Hand Tracking and Bimanual Movement Understanding

by

Atid Shamaie

A thesis submitted in partial fulfilment of the requirements for the degree
of
Doctor of Philosophy

at the

School of Computing

Dublin City University

Academic Supervisor

Dr Alistair Sutherland

2003

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of PhD 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 AHd SL (Candidate)

ID No 99145863

Date 22 SEPTEMBER 2003

In the name of God

به نام خداوند جسم و جان

عشق شاهی

تابستان ۱۳۸۲

Although my senses were searching the desert fanguelessly
discovering nothing although finding a lot
my soul was illuminated by a thousand suns
but could never ever touch the perfection of a single atom

Avicenna 10th century

ACKNOWLEDGEMENT

I would like to thank my supervisor, Dr Alistair Sutherland for his guidance over the course of my working on this thesis and preparing my publications

I would also like to thank Professor Alan Smeaton for buying us a CCD Camera, which was used for almost all the experiments in this thesis I would like to thank Dr John McKenna for his occasional advice on Hidden Markov Models and Kalman Filter

I would like to thank Dr S Mehdi Fakhraie of the University of Tehran his advice was a light in the darkness, especially when things were going wrong

I must thank my wife, Saba for being beside me and for her patience when I spent most of my time in the school working on this research project

I would also like to thank some researchers from other universities, particularly Dr Franz Mechsner of the Max Planck Institute for Psychological Research, Professor Richard Ivry of the University of California, Berkeley, Dr Arash Fazl of the University of Boston and Dr Hadi Fallahnejad with whom I was in occasional contact on neuroscience subjects

I would like to thank Dr Hai Wu for giving me some of his C codes when I started this research project

I would like to thank my friends in the School of Computing, particularly Niall, Tom, Dalen, Damir, Paul, Cathal, Yaw, Karl, Hyowon, and Michelle who were beside me in the school

I would like to express my deepest appreciation to my parents and my brothers who, regardless of the thousands of miles distance between us, were in my mind and beside me when I needed them

As this is probably the last educational degree I can earn, I would like to go back to all those years of study and thank all the teachers, lecturers and professors in the schools and universities who helped me to get to this point in my life

ABSTRACT

Bimanual movements are a subset of human movements in which the two hands move together in order to do a task or imply a meaning. A bimanual movement appearing in a sequence of images must be understood in order to enable computers to interact with humans in a natural way. This problem includes two main phases, hand tracking and movement recognition.

We approach the problem of hand tracking from a neuroscience point of view. First the hands are extracted and labelled by colour detection and blob analysis algorithms. In the presence of the two hands one hand may occlude the other occasionally. Therefore, hand occlusions must be detected in an image sequence. A dynamic model is proposed to model the movement of each hand separately. Using this model in a Kalman filtering process the exact starting and end points of hand occlusions are detected. We exploit neuroscience phenomena to understand the behaviour of the hands during occlusion periods. Based on this, we propose a general hand tracking algorithm to track and reacquire the hands over a movement including hand occlusion. The advantages of the algorithm and its generality are demonstrated in the experiments.

In order to recognise the movements first we recognise the movement of a hand. Using statistical pattern recognition methods (such as Principal Component Analysis and Nearest Neighbour) the static shape of each hand appearing in an image is recognised. A Graph-Matching algorithm and Discrete Hidden Markov Models (DHMM) as two spatio-temporal pattern recognition techniques are investigated for recognising a dynamic hand gesture.

For recognising bimanual movements we consider two general forms of these movements, single and concatenated periodic. We introduce three Bayesian networks for recognising the movements. The networks are designed to recognise and combine the gestures of the hands in order to understand the whole movement. Experiments on different types of movement demonstrate the advantages and disadvantages of each network.

TABLE OF CONTENT

Chapter 1

INTRODUCTION

Everyday, millions of people, cars, animals, and many other subjects around the world move in order to do their tasks. A subset of these movements is human movements. A man walks in order to reach a place, moves the hands to take an object, point to somewhere, show the size of an object, and imply a meaning. He/she talks, laughs, cries, etc. in order to convey meanings, and emotions, in order to transfer information to others, to voice his/her needs, etc.

Understanding all these movements requires a huge amount of knowledge that people learn everyday from the very first day of their life. We look at objects and people to understand them, to understand their motion, their purpose, and their emotion. Looking at all these movements and understanding them give the power of understanding the environment to us. By understanding the environment we arrange our needs and tasks, problems and programmes.

Computers have provided us with a better quality of life. Equipping computers with the knowledge to understand the environment is a target of much research in computer science, as we will show in Chapter 2. Particularly, getting computers to look at the environment in order to analyse it in the same way as we do is the basis of research in a wide spectrum of scientific, engineering, and cognitive research projects. Equipping computers with this knowledge provides us with better ways to interact with computers.

Interacting with computers have been traditionally based on artificial devices like keypads and switches. With the current research in *Artificial Intelligence*, computers are able to understand a large part of their environment. The new means of interactions have opened new horizons in better involving these devices in our daily life.

We investigate the subject of *Human Computer Interaction* (HCI) from an engineering and cognitive science point of view. Different aspects of HCI will be briefly reviewed. We study

the problems involved in *looking at people* by computers for Human Computer Interaction Artificial visual systems are introduced and studied in different parts of this thesis We use a wide range of techniques and methods from engineering to neuroscience in order to understand a group of movements The integration of all these methods, techniques and phenomena enables us to develop an integrated solution to an important and crucial problem in Human Computer Interaction

Hand gestures in linguistic communication (e g sign languages) and paralinguistic communication (e g reinforcement gestures) are used in HCI to communicate with computers In this thesis we aim to recognise a set of bimanual gestures

1 1 Human Computer Interaction

"Human Computer Interaction (HCI) entails the study of physical, social, cognitive and engineering aspects of designing information technology for ease of use [Salvendy 2001]" These aspects are illustrated by the author in Figure 1 1 In HCI a wide range of applications is being developed in order to provide easier means of interacting with computers than traditional keyboards and mice Smart environments, wearable computers and perceptual user interfaces are considered as the "fourth generation" of computing and information technology [Pentland 2000]

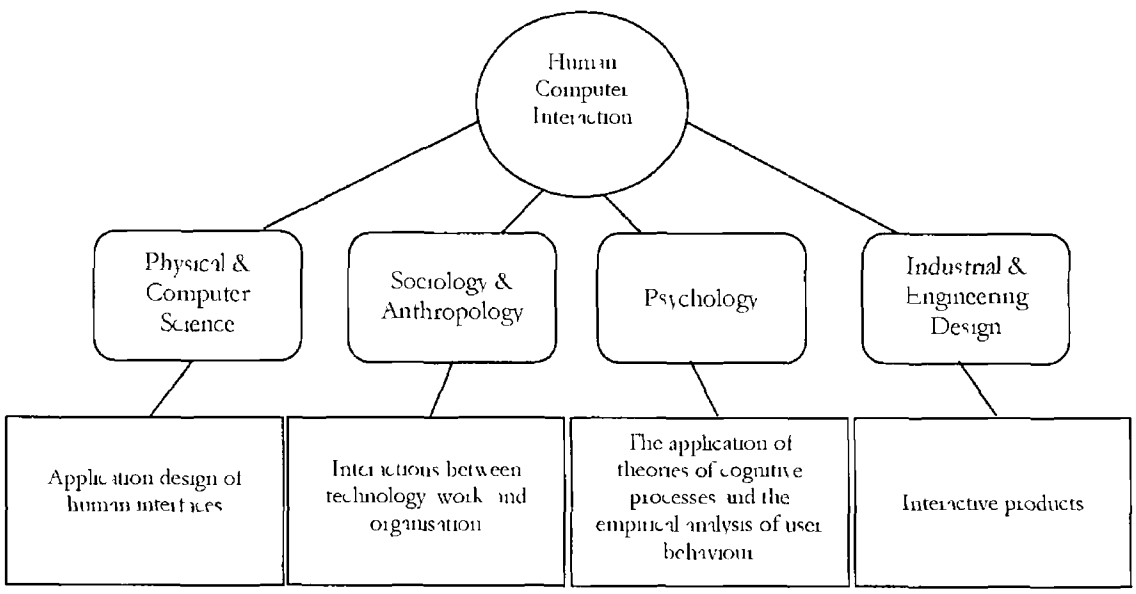


Figure 1 1 Human Computer Interaction aspects

Sensing technology that allows computers to be used without detailed instructions is the main technology that this new generation of computers should be equipped with Enough

knowledge of people should be provided for computers to act appropriately with the minimum of detailed traditional instructions [Pentland 2000]

Many sensing technologies are employed to improve the interactions and avoid explicit instructions Audio and visual techniques are some of the most common methods Voice and speech recognition techniques enable computers to communicate with people by using human languages (see Figure 1 2)

