets by random sampling. The proportion of test and training data was then varied over the recognition experiments. A DHMM was trained for each gesture using the selected training data. We then tested the recognition accuracy using the remaining unseen data.

In our first experiments we randomly selected 10 videos of each gesture to train the DHMMs. The remaining 10 are used for testing. In this experiment, 5 states in the HMM are used. An average recognition rate of 97.1% was achieved. The confusion matrix for this experiment is shown in **Table 5.1**. Here we see only 4 gestures caused minor confusion. If we explore these violating gestures obvious similarities in gestures are evident. For example Gesture 6 is confused with Gesture 1, both of these involve the same hand-shape in roughly the same position. Gesture 1 involves moving an open hand across the chest, while Gesture 2 moves a reasonably similar same hand-shape in the same position in a image in a circular motion. This comparison is illustrated in

**Figure 5.5.** It could be argued that the gesture should be identifiable, given the difference in the hand shape. However, if we consider the variation included in the data set, capturing gestures from different people, of different familiarity with system over different sitting, some confusion can be introduced. Removing such confusion could be achieved by introducing some motion information that describes the local motion of the hand; this solution is discussed further in the next chapter. **Figure 5.4** illustrates the vast difference in the motion trajectories for these two gestures.



**Figure 5.4** Motion trajectories for Dynamic gestures



**Table 5.1** Dynamic Gesture Recognition Confusion matrix for a sample Training/Testing set

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	X					2											
2		X					1										
3			X														
4				X													
5					X												
6						X											
7						1	X					1					
8								X									
9									X								
10										X							
11											X						
12												X					
13													X				
14														X			
15															X		
16																X	
17																	X



**Step (1)**

**Step (2)**

**Step (3)**

**Figure 5.5** Similar Dynamic gestures that cause confusion.

We then performed some experiments to find the optimal number of states for our DHMMs. Once again 10 randomly chosen sample were selected for training and the remaining used for testing. The accuracy results are as shown in **Table 5.2**. Here we can clearly see that having 4 states in our DHMMs provides the optimal number of states to model the execution of the gesture.

**Table 5.2** Accuracy rates for different number of states in DHMMs.

<b>Num of States</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>Accuracy %</b>	<b>95.3</b>	<b>97.2</b>	<b>98.0</b>	<b>97.2</b>	<b>97.2</b>	<b>96.5</b>

In order to provide an accurate recognition rate we need to test the accuracy of the system when different random samples of the data set are used for testing and training. We also wish to test the accuracy of the system when a different number of samples are used for training.

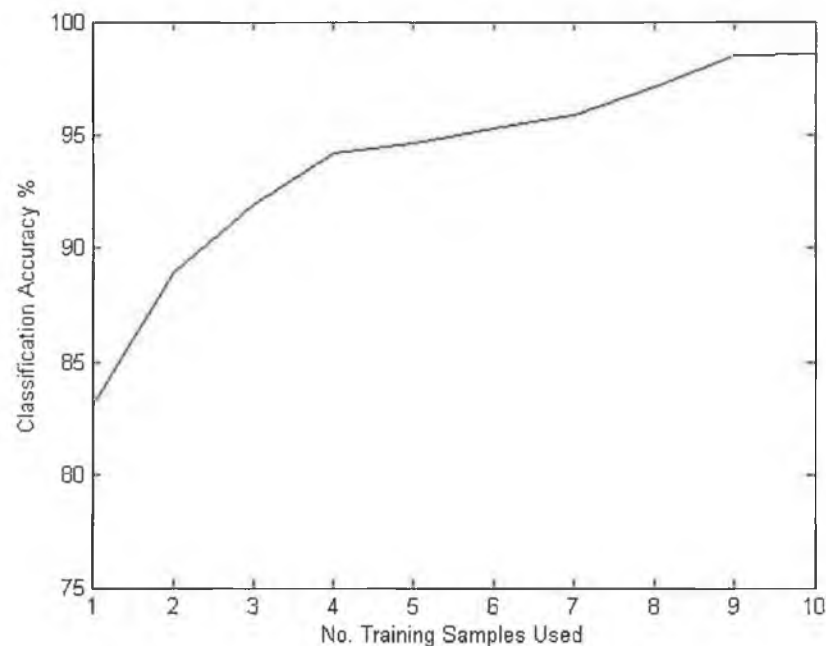
In these experiments five different samples are used to calculate this performance. This means the experiment is run fives times for each testing and training data ratio using different randomly chosen samples for testing and training each time. Recognition accuracy was calculated by computing the average performance for different sampling of the training data.