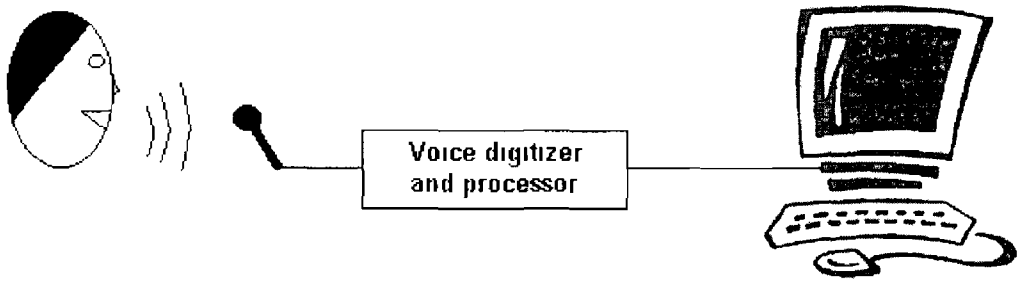


Figure 1 2 A system of voice recognition

Visual sensing is another technique, which is used by humans as one of the five senses Getting a machine to do the visual tasks of a human is the subject of the current research in artificial visual sensing Even for some simple tasks that a human eye can do, large difficulties in mechanisation are involved

1 2 Visual Sensing

By using visual sensors physical connections like wires to transfer information into computers are removed Bar codes are a well-known example for connecting information on physical packages of goods to a database Understanding information within an image employs a large number of algorithms

1 3 Machine Vision

Partially providing the ability of human vision for computers is known as machine vision [Davies 1997]

Machine vision either from the engineering and technological or the theoretical point of view is an important research area By looking at scenes, objects, colours and movements machine vision provides computers with the ability of understanding environment The domain of machine vision includes a wide spectrum of science and technology such as computer science and engineering, signal processing, physics, statistics and applied mathematics, neuroscience and cognitive sciences

The components of a visual system include image signal processors, colour modelling, geometrical processors, high-speed computers, artificial intelligence techniques, mathematical models, programming techniques, etc

1 3 1 Machine Vision Aspects

By machine vision we refer to a wide range of methods from low-level image processing to real-time motion analysis and pattern recognition

In low-level image processing some basic operations on either binary, grey-scale or colour images are done for purposes like noise suppression, edge detection (see Figure 1 3), image filtering and masking, etc This level of processing also includes object locating via edge detection, binary shape analysis and boundary pattern analysis

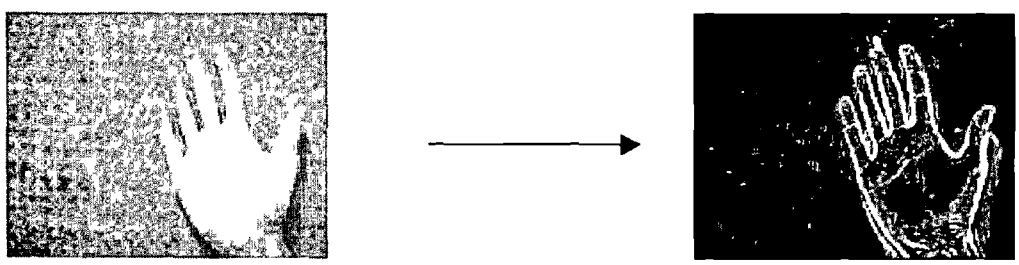


Figure 1 3 Edge detection

An intermediate level of processing includes geometrical shape analysis, line and curve detection, circle and ellipse detection [Shamaie 1997] (see Figure 1 4), hole detection, polygon and corner detection

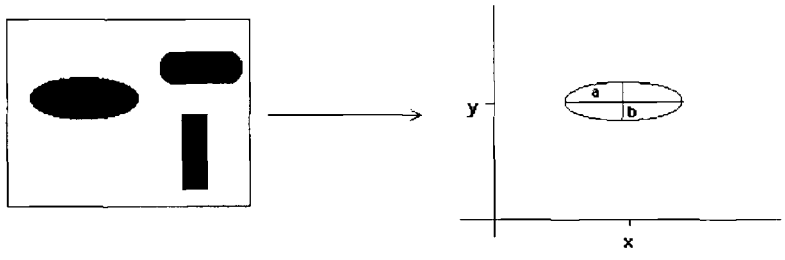


Figure 1 4 Ellipse detection and parameter extraction

Many different industrial and medical applications use this level of processing For example, in industrial and manufacturing environments locating a circular or elliptical object, which can be handled by a robot, is one of the basic problems Different aspects of this problem are recognition of an elliptical (or circular) shape, extraction of the ellipse parameters and location estimation

For the set of well-known geometrical shapes like lines and circles, many different techniques have been proposed in the literature such as Hough Transform [Illingworth 1990], adaptive fuzzy C-Shell clustering [Dave 1992], Fourier Descriptors [Persoon 1977], polygon estimation [Pavlidis 1977] and numerical optimisations [Shamaie 1999]

The highest level of processing which is called application-level includes abstract pattern matching, three-dimensional environments, motion analysis, texture analysis and pattern recognition At this level more natural things enter into the processing Natural three-dimensional objects and environments, motion analysis and matching, and recognition of natural objects with unknown geometrical shapes are the problems that employ not only image processing techniques but also artificial intelligence methods

1 3 2 Human Recognition

Recognition of different parts of a human body, analysis and information extraction are the tasks at the application-level processing Face and gesture recognition, 3D person tracking and behaviour understanding have been mainly addressed in the literature as the important problems in looking at people by machine vision systems [Pentland 2000] In this area there is no complete analytical models for the parts of human body This makes it difficult to find a totally analytical algorithm to recognise the parts and their features Therefore, many

statistical algorithms are proposed to deal with this problem In Chapter 2 we will briefly discuss these methods The main applications of this area are person identification via face recognition, hand and body gesture recognition to interact with computers, and surveillance and monitoring of human behaviour for security reasons [Pentland 2000]

Since the main focus of this thesis is on hand gesture recognition, we discuss this problem more and for the other problems we refer the reader to the references

1 4 Machine Learning and Gesture Recognition

Hand gestures are very common in social interaction People usually use hand gestures to explain their speech and internal emotions Speech reinforcements, showing directions, and showing the size of an object are usual gestures However, a large community of people uses hand gestures for very basic communication Deaf people use sign language to communicate with each other Sign languages contain a large number of hand gestures These gestures have a set of defined static shapes and hand movements The number of recognisable static hand shapes is limited to less than 100 So, in each sign language (e g Irish or American Sign Language), normally, a static hand shape is defined for every letter in the alphabet (see Figure 1 5) However, for the three letters 'j', 'x' and 'z' movements of hand are defined in Irish Sign Language Although most of the words in sign languages are dynamic and start with a defined static shape of hand and continue as a movement, these static signs are useful mainly for spelling the words not defined in a language vocabulary like names This is called finger spelling

Learning the hand shapes and recognising them is the first part of the research in this thesis Recognition of a hand gesture continues by the recognition of hand movements and changes in the shape of hand For this, a system needs to learn the movements to be able to classify and recognise them Many machine learning techniques have been proposed for spatio-temporal learning and recognition of hand gestures

All these are the pre-requisites to an important problem that has not been yet specifically worked on before

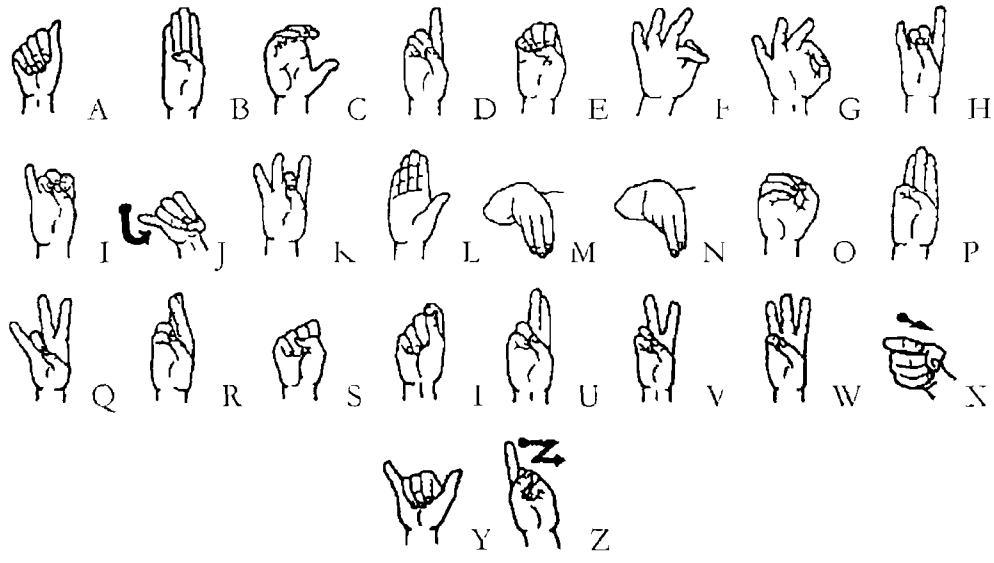


Figure 1 5 Irish Sign Language – static hand shapes

A large group of movements is called *bimanual movements* In these movements both hands move in order to do a task or imply a meaning Clapping, opening a bottle, typing on keyboard, eating with fork and knife, drumming, guiding a pilot driving an aeroplane into parking, etc are some of the usual bimanual movements (see Figure 1 6)

Another group of bimanual movements are the movements in sign languages Particularly, British Sign Language is one of the sign languages in which the two hands are used for most of the gestures, even the alphabet (see Figure 1 7)

Learning bimanual movements by a machine is different from unimanual movements Due to the fact that the both hands move simultaneously in order to do one thing or imply one meaning, a system should analyse the movement of the hands together in order to understand the whole movement In this order, many problems arise but the main is occlusion Since, in a bimanual movement, one hand may cover the other for some moments recovering the movement of each hand and analysing it is difficult Many problems like detecting occlusion and tracking the hands before and after occlusion are the important parts of this issue

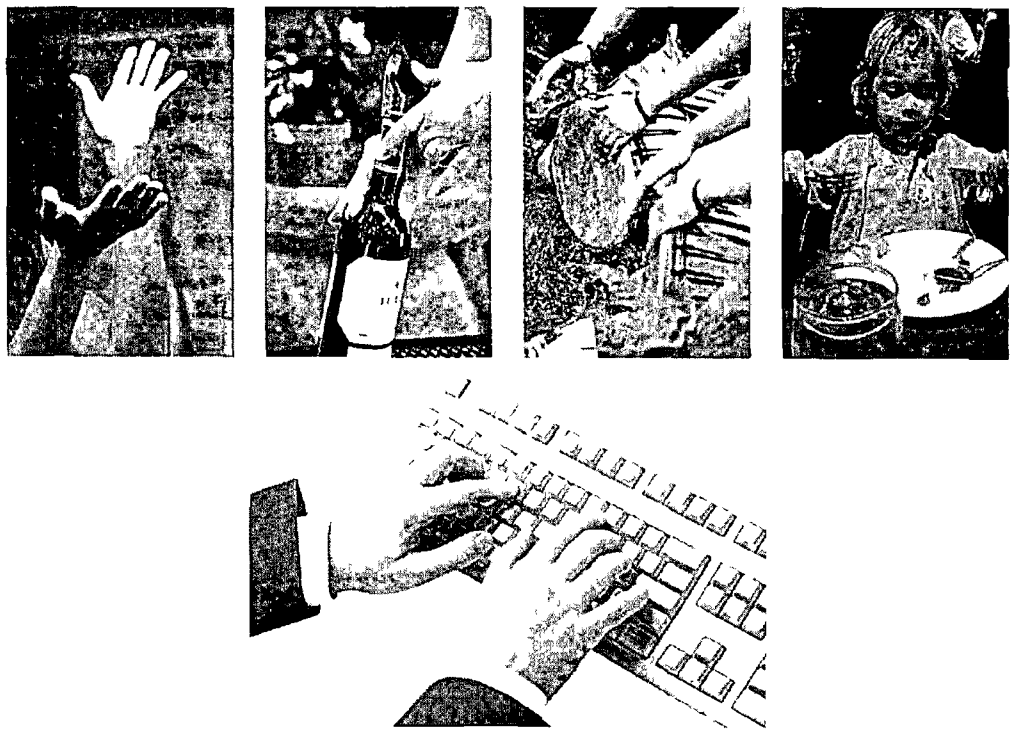


Figure 1 6 Examples of human movements

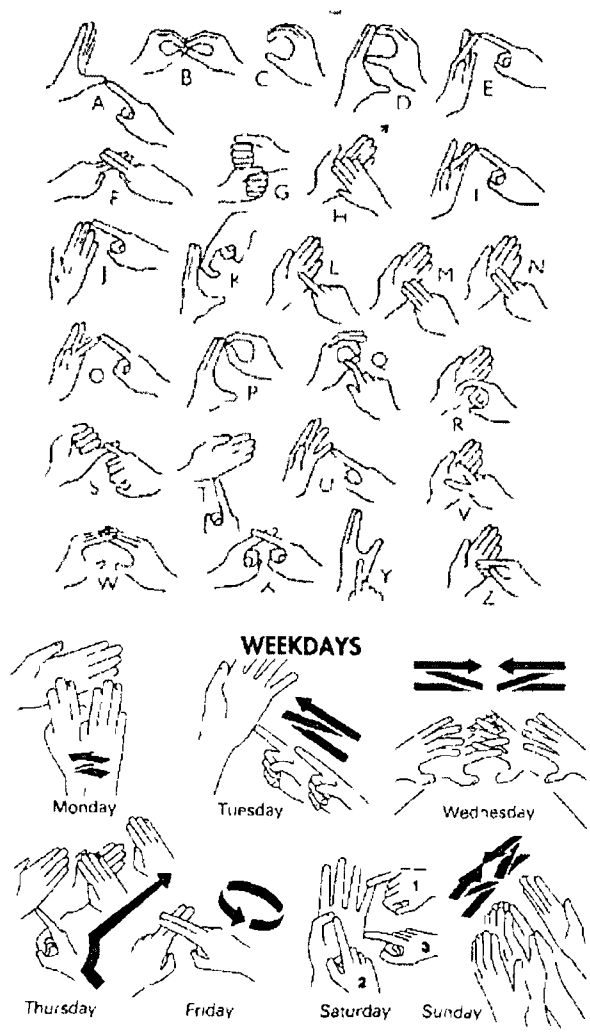


Figure 1 7 Alphabet and weekdays in British Sign Language

The other problem is to synchronously analyse the movements of the hands and combine them in order to understand the whole movement

In this thesis, we aim to recognise hand shapes and gestures, and track the hands in order to understand bimanual movements We start from the basic problems of static hand shape recognition When the static hand shapes are recognised we do temporal recognition of hand shapes in order to recognise a hand movement If we can recognise a single-hand movement, we can involve the movement of the other hand For this we first need a method to track the two hands separately, detect occlusions and resume tracking the hands correctly after occlusion Given that the hands are tracked correctly how can we recognise the whole movement? In other words, how can we combine the individual movements of the two hands to recognise the whole movement? What should we do with the parts of the movement where one hand occludes the other? How can a machine learn to deal with hand occlusions as well as the other parts of a movement? We will answer all these questions in this thesis Techniques, methods and algorithms will be introduced for different parts of the problem of bimanual movement recognition At each part we try to introduce solutions as general as possible In other words, we try not to limit the solutions to a few number of restricted cases

In the next chapter we state the main problem of this research and briefly explain about different proposed methods involved directly or indirectly in gesture recognition In Chapter 3, some techniques for static hand shape recognition are presented and an application of these methods in a real time mouse simulator is introduced In Chapter 4, the problem of dynamic hand gesture recognition is discussed We introduce a new technique for recognising dynamic hand gestures Chapter 5 is dedicated to a dynamic model, which will be used in our tracking algorithm for detection of hand occlusions and tracking hands in bimanual movements

In Chapter 6, Hidden Markov Models and their application in dynamic gesture recognition are discussed In this chapter we compare our proposed gesture recognition technique with the Hidden Markov Models for single-hand gestures and discuss the advantages and disadvantages of each technique In Chapter 7, the problem of occlusion detection and hand tracking in bimanual movements is discussed We introduce a new algorithm for detecting occlusion, tracking the hands and re-acquiring them at the end of occlusion parts of a movement Chapter 8 is dedicated to a new a technique for combining the movements of the

two hands and dealing with the occlusion parts of a movement A technique is introduced to segment a bimanual movement A new Bayesian network is proposed for combining the hand movements and dealing with occlusion parts simultaneously

At the end of each chapter we present some experimental results to demonstrate the performance of our proposed techniques and algorithms Conclusions and potential future work will be presented at the end of the thesis

Chapter 2

PROBLEM STATEMENT AND LITERATURE REVIEW

A visual sensor such as a camera captures what visually is observable in a scene The captured images constituting a sequence represent a temporal event By using a camera connected to a computer via an interface an image or a sequence of images are transferred to the computer (see Figure 2 1) The interfaces are called frame grabbers This type of data acquisition is called image acquisition "Image acquisition is concerned with the task of interrogating the scene under consideration with some energy sources, and subsequently sensing the returned energy, which has been modulated by interaction with elements of the scene" [Ellis 2001] Usually a visual sensor converts the received energy to electrical signals These signals enter the digitizer and frame grabber to be digitised and stored in the memory of the computer

A hand gesture can be captured as a temporal movement in a sequence of images

In this chapter we state the main problem to be solved in the thesis A review of the related works is presented We survey a wide spectrum of techniques, methods and algorithms presented in the literature for static and dynamic pattern recognition Basic statistical pattern recognition techniques to advanced temporal pattern matching algorithms are briefly presented in this chapter

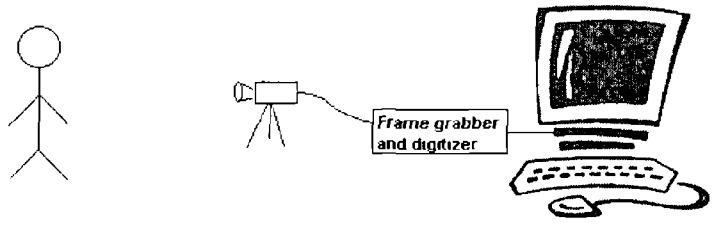


Figure 2 1 A visual system

2 1 Hand Gestures

A hand gesture is a movement of hand to imply a meaning, represent a word or sentence, or show an emotion In order to recognise a hand gesture by a visual system, a sequence of images containing the gesture is captured Every frame of this sequence contains a hand shape, which is to be analysed However, first, one should extract the hand from the background of the image