The performance for each of the different number of data samples used in training are as shown in **Table 5.2** and diagrammatically in **Figure 5.6**. As expected the performance

increases as the number of training samples increases. It is also interesting to note that reasonably high classification rates can be achieved using only one training sample for the DHMM. This classification has been achieved on a standard PC using the Matlab interpreter with non-optimized code in real time at 12 fps.

**Table 5.2.** Illustrates the performance for each of the different number of data samples used for training

No. Training samples	1	2	3	4	5	6	7	8	9	10
Average Performance	83.0	88.9	91.9	94.2	94.6	95.3	95.9	97.1	98.5	98.6



**Figure 5.6** Illustrates the average recognition when a different number of training samples are used.

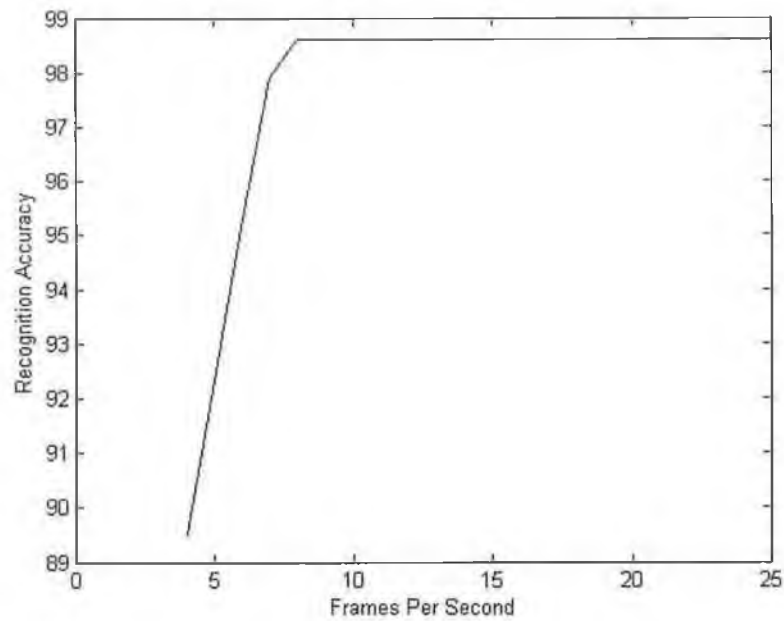
It must be noted that these tests were carried out using videos at a frame rate of 25fps. The DHHMS were both trained and tested using the full 25fps that were captured.

However, by briefly investigating the feature vectors it became apparent that a significant amount of redundancy was present. On average the video samples are approximately two seconds, at a frame rate of 25fps, meaning usually at least 50 frames are captured for each gesture sample. As the gestures chosen are uncomplicated it can be observed in the feature vector that many successive features are identical. This opens the possibility of sampling the feature vector in order to speed up the classification time of the HMMs. Initial experiments proved this was possible while still achieving the comparable recognition rates. The process lead to a marginal performance improvement.