A very common technique for extraction is colour detection and segmentation A colour segmentation procedure should identify the principal object colour and separate it from the background Many different colour segmentation algorithms are addressed in the literature [Jain 1989]

We use a simple grey-scale detection algorithm, in which every pixel above a certain pre-defined threshold of grey levels is considered as a point on the principal object This algorithm extracts the hand from the background (see Figure 2 2) Since we are using a grey-scale camera our colour detection algorithm is based on the grey levels

Now, for a sequence of images of an extracted hand the main problem of this research can be stated as given in the next section

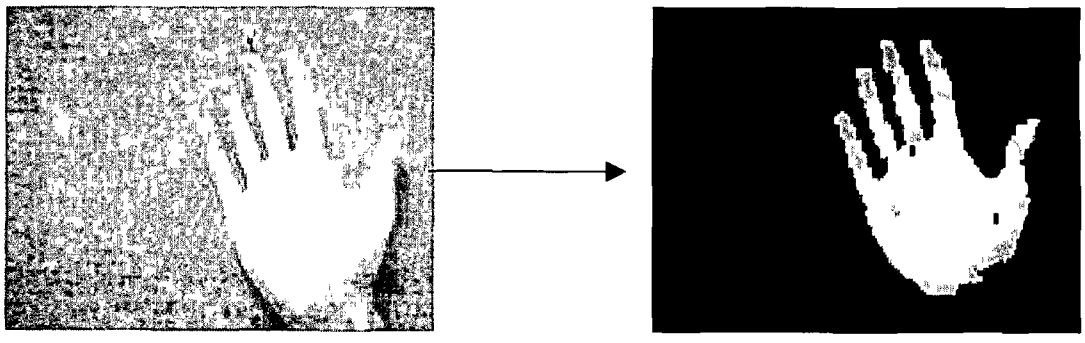


Figure 2 2 Colour detection and segmentation

2 2 Main Problem

In this research we are going to solve the following problem

Given a bimanual movement including hand occlusion find a movement in a given database of movements that best matches with it spatially and temporally In other words recognise the input bimanual movement given that a set of predefined bimanual movements is provided

Static hand shape recognition, dynamic hand gesture recognition, occlusion detection, hand tracking and bimanual movement recognition are the main parts of this problem

2 3 Approaches to Hand Gesture Recognition

A general view of hand gesture recognition has been presented in [Pavlovic 1997] Hand gesture recognition is divided into two parts static hand shape analysis and dynamic gesture analysis

2 3 1 Static Shape Analysis

Static shape analysis has been widely addressed in the literature [Theodoridis 1999] as a pattern recognition problem Statistical pattern recognition and artificial neural networks are some well-known approaches In statistical pattern recognition, different techniques have been discussed for classification By classification we mean classifying a shape into one set of predefined classes Therefore, every shape must be represented by its features that distinguish it from the other shapes

By selecting a set of features of the shape, a space is formed in which every axis is represented by one of the selected features This space is called a feature space In image analysis the features are sometimes the pixel values In the feature space every shape has a particular position distinguishing it from the other shapes

- **Nearest Neighbour**

In order to classify an unknown shape, one can find the position of the shape in the feature space Then by finding the nearest point in this space representing a predefined class, the unknown shape can be identified (see Figure 2 3) This is known as the Nearest Neighbour algorithm

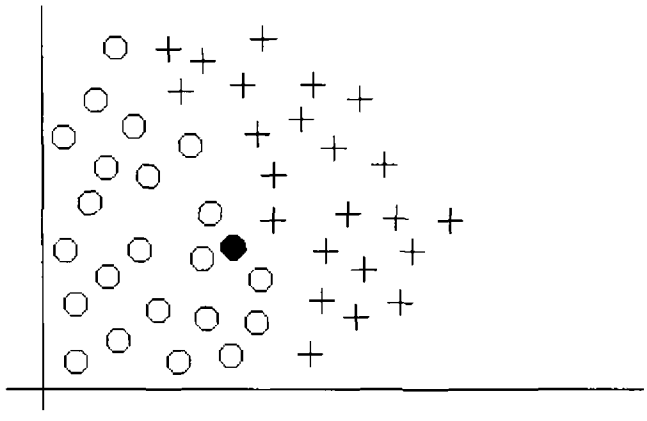


Figure 2 3 Principle of the Nearest Neighbour algorithm for a two class problem

The predefined classes are represented by a set of known shapes, which is used for training the system This set is also known as the training set

- **Bayes' Theorem**

Another important method in statistical pattern recognition is Bayes' decision theory In this theory, for a single feature x, the probability of x being in the class C_i is defined as

$$P(C_i \mid x) = \frac{P(x \mid C_i)\,P(C_i)}{P(x)} \qquad (2\,1)$$

where

$$p(x) = \sum_j p(x \mid C_j)\,P(C_j)$$

Mathematically, the variables are

$P(C_i)$ *a priori* probability of class C_i,

$P(x)$ probability density for feature x,

$P(x \mid C_i)$ class conditional probability density for feature x in class C_i, i e the probability that feature x arises for objects known to be in class C_i,

$P(C_i \mid x)$ the *a posteriori* probability of class C_i when x is observed

In a real-world problem, the number of features is usually more than one Bayes' rule can be generalised to cover the case that feature space is a multidimensional space [Davies 1997]

Now the classification procedure is to compare the values of the probabilities $P(C_j \mid x)$ and to classify an object as class C_i if

$$P(C_i \mid x) > P(C_j \mid x) \qquad \forall j \neq i \tag{2.2}$$

Bayes' rule is the base of other recognition techniques which will be explained in the next chapters

- **Cluster Analysis**

As the data in the training set is located in the feature space, many clusters of data are found based on the similarities among the data In a set of data, those with the same features are to be identified as a single point in the feature space However, small variations in the features make the points very close but separated Therefore, a set of data with almost the same features forms a cluster of points in the feature space (see Figure 2 4) The variation in the features are normally because of noise or other sources of variations like rotation and translation

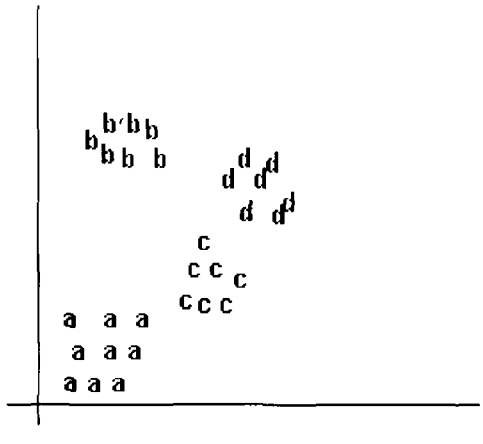


Figure 2 4 Clusters in a two dimensional feature space

For an object with feature x in the space one can find the nearest cluster and classify x as that class Identifying the nearest centre of gravity of each cluster, is one possible way of finding the nearest cluster assuming the clusters are roughly spherical

- **Principal Component Analysis**

Principal Component Analysis (PCA) is usually used to reduce the dimensionality of data For example, in an image of 32 × 32 pixels there are 1024 pixels and every pixel represents a feature in a 1024 dimensional feature space Working in a 1024-dimensional space needs intensive computational power But, usually, a large number of features do not carry useful information Therefore, selecting the features containing the most useful information is very important By using PCA, one can extract the features with useful information and eliminate the rest This technique involves finding the mean of a cluster of points in feature space, and then finding the principal axes of the cluster

First an axis is found that passes through the mean of data points and which gives the maximum variance when the data are projected onto it Then the second axis with the same specifications is found in a direction normal to the first This process continues until N principal axes are found The N principal axes form the new feature space Mathematically, if a matrix of observations (images) is given by

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \quad, \mathbf{x}_n\}^T$$

where $\mathbf{x}_i$ is the i^{th} image reshaped as an $M\times 1$ vector of image pixels in the sequence and

$$\mu_x = \mathrm{E}\{\mathbf{x}\}$$

is the mean of the observations, the covariance matrix of the data can be calculated by

$$\mathbf{C}_x = \mathrm{E}\{(\mathbf{x}-\mu_x)(\mathbf{x}-\mu_x)^T\} \tag{2 3}$$

We can estimate $\mathbf{C}_x$ by the following equations

$$\hat{\mathbf{C}}_x = \frac{1}{P}\sum_{p=1}^{P}\mathbf{x}_p\mathbf{x}_p^T - \hat{\mu}_x\hat{\mu}_x^T \tag{2 4}$$

$$\hat{\mu}_x = \frac{1}{P}\sum_{p=1}^{P}\mathbf{x}_p \tag{2 5}$$

where P is the length of each vector $\mathbf{x}$

The eigenvectors of the covariance matrix C form the new feature space called the eigenspace. However, the number of eigenvalues and eigenvectors of the matrix C is the same as the number of dimensions in the feature space. One should select fewer eigenvectors that represent the best features and the most valuable data. The eigenvectors corresponding to the largest eigenvalues show the directions in which the data have the largest variations. Therefore, a few eigenvectors are selected and a lower dimensional feature space is formed.

By projecting a data vector x_i onto the feature space, we get

$$\mathbf{y}_i = \mathbf{A}(\mathbf{x}_i - \mu_x) \tag{2.6}$$

where $\mathbf{A}$ is the matrix of the selected eigenvectors and $\mathbf{y}_i$ is a point in the coordinate system defined by the eigenvectors. This point is the data vector $\mathbf{x}_i$ in a lower dimensional space that contains the most valuable information in $\mathbf{x}_i$.

In order to reduce the dimensionality of our images every image is reshaped to a 1024x1 vector (see Figure 2.5). We call this vector $\mathbf{x}$. In order to form the covariance matrix many independent examples of $\mathbf{x}$ are required. The covariance matrix is calculated and the eigenvalues and eigenvectors of the matrix are determined.

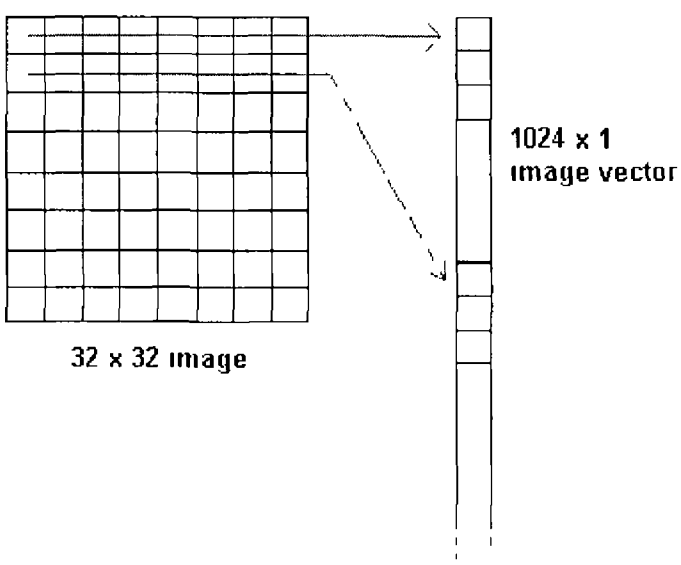


Figure 2.5 Reshaping a square to a vector

By selecting a few eigenvectors corresponding to the largest eigenvalues a lower dimensional feature space is formed The projection of an image vector into this subspace is a point (see Figure 2 6)

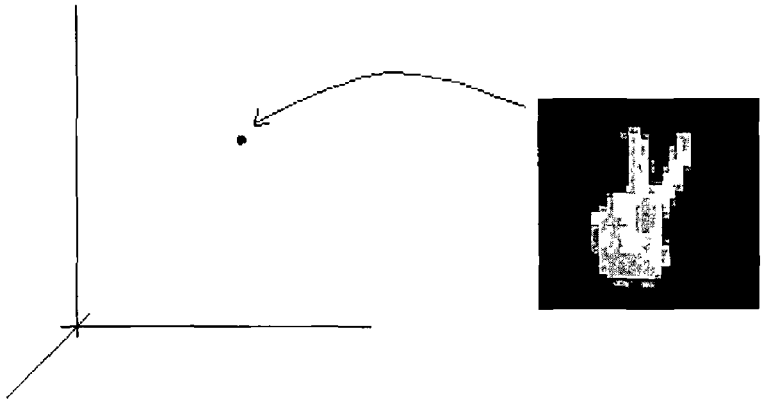


Figure 2 6 A three dimensional eigenspace and a projected image

By using PCA in a lower dimensional space and cluster analysis one can classify an unknown shape in a set of known clusters

The above mentioned techniques are mostly used for the purpose of static shape recognition But there are other methods like image *cross correlation* "Cross-correlation is a standard method of estimating the degree to which two series are correlated" [Bourke 1996] Consider two series $\mathbf{x}_t$ and $\mathbf{y}_t$ where $t=0\,1,2,\ ,N\,1$ The cross-correlation r at delay d is defined as

$$r_d = \frac{\sum_t (\mathbf{x}_t - \mu_x)(\mathbf{y}_{t-d} - \mu_y)}{\sqrt{\sum_t (\mathbf{x}_t - \mu_x)^2}\sqrt{\sum_t (\mathbf{y}_{t-d} - \mu_y)^2}} \qquad (2\,7)$$

where μ_x and μ_y are the means of the corresponding series If the value of r is calculated for all the delays $d=0,1,2,\ N\,1$ it results in a cross-correlation series of twice the length of the series $\mathbf{x}_t$ and $\mathbf{y}_t$ assuming they are the same length For example, for the two hand shapes shown in Figure 2 7 the graphs of energy level for the pixels of these shapes are shown in Figure 2 8 These pictures are in 32x32 resolution and 256 grey level Therefore, each picture has 1024 pixels The graph of the cross-correlation of these images is shown in Figure 2 9 It is clear that at a little before $d=0$ (zero delay) the two series have the highest cross correlation, and they have the highest similarity

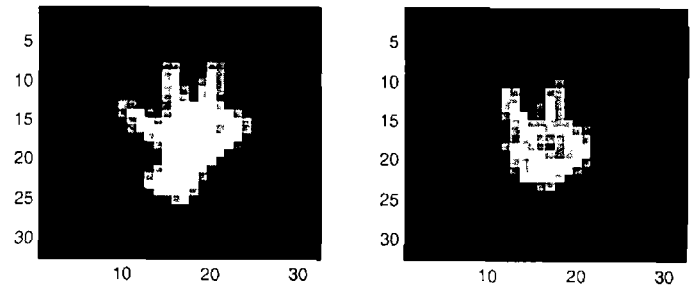


Figure 2 7 The two hand shapes in 32x32 resolution

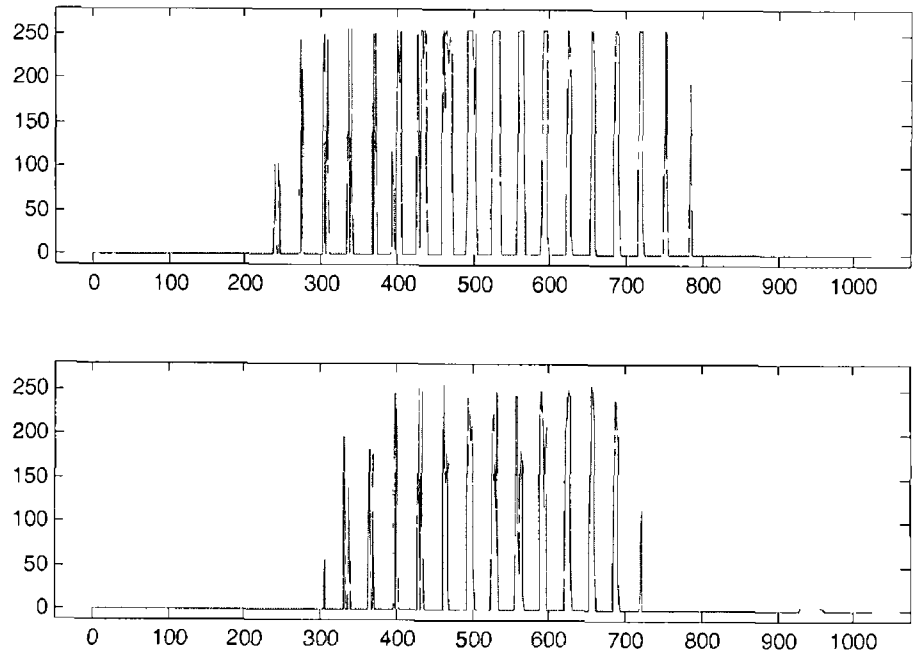


Figure 2 8 The energy level of the pixels of the hand shapes

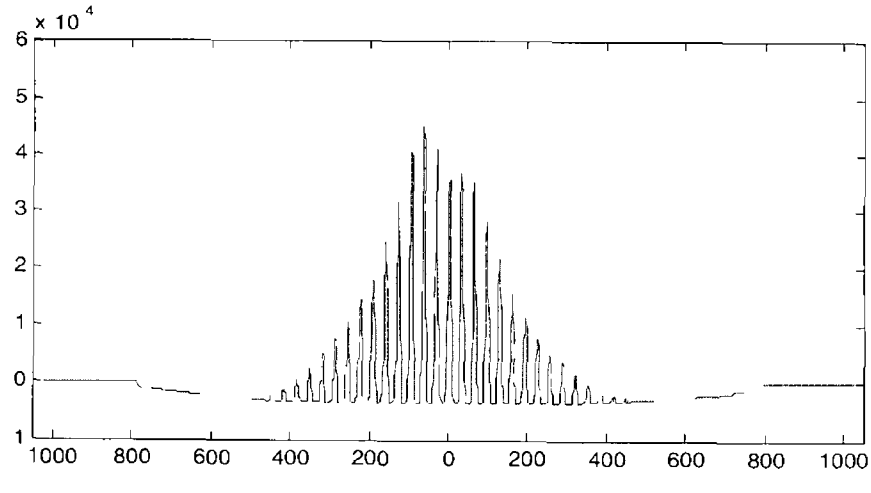


Figure 2 9 Cross correlation of the pixel values of the images

However, our experiments show that the cross-correlation technique is an extremely time consuming process, and we will not use it for pattern matching Many other recognition techniques have been addressed in the literature