A significant performance increase can be obtained, however, if the sampling process is regressed one step and carried out on the frame sequence. A considerable amount of execution time is employed in processing, segmenting, tracking and classifying each individual frame. By reducing the number of frames to be administered the classification time of a dynamic gesture can be greatly enhanced

We incorporated this notion into our dynamic recognition system in order to improve the application of our system in a real time system. In this experiment the HMMS were trained as before with the full 25fps captured. However, we then tested the accuracy of dynamic gestures while sampling at different frame rates. The results are illustrated in **Figure 5.7**. Here we can see that recognition accuracy remains constant from 25 fps to 8 fps, it only from 8fps that recognition rate begins to decrease. These results are encouraging to further speed up the system. Using 8fps instead of 25fps we can obviously perceive that only one third of the frames need to be processed and this lead to an analogous reduction in processing time.

It must be noted that if different vocabulary of gestures was employed, using more intricate signs from sign language, that are performed at a higher speed, we might need to use a higher sampling rate and maybe even the full 25fps.



**Figure 5.7** Dynamic gesture recognition accuracy when testing the HMMS using different fps samples.

## 5.5 Summary

In this chapter we have presented a dynamic gesture recognition system. We introduced our technique of combining the Subspace recognition technique with a position classifier to train and recognise dynamic hand gestures. A detailed set of experiments was outlined to show the effectiveness of this technique. Average accuracy exceeding 98% was displayed using optimal training and optimal HMM parameters.

# ***CHAPTER 6***

## ***CONCLUSIONS AND FUTURE WORK***

### **6.1 Summary**

In this report a detailed framework is presented for accurate real time gesture recognition. Our novel approach to develop a hand shape classifier trained using computer animation is described along with its application in dynamic gesture recognition. We have developed a real time, multi-user, accurate gesture recognition system. The system uses a single low resolution camera and operates in Matlab on a conventional PC running windows XP. In this work an emphasis was achieving a high speed system that could work in real time with high accuracy.

During the course of this thesis we have described in detail the inspiration and motivation behind our research and its possible applications. A thorough exploration of both current and previous efforts in Gesture recognition was revealed. Once this prelude was given we then offered a thorough description of our system and the technologies incorporated. During the design and implementation an importance was made to keep the system modular. This is to allow future enhancement and will alleviate the complexity of modifying or upgrading the system. Individual components can simply be switched as long they interface with the main system in a similar fashion. While developing this, an effort was made to evaluate each of the individual segments of the system before appraising the whole system.

**Chapter 2** was dedicated to outlining the latest techniques and systems both historically and currently used in gesture recognition. Here we endeavoured to provide an unbiased survey of these techniques with relation to technologies used, accuracy, and vocabulary size. The Poser computer animation and modelling package was also introduced along with some of its benefits and advantages.

In **Chapter 3** a robust hand-shape recognition system has been proposed based on a Subspace Classifier. The subspaces are constructed, with a-priori knowledge, from images acquired using Poser modelling software. One important aspect of this approach is that once the allowable hand-shapes and their bounds have been defined, the set of images of allowable transformations can then be automatically extracted without the expensive need of cameras and actors. This novel method also means our transformation subspaces can be complete and free from outliers allowing for accurate robust recognition. Another important aspect is that the subspaces of 2D images are created from 3D transformations; this further enhances the accuracy of the recognition. The main novelties of this chapter were as presented in [39].

A means to classify images of real hands was then presented in **Chapter 4**. Using image-processing techniques we have shown that accurate recognition is possible for human hands. In this chapter we dealt independently with one handed static gestures using the right hand. It was flagged, however, that the technique could easily be adopted using identical training data by considering the left hand image as a mirror image of a right hand image.

Combining this hand-shape information with the position information, in **Chapter 5**, enabled us to build a dynamic gesture recognition system. A metric for calculating position independently of camera position and distance was disclosed. A model was built for each gesture using DHMMs trained from a number of repetitions of each gesture. An effort was made to include variation in training these DHMMs. Successful classification was achieved for isolated gestures even with limited training. This notion of having a limited training set is quite important to the practicality of a gesture recognition system. It would be advantageous for such a system to be portable, and easily implemented in a target application, without the need to squander hours training the system. We have presented a summary of Chapters 4 and 5 in [40,41].