An error-correcting encoding framework for solving multi-class pattern recognition problems has been addressed in [Erenshteyn 1999] Under this framework they have developed an algorithm for code generation The algorithm allows generating codes of different lengths

A hierarchical static shape recognition technique has been introduced in [Wu 2002] This hierarchical approach works by using the idea of "divide-and-conquer" They divide up the data set into groups of images, which are similar to each other This is done by deliberately blurring the images so that small differences between similar images will be eliminated Thus a group of original images may become reduced to just one image, which represents the entire group So the total size of the data set will be reduced For example, the images shown in Figure 2 10(a) are the sign 'a' in Irish Sign Language at full resolution, then scaled to 32x32 pixels and finally blurred The images of Figure 2 10(b) are the sign 'e' at full resolution, scaled to 32x32 and then blurred

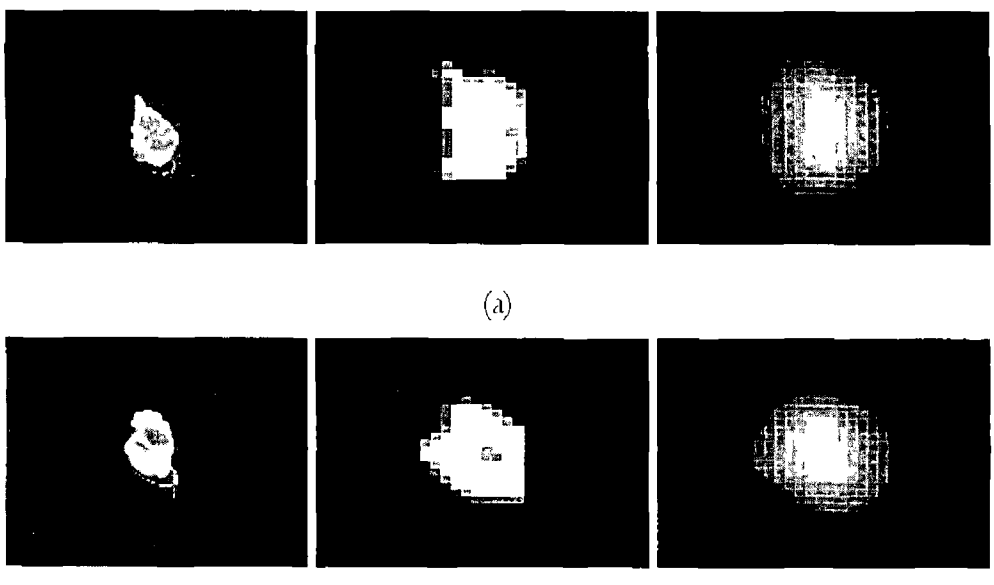


Figure 2 10 Blurring images to make the hierarchical database The images are blurred at different levels (from left to right) so that the final blurred images look similar

By blurring the images at a series of different levels and grouping the similar blurred shapes together a hierarchical database of shapes is made Then for an unknown hand shape it is blurred at the corresponding levels and classified The classification is done by recognising

that the blurred hand shape belong to which group of similar shapes at a level of blurring Therefore, instead of searching the whole database of hand shapes that contains thousands of images they search the sets of similar shapes and recognise the most similar set By reducing the level of blurring they step-by-step search the smaller sets of shapes to recognise the hand shape in the final stage with zero level of blurring Therefore, a hand shape can be extracted from within the hierarchy containing thousands of hand shapes in a short time

In [Eisenstein 2001] a clustering technique has been presented to detect hand signs They use K-means and adaptive clustering techniques to recognise static hand shapes using a smart glove with 22 sensors that pertain to the position of different joints that constitute a hand The results show that for 10 different hand shapes performed by 10 people between 55% to 83% accuracy (varies from person to person) is obtained with the K-means algorithm While the K-means algorithm shows sensitivity to the input data and the order it is presented to the algorithm the adaptive algorithm is less sensitive with 66% to 77% accuracy They have also tested the Nearest Neighbour technique and achieved 81% to 88% accuracy Then they conclude that Nearest Neighbour provides superior performance when it is provided with a large training set size

An interesting model called the Point Distribution Model (PDM) has been introduced in [Cootes 1992] for building shape models In this method a shape is represented by a set of labelled points (see Figure 2 11), in which variation in shape can be included in the model

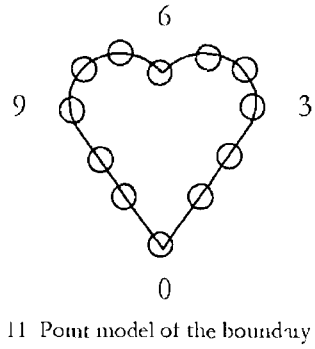


Figure 2 11 Point model of the boundary of a shape

However, for different samples of a shape the equivalent points must be aligned in the same way with respect to a set of axes The required alignment is achieved by scaling, rotating and translating the shapes so that they correspond as closely as possible This is done by minimising a weighted sum of squares of distances between the equivalent points on different shapes There are other methods for automatic landmark identification [Hill 2000]

Others have improved the PDM by making non-linear generalisation either by polynomial regression [Sozou 1994] or multi-layer perceptron [Sozou 1995]. In [Heap 1997] the principal limitation of PDM, non-specificity, has been considered, and a two-level hierarchy in shape space employed to improve efficiency.

Another hierarchical recognition architecture presented in [Heidemann 2000a] consists of an adaptive feature extraction based on Vector Quantization and local PCA. They first structure the input data by Vector Quantisation. Therefore, a set of reference vectors is extracted in this stage. Then, for each reference vector a locally valid Principal Component Analysis is performed. For classification of the features extracted by the local PCA an expert net of the local linear map (LLM) type is attached to each reference vector. A neural network is used in this algorithm to classify features.

Classification and recognition of articulated and occluded objects have also been addressed in the literature. A hierarchical 3D object representation model [Hauck 1997] has been introduced in which recognition is done by first estimating the 3D pose of the static component and afterwards determining the relative 3D pose of the remaining components recursively. This model can cope with the problem of self-occlusion.

A model-based statistical algorithm has been developed in [Ying 1999] to recognise occluded objects from noisy features. In this paper, two different statistical occlusion models, an independent prior model and a Markov Random Field prior model, are examined to measure their robustness in the presence of occlusion. The first model is based on the assumption that each feature in an object template can be occluded with a certain probability independent of whether any other features are occluded. The second model is based on the assumption that if one feature is occluded the likelihood that the other nearby features are occluded should increase which shows a spatial correlation to the process of occlusion. Their results show that the model with spatial correlation is more robust than the one without.

An interesting occlusion recognition algorithm based on a neural network model has been addressed in [Fukushima 2000]. The authors argue that in the presence of occluding object, recognition of partially occluded object is easier (see Figure 2.12). When the occluding object is not visible, distinguishing which features are relevant to the original pattern and which are newly generated by the occlusion is hard. These irrelevant features will hinder a visual recognition system from recognising the occluded pattern correctly. A hierarchical neural

network has been presented to block the signals for irrelevant disturbing features to reach higher stages of the recognition system

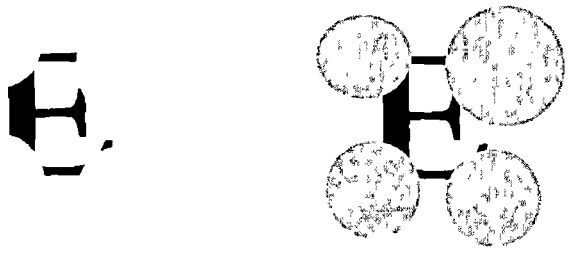


Figure 2 12 Patterns partially occluded by invisible and visible masking objects

A matching technique has been presented in [Edwards 1997] for interpretation of colour scenes containing occluded objects This algorithm uses an interactive, coarse-to-fine correlation-based method that uses hypothesised occlusion events to modify the scene-to-template similarity measure at run-time An algorithm has been introduced in [Heidemann 2000b] for supervised learning the segmentation of partially occluded objects The algorithm works based on the classification of object windows The object windows are smaller than the object size but large enough to evaluate the structural object features and colour In every input window, features are extracted by local PCA and subsequently classified by a neural network

An unsupervised clustering technique has been presented [Yanez-Suarez 1999] for identification of partially occluded objects The contour of each object is modelled with an approximating polygon whose edges are then projected into the Hough space (for more information about Hough Space please refer to [Leavers 1992]) A structurally adaptive neural network generates clusters of collinear and/or parallel edges, which are used for identifying the partially occluded objects within each polygonal approximation

Dealing with occlusion, also, has been addressed in stereo imaging [Intille 1994 Jojic 1999] In stereo imaging, except for thin objects with large matching disparity, all stereo scenes obey the *ordering constraint* if object *a* is to the left of object *b* in the left image then *a* will be to the left of *b* in the right image In [Intille 1994] they have developed a data structure called *disparity space image* (DSI) which is used in a stereo algorithm that finds matches and occlusions simultaneously By incorporating highly-reliable matches or *ground control points* (GCPs) the algorithm's occlusion-cost sensitivity and algorithmic complexity is reduced significantly

Many gesture recognition algorithms use Principal Component Analysis (PCA) for feature extraction and representation A standard PCA technique discussed in [Jolliffe 1986] was explained in detail earlier A comparison among PCA and *Linear Discriminant Analysis* (LDA) has been presented in [Martinez 2001] Despite some reservations about the superiority of algorithms based on LDA versus the algorithms based on PCA it has been shown that this is not always true Indeed, it has been demonstrated that PCA can outperform LDA when the training data set is small and, also, that PCA is less sensitive to different training data sets

Linear and *quadratic discriminant analysis* have been presented in [Friedman 1989] *Non linear Component Analysis* is a non-linear form of PCA [Scholkopf 1996, 1998] It generalises PCA to the case where the principal components in input space are not of interest, but rather in principal components of variables, or features, which are non-linearly related to the input variables

Another approach in *Non linear PCA* has been studied in [Moghaddam 1999] The Linear PCA manifold, Linear Independent Component Analysis (ICA) manifold and Non-linear Principal (NLPCA) manifolds have been compared in his paper They are employed to recognise a large set of individual faces It has been shown that the PCA and ICA result in better recognition rate than the NLPCA The general difficulty of computing non-linear manifolds and complexity of cost functions riddled with local minima can be the main reasons for the poor performance of NLPCA

A probabilistic matching technique has been addressed in [Moghaddam 1998, 1999] for direct visual matching of faces in a database They divide the variation of facial images into two classes *intra personal*, which are the variations in appearance of the same individual, due to different expressions or lighting and *extra personal* which are the variations in appearance due to a difference in identity Then a *probabilistic measure* to measure the similarities based on a Bayesian analysis of images differences in the mentioned classes has been introduced

Non-linear component analysis as a kernel eigenvalue problem [Scholkopf 1996] and statistical pattern analysis based on nonlinear Kernel PCA [Ruiz 2001] have been addressed as a new generation of PCA in pattern recognition A kernel version of Mahalanobis distance and a kernel version of minimum squared error (MSE) have also been introduced [Ruiz 2001]

PCA has been used to find adaptive bases for multiresolution [Brennan 2000] An input image is decomposed into components, which are uncorrelated Then a single layer network using *Generalised Hebbian Algorithm* (GHA) with a minor modification is used to implement the multiresolution PCA A method has been introduced [Lyons 1999] for automatic classification of facial images based on elastic Graph-Matching, a 2D Gabor wavelet representation, and LDA First, the images are transformed using a multiscale, multiresolution set of Gabor filters Then two type of grids, rectangular and fiducial, are registered with the face The amplitude of the Gabor transform coefficients are sampled on the grid and combined into a single vector called labelled graph vector or LG vector Then they use PCA to reduce the dimensionality of input space for the ensemble of LG vectors from a training set of images The LG-PCA vectors from the training set are then analysed using LDA in order to separate vectors into different clusters with individual facial attributes

A multi-view dynamic face model has been developed for the *shape and pose free* facial texture patterns [Li 2001] Using a kernel technique to perform LDA in high-dimensional feature space a Kernel Discriminant Analysis is developed to extract the significant non-linear features which maximise the between-class variance and minimise the within class variance They have implemented this technique in the problems of modelling faces across multi-views, extracting the non-linear discriminant features, and recognising moving faces in image sequences

Finally, a general review of statistical pattern recognition has been presented in [Jain 2000]

2 3 2 Dynamic Movement Tracking and Recognition

A dynamic gesture is a movement and change in the shape of hand appearing in a sequence of images In dynamic gesture recognition a sequence of images is analysed to be classified as one of the known classes of gestures In every image a hand is presented which is gradually moving and changing in the sequence

A prerequisite to recognition of hand and body movements is tracking In a movement where the hands are moving in order to do something or imply a meaning tracking the hands is crucial

A 3D model of hand has been presented in [Davis 1999] for tracking hand movements In this model, the hand is represented by five cylindrical models, which are fitted, to the third

phalangeal segments of the fingers (see Figure 2 13) Six 3D-motion parameters for each model are calculated that correspond to the movement of the fingertips in the image plane Then the 3D nature of the hand motion is presented by establishing the trajectories of the moving models

In order to track the articulated objects in motion the *EigenTracking* algorithm has been introduced in [Black 1996] Estimating the view and the transformation that takes this view into the image are the main problems in pattern matching in eigenspaces For a particular view of an object they define a *subspace constancy assumption* between the eigenspace and the image Then they do a non-linear optimisation in order to recover the view and transformation

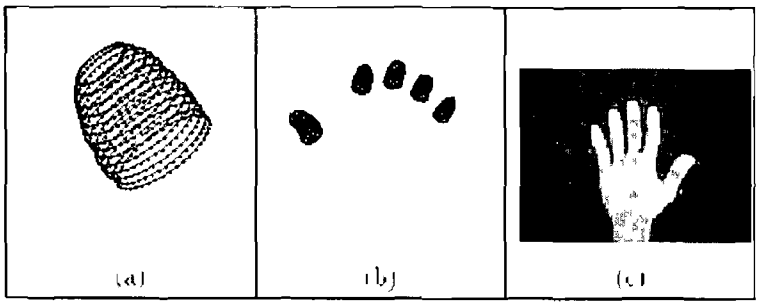


Figure 2 13 3 D cylindrical model

Furthermore, an *EigenPyramid* is defined for the problems with large transformation between model and image

A simple Kalman Filter-based method [Kohler 1997] has been used for remote controlling devices in a very natural environment without the need of markers attached to the user's body

Tracking a moving object by Kalman filtering has been widely addressed in the literature [Brown 1997 Chui 1999 Bowden 2000] An algorithm for tracking multiple objects in the presence of occluded motion has been addressed in [Dockstader 2000] They use dynamic frame differentiating to detect changes between the frames and motion Then by using a probabilistic mixing of coarse motion estimates, change detection information, and unobservable prediction the algorithm creates accurate trajectories of moving objects

In stereo imaging an algorithm has been introduced [Jojic 1999] for tracking articulated structures in dense disparity maps They introduce a Bayesian network where tracking is

addressed as an inference problem in the image formation model This model is a statistical model where occlusion has been taken into account

Another tracking algorithm is called Conditional Density Propagation (CONDENSATION) [Isard 1998a] CONDENSATION uses learned dynamical models, together with visual observation, to propagate a random set over time This random set represents the probability distribution of possible interpretations, which is used in *factored sampling* The results are robust but the process is time-consuming A smoothing filter has been developed [Isard 1998b] for CONDENSATION It is a statistical technique of conditioning the state distribution on both past and future measurements when tracking is complete However, it has been shown in [Sherrah 2000] that in the case of occlusion a Bayesian network has some advantages over CONDENSATION from the computationally expensive point of view

In [Gong 2002] an approach has been addressed to learning the semantics of scene context in order to interpret visual events without object segmentation and motion grouping They have used adaptive Gaussian Mixture models to separately model and recognise slow changes such as illumination cycles A Bayesian network has been presented to model the semantics of human body configuration Then the visual tracking problem has been addressed by reasoning about observations using a semantic-based inference model

Another model has been introduced in [McAllister 2002] for hand tracking in smart desks and driving In this technique, a circle is fitted to palm and a line to the forearm in order to model each hand (see Figure 2 14) Therefore, a top-view camera is needed to observe the hands from the correct angle Also, the palm shape should be so that the algorithm can fit a circle to that It has been shown that for some shapes of palm the model fails to fit a circle

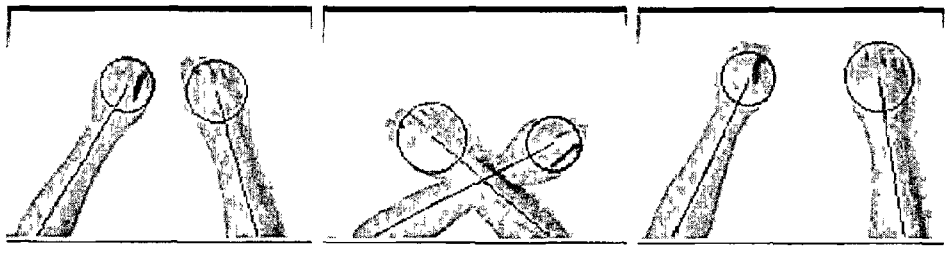


Figure 2 14 A circle and a line is fitted to the hands for tracking

A tracking algorithm has been also proposed [Zieren 2002] for tracking the hands of a user in a frontal camera view They use multiple cues, incorporate tracing and prediction algorithms,

and apply probabilistic inference to determine the trajectories of the hands even in the case of hand-face overlap, but not hand-hand overlap