In summary, we have presented a novel, real-time, dynamic gesture recognition system that incorporates a static gesture recognition system trained using computer animation images. This novel notion of using computer animation to train an appearance based hand gesture recognition system offers many possible future developments. While the vocabulary of the system is modest compared to that of a sign language, we have achieved a high accuracy in high speed. It is hoped that using some of the techniques described in **Section 6.2** can help increase the vocabulary while retaining the high speed and accuracy.

## **6.2 Future Work**

To improve performance over a larger lexicon we intend to introduce a more detailed position gauge, increase the bank of allowable hand-shapes along with adding new features such as hand motion. A simple motion descriptor that could be used is a direction code. This was incorporated in a gesture recognition system developed by Wu



[1]. This was achieved by using the hand position of the current and previous frames, then calculating which of the eight predefined directions the hand is travelling in within the two dimensional image. Bobick [36] introduced another technique for capturing motion information using Motion History image. This technique captures and stores temporal motion information in an image format. A possible disadvantage of this technique would be that each different gesture would have a different motion history image. Another technique our group is currently researching is dividing gestures into gesture subunits. These subunits would represent an isolated recognisable unit of gestures, similar to phonemes in speech recognition. Each of these subunits would have a constant motion pattern which is calculated by investigating the two dimensional motion trajectory. An incorporation of subunits into our gesture recognition procedure should improve the recognition over a larger array of gestures.

One of the other areas we aim to improve is increasing the range of the rotation transforms that can be recognised. Currently a new process is being investigated in order to rotationally align the hand images so that the wrist is located at the bottom of the image. What this will achieve is to ensure that when static hand gesture is performed, that contains a large rotational deviation, it can be rotationally aligned to a recognisable posture suitable for the Subspace Classifier. This technique is being achieved by finding the wrist of the hand using gradient techniques, then aligning the hand accordingly. An added advantage of finding the wrist is that we now also improve the hand segmentation by removing excess arm details from the hand image.

During the course of the work in this thesis we have concentrated on one-handed gestures. An obvious future direction would be to pursue the recognition of two-handed





























gestures. A simple protocol might be to compute the features of each hand separately and enter these into the DHMM.

While the DHMM approach described in this thesis was sufficient for our dynamic gesture recognition system, it is envisaged that a more complex HMM structure would be necessary when considering recognising a larger subset of ISL gestures. CHMMs were mentioned in **Section 5.1**. However, more options exist such as second order HMMs and coupled continuous HMMS (CCHMMs). Much research has been done on HMMs to date. Each of the HMMs techniques would need to be evaluated in order to find the correct variation for any future system.

Any future system that attempts to recognise a significant subset of a sign language would have to incorporate the recognition of non-manual features. Non-manual sign language features include orientation of the lips, eye gaze, frowning and tilting of the head.

With the incorporation of these extra features it is hoped to achieve multi-user recognition over a large vocabulary of hand-shapes and we aspire to combine with a sign language grammar to achieve reasonable recognition for a substantial subset of Irish Sign Language. As part of our future work we intend to test the robustness and effectiveness of both existing and proposed techniques on a larger, more diverse database that would contain images of up to ten different users.

## *APPENDIX A- ISL HANDSHAPES*

A 	B 	C 	D 
E 	F 	G 	H 
I 	K 	L 	M 
N 	O 	P 	Q 
R 	S 	T 	U 
V 	W 	Y 	1 
2 	3 	4 	5 

## ***APPENDIX B – PERFORMING PCA ON A SET OF IMAGES***

Given a set of images

$$X = \{x_1, x_2, x_3, \dots, x_n\}^T \quad (\text{A.1})$$

where  $x_i$  is the  $i^{th}$  image reshaped as an  $M \times 1$  vector of image pixels and

$$\mu_x = E\{x\} \quad (\text{A.2})$$

is the mean of  $X$ , the covariance matrix can be calculated by

$$C_x = E\{(x - \mu_x)(x - \mu_x)^T\} \quad (\text{A.3})$$

We can estimate  $C_x$  by the following equations

$$\hat{C}_x = \frac{1}{P} \sum_{p=1}^P x_p x_p^T - \hat{\mu}_x \hat{\mu}_x^T \quad (\text{A.4})$$

$$\hat{\mu}_x = \frac{1}{P} \sum_{p=1}^P x_p \quad (\text{A.5})$$

Where  $P$  is the length of each vector  $x$ .

Finding the eigenvectors of  $C$  we get  $P$  eigenvectors, each of length  $n$

$$E = \{e_1, e_2, e_3 \dots e_p\} \quad (A.6)$$

To find the Principal components of  $E$ , the eigenvectors are sorted in descending order using the corresponding eigenvalues. Now a subset of  $E$  is retained called  $W$ , where  $w < p$ , and  $W$  represents the eigenvectors of  $E$  with the greatest variance.

$$W = \{e_1, e_2, e_3 \dots e_w\} \quad (A.7)$$

The new feature vector is attained as follows:

$$y = W^T x \quad (A.8)$$

Similarly a test image  $I$  can be projected into the subspace

$$I' = W^T I \quad (A.9)$$

## ***APPENDIX C – DYNAMIC GESTURES***



**Gesture 1** – Right hand moves from right to left, then back to right across the upper chest. Hand maintains a flat 'L' hand shape.



**Gesture 2** – Right hand moves from down to up in front of the body. Hands has an inverted 'A' hand shape.



**Gesture 3** – Right moves from up to down at the side of the body. Hand maintains a flat 'L' hand shape.



**Gesture 4** – Right moves from up to down at the side of the body. Hand maintains an inverted 'D' hand shape.



**Gesture 5** – Right hand swings from left to right in front of the chest. Hand maintains the 'E' hand shape



**Gesture 6** – Right moves from right to left in front of the chest. Hand maintains a 'Thumbs Up' hand shape.



**Gesture 7** – Right hand moves from down to up at the side of the body. Hand maintains the 'W' hand shape.



**Gesture 8** – Right hand moves from right to left across the chest. Hand maintains a rotated 'D' hand shape.



**Gesture 9** – Right hand pivots on the wrist from right to left. Hand maintains a rotated ‘L’ hand shape.



**Gesture 10** – Right hand pivots on the wrist from right to left. Hand maintains a rotated ‘D’ hand shape.



**Gesture 11** – Right moves from right to left in front of the body. Hand maintains a hand shape similar to a rotated ‘L’ with the thumb hidden.



**Gesture 12** – Right hand moves from right and taps on left elbow. Hand maintains a rotated ‘P’ hand shape.





**Gesture 13** – Right hand moves in circular motion in front of the chest. Hand maintains a hand shape similar to a rotated ‘L’ with the thumb hidden



**Gesture 14** – Right tilts from up to down at the side of the body. Hand maintains an ‘L’ hand shape.



**Gesture 15** – Right tilts from up to down at the side of the body. Hand maintains a ‘D’ hand shape.



**Gesture 16** – Hand moves from up to down at the side of the body. Hand maintains a ‘D’ hand shape.



**Gesture 17** – Right hand moves from down to up at the side of the body. Hand maintains the ‘W’ hand shape.

## ***APPENDIX D – CODE LISTING***

## ***REFERENCES***

- [1] H. Wu, "Gesture Recognition Using Principle Component Analysis, Multi-Scale Theory, and Hidden Markov Models", *PHD Thesis*, Dublin City University 2002.
- [2] W. Zhao, "Improving the robustness of subspace FR system" *2<sup>nd</sup> Int. Conf. Audio and Video Based Person Authentication, Washington DC*, pp 78-83, 1999.
- [3] Atid Shamaie and Alistair Sutherland, "Accurate Recognition of Large Number of Hand Gestures," *2nd Iranian Conference on Machine Vision and Image Processing*, K.N. Toosi University of Technology, Tehran, Iran, 2003.
- [4] Feng-Sheng Chen, Chih-Ming Fu, Chung-Lin Huang, "Hand Gesture Recognition Using a Real-time Tracking Method and Hidden Markov Models", *Image and Vision Computing*, Vol 21, pages 745-758, 2003.
- [5] T. Kadir, R. Bowden, E. J. Ong, A. Zisserman, "Minimal Training, Large Lexicon, Unconstrained Sign Language Recognition", *Proceedings of the 15th British Machine Vision Conference, Kingston*, 2004.
- [6] J. M. S. Dias, P.Nande, N. Barata, A. Correia, "O.G.R.E – Open Gestures Recognition Engine" *Procedings of the 17<sup>th</sup> Brazilian Symposium on computer Graphics and Image Processing 2004 (SIBGRAPI'04)*, 2004.
- [7] R. Vidal, Y. Ma, S. Sastry, "Generalized Principle Component Analysis (GPCA)" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 27, No 12, 2005.
- [8] H. Bishoff, A. Leonardis, J. Maver, "Multiple Eigenspaces", *Pattern Recognition*, Vol 35, pp 2613-2627, 2002.

- [9] P. Y. Simard, Y. A. Le Cun, J. S. Denker, B Victorri, "Transformation Invariance in Pattern Recognition – Tangent Distance and Tangent Propagation", Proc. Neural Networks: Tricks of the trade, Lecture Notes in Computer Science, vol. 1524, pp 239-274, Springer Verlag, 1998.
- [10] Curious Labs, [Online], Available: <http://www.e-frontier.com/>, (Accessed 9 December 2005).
- [11] D. Keysers Tangent Distance Implementation, [Online], Available: <http://www-i6.informatik.rwth-aachen.de/~keysers/td/>, (Accessed 9 December 2005).
- [12] L Gupta, S. Ma, "Gesture-Based Interaction and Communication: Automated Classification of Gesture Contours", IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and reviews, Vol 31, No 1, 2001
- [13] K. S. Patwardhan, S. Dutta Roy, "Hand gesture modeling and recognition involving changing shapes and trajectories, using a Predictive EigenTracker, Pattern Recognition, (Article in Press), 2006.
- [14] J. Carreira, P. Peixoto, "A Vision Based Interface for Local Collaborative Music Synthesis, in Proc. IEEE International Conference on Automatic Face and Gesture, 2006
- [15] Y. Yuan, K. Barner, "An Active Shape Model Based Tactile Hand-shape Recognition with Support Vector Machine".
- [16] S.C.W Ong, S. Ranganath " Automatic Sign Language Analysis: A Survey and the future beyond Lexical Meaning", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 27, No 6, 2005.
- [17] J. Triesch, C. von der Malsburg, "Classification of hand postures against complex backgrounds using elastic graph matching", Image and Vision Computing Vol 20, pp 937-943, 2002.

- [18] Eng-Jon Ong, R. Bowden, "A Boosted Classifier Tree for Hand-shape Detection", in Proc. IEEE International Conference on Face and Gesture, pp 889-894, 2004.
- [19] Z. Zhang, M. Li, S. Li, H. Zhang. Multi-view face detection with floatboost. Proc of the Sixth IEEE Workshop on Applications of Computer Vision.
- [20] A. Just, Y. Rodriguez, S. Marcel, "Hand Posture Classification and Recognition using the Modified Census Transform", in Proc. IEEE International Conference on Face and Gesture, pp 351-356, 2006.
- [21] Y. Wu, T. S. Huang, "View-independent Recognition of Hand Postures", In Proc. IEEE Conference on Computer Vision Pattern Recognition, Vol 2, pp 88-94, 2000.
- [22] A. Shamaie, "Hand Tracking and Bimanual Movement Understanding", *PHD Thesis*, Dublin City University 2003.
- [23] G. Awad, J. Han, A. Sutherland, "A Unified Method for Segmentation and Tracking of Face and Hands in Sign Language Recognition", in Proc 18<sup>th</sup> International Conference on Pattern Recognition, 2006.
- [24] Gahramani, Z. "An Introduction to Hidden Markov Models and Bayesian Networks, International Journal of Pattern Recognition and Artificial Intelligence, vol 15, pp 9 – 42, 2001
- [25] T. Kadir, R. Bowden, E. J. Ong, A. Zisserman, "Minimal Training, Large Lexicon, Unconstrained Sign Language Recognition, In Proc BMVC'04, Vol 2, pp849-858, 2004.
- [26] W. T. Freeman and C. Weissman, "Television control by hand gestures", International Workshop on Automatic Face- and Gesture- Recognition, IEEE Computer Society, Zurich, Switzerland, June, 1995, pp. 179--183.

- [27] Graetzel, C., Grange, S., Fong, T., and Baur, C., "A Non-Contact Mouse for Surgeon-Computer Interaction", IEEE Medical Image Computing and Computer Assisted Intervention, Toronto, Canada interface, November 2003
- [28] Tong, X.F., Duan, L., Xu, C., Tian, Q., Lu, H., "Local Motion Analysis and it's Application in Video Based Swimming Style Recognition". International Conference on Pattern Recognition, 2006.
- [29] Guo, F., Qian, G., "Dance Posture Recognition Using Wide-Baseline Orthogonal Stereo Cameras", in Proc. IEEE International Conference on Face and Gesture, pp 481-486, 2006.
- [30] Chai, Y., Ren, J., Zhao, R., Jia, J., "Automatic Gait Recognition Using Dynamic Variance Features", in Proc. IEEE International Conference on Face and Gesture, pp 475-480, 2006.
- [31] Starner, T., "Visual Recognition of American Sign Language Using Hidden Markov Models", Masters thesis, MIT Media Lab, USA, 1995.
- [32] Vogler, C., Metaxaz, D., "ASL Recognition Based on a Coupling Between HMMs and 3D motion Analysis", in Proceedings of IEEE International Conference on Computer Vision, Bobbay, India, 1998, 363-369.
- [33] Huang, C-L., Wu, M-S., Jeng, S-H., "Gesture Recognition using the multi-PDM method and Hidden Markov Model", Image and Vision Computing, Vol 18, pp 865-879, 2000.
- [34] Bunke, H., Caelli T., "Hidden Markov Models – Applications in computer Vision", World Scientific Publishing, Singapore, 2001.
- [35] Rabiner, L., R., Juang, B., H., "Fundamentals of Speech Recognition", PTR Prentice Hall, New Jersey, USA, 1993.





- [36] A. F. Bobick and J. W. Davis. "Real-time recognition of activity using temporal templates". In Proceedings of Workshop on Applications of Computer Vision 1996, 1996.
- [37] Kohler M. R. J. (1997). "System Architecture and Techniques for Gesture Recognition in Unconstraint Environments", International Conference Virtual Systems and MultiMedia, Geneva, 1997.
- [38] Isard M., Blake A. (1998). "CONDENSATION- Conditional Density Propagation for Visual Tracking", International Journal of Computer Vision, Vol. 29, No. 1, pp. 5-28, 1998.

#### **Publications Arising From This Thesis**

- [39] T. Coogan and A. Sutherland. "Transformation Invariance in Hand-shape Recognition". In proceeding of International Conference of Pattern Recognition, ICPR, 2006.
- [40] Coogan T., Awad G., Han J. and Sutherland A. (2006). "Real Time Hand Gesture Recognition Including Hand Segmentation and Tracking", In proceedings of International Symposium on Computer Vision, ISVC06, LNCS, 2006.
- [41] Coogan T., Awad G., Han J. and Sutherland A. (2006). "Real Time Hand Gesture Recognition Including Hand Segmentation and Tracking", demonstration at European Conference of Computer Vision, ECCV, 2006.