Another approach to the tracking problem has been presented in [Stauffer 1999] Given that the tracking system is unaware of the identity of object the identity remains the same for the entire tracking sequence They use Linear Vector Quantisation (LVQ) to develop a code-book of representations on the entire set of representations acquired by the tracker Then they accumulate joint co-occurrence statistics over the code-book by treating the set of representations in each sequence as an equivalency multi-set Finally, they perform hierarchical classification using only the accumulated co-occurrence data

After tracking the hands in a sequence of images we should recognise the gestures Many different approaches to this problem have been addressed in the literature [Pavlovic 1997] Gesture recognition via pose classification [Ng 2000] employes a Radial Basis Function (RBF) Neural Network to recognise a static hand pose Then the combined outputs from a set of recurrent neural networks (RNN) and Hidden Markov Model (HMM) chains have been used to recognise gestures from the temporal sequence of pose classifier outputs

In a spatio-temporal approach to the hand gesture recognition [Lin 1998], an RBF network and Dynamic Time Warping method are used to design a space invariant and time interval invariant system, which can distinguish between various spatio-temporal data representing hand gestures

A method based on a *Hyper rectangular Composite Neural Network* (HRCNN) has been presented [Su 1998], in which the HRCNN is trained to generate templates for basic hand shapes Then by computing accumulative similarities a hand gesture is classified

A technique has been developed [Campbell 1995] for representation of human body movements based on space curves in subspaces of a *phase space* Using this representation, they develop a system for learning new body movements from ground truth data by searching for constraints which are in effect during the movement to be learned, and not in effect during other movements

Dynamic Time Warping and Hidden Markov Models (HMM) are two techniques, which are used in speech [Jelinek 1997] and hand gesture recognition [Schlenzig 1994, Starner 1995a Huang 2000 Bunke 2001]

In [Starner 1995a Starner 1995b] HMMs have been introduced for the recognition of American Sign Language (ASL) They use colour gloves in order to track the hand Then by using Hidden Markov Models they recognise hand gestures excluding finger spelling The same research group have tested their system [Starner 1996] using a wearable camera in a cap worn by the user and have shown that it results in a better recognition rate

Many other approaches have been made based on Hidden Markov Models A parametric HMM (PHMM) has been introduced in [Wilson 1999] for the recognition of gestures Their approach is to extend the standard HMM method of gesture recognition by including a global parametric variation in the output probabilities of the HMM Using a linear model of dependence they developed an expectation maximisation (EM) algorithm for training the parametric HMM A similar EM algorithm maximises the output likelihood of the PHMM for a given sequence during testing

In [Lee 1999], a threshold model based on HMM has been introduced in order to handle the non-gesture patterns in a hand motion This model calculates the threshold likelihood of an input pattern Then it approves or rejects the pattern as a gesture

A multi-Principal-Distribution-Model (PDM) has been presented in [Huang 2000] that uses a PDM model to track the hand shape The training hand shapes are divided into a number of similar groups, with each group trained for an individual PDM shape model Then the HMMs are employed to determine model transition among these PDM shape models

An HMM-based method has been introduced in [Nam 1996] for recognising the space-time hand movement patterns In this method, an HMM models the spatial variance as well as time variance in the hand movements Then an HMM-based segmentation method has been introduced to deal with continuous connected hand gestures Since the dimensional complexity is high in a 3D space, a plane fitting method is used to reduce the dimensionality into 2D The 2D data are encoded as the input to the HMMs

Brand et al [Brand 1997] have used a special structure of HMM called Coupled HMM for modelling and recognising two interactive temporal sequences like the two hands With a vocabulary of 3 T'ai Chi gestures performed by one person 94 2% recognition rate has been observed using a test set that includes one third of the examples of the training set

Comparisons of some HMM-based approaches to gesture recognition have been presented in [Morguet 1999]

Many other approaches have been made in the literature for gesture recognition A state-based technique for the summarisation and recognition of gestures has been presented in [Bobick 1995] In this method, a gesture is defined to be a sequence of states in a measurement or configuration space These states, for a given gesture, are used to capture both the repeatability and variability evidences in a training set of example trajectories They have developed techniques for computing a prototype trajectory of a group of trajectories for defining configuration states, and recognising gestures from an unsegmented, continuous stream of sensor data

A neural network model called modified CombNET-II has been used [Tamar 1999] to do temporal analysis and to be used in the large set of human movements recognition systems They present a feature extraction method based on morphological Principal Component Analysis that completely describes a hand gesture in a 22-dimensional time varying vector Then by using a combination of the network and the feature extraction method they have developed a complete Japanese Kana hand alphabet recognition system

A multi-stage hand and face segmentation consists of colour segmentation, temporal segmentation, and video object plane generation [Habili 2001] In colour segmentation the skin colour is modelled as a normal distribution for classifying the pixels of an image to be skin or non-skin In temporal segmentation they localise the moving object in the sequence of images Then the results are analysed to yield a change detection mask

A fast gesture recognition algorithm detects the number of fingers in an image [MacLean 2001] to recognise a gesture for teleconferencing applications This algorithm works with a stereo active visual system Another stereo imaging system for recognition of gestures in real time is based on 3D prediction of the hand pose [Ishibuchi 1993] In [McKenna 1998] gestures are modelled probabilistically as sequence of events Then the events are matched to the visual input using probabilistic models estimated from the motion feature trajectories

A comparison between the trajectory-based and history-based representation for recognition of gestures has been presented in [Morrison 2003] They compare these representations using Hidden Markov Models, moment features, and normalised template matching Relative advantages and disadvantages of each method have been also presented

Graph matching-based pattern recognition algorithms have been reviewed in [Bunke 2000] Graph matching and graph theory [Diestel 1997 Karpinski 1998] are the basis of current research in computer vision and artificial intelligence In [Shamaie 2001] a graph matching algorithm has been introduced to find the best match among a group of gestures

Finally, a general review of recognising hand gestures, body motion, and face detection has been presented in [Pentland 2000]

As there has not been much research on the particular problem of recognition of bimanual movements including occlusion we consider this problem as the main problem of this thesis Since the bimanual movements form a large set of movements we investigate this problem, its different aspects and introduce new models, techniques and algorithms We try to propose general solutions with the least restrictions We will compare some of the techniques during our investigation and find the best way through them

Summary and Conclusion

In this chapter we presented the main problem of this thesis A literature review of the problems, methods, models, and algorithms, which are directly or indirectly related to our recognition problem, was presented A wide spectrum of algorithms and methods from static shape recognition to dynamic gesture recognition were reviewed briefly We surveyed the advantages and disadvantages of some of the very well-known techniques in temporal and spatial pattern recognition

Details of some of the models and algorithms like Hidden Markov Models, Bayesian Networks, Kalman Filtering, Vector Quantization and Graph-Matching will be given in separate chapters in future

Chapter 3

HAND SHAPE RECOGNITION

Recognition of shapes without a particular analytical model is more difficult than the recognition of regular geometrical shapes. Geometrical shapes such as circles and ellipses obey well-known analytical descriptions. These descriptions are usually used in pattern recognition in order to recognise the geometrical shapes appearing in an image. However, known analytical and mathematical models cannot describe natural shapes such as the hand shapes. The problem becomes more difficult when we meet non-rigid objects. There are many statistical pattern recognition techniques for the recognition of non-rigid objects, which we reviewed in the last chapter.

In this chapter first we take a look at the basics of image formation and acquisition. Then the recognition of hand shapes is addressed using some statistical pattern recognition methods. A new gesture recognition algorithm with a real-time application for communicating with computers with no physical contact is presented at the end of the chapter.

3.1 Image Formation and Acquisition

General criteria for characterising an image acquisition system are illumination, focusing, sensing, and digitisation.

3.1.1 Illumination

An object should be properly lighted so as to make it visible to a visual system like the human eye. In machine vision, the first step is to light the object to make it detectable by the sensor. In an image-capturing system a two-dimensional projection of a three-dimensional object is acquired. A 2D image has two important attributes *contrast* and *resolution*. These attributes are used as a base for an action or for a decision and must be measurable.

Contrast is the range of differences between the light and dark parts of an image [Zuech 2000]. This value is measured between the principal object and the background.

Resolution is a distance measurement associated with the smallest detectable object [Zuech 2000] Based on the area that a vision system is working on, the required resolution is obtained For example, for locating an object within an area of 1x1 inch, the system resolution must be less than 1 inch (see Figure 3 1)

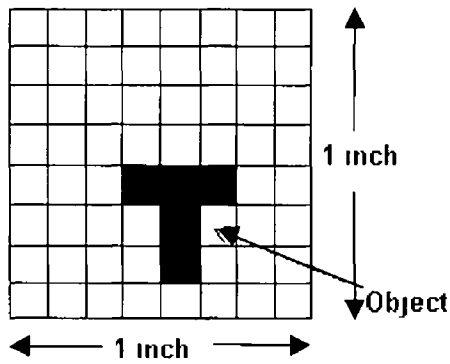


Figure 3 1 Locating an object in the area of 1x1 inch

Lighting is a dedicated source of illumination that is needed to get rid of other illumination sources which usually are the main source of environment light and the reflections from the objects in the environment These sources of light can result in a complex pattern of light and affect the recognition of the principal object

The main objectives of lighting are optimising the contrast (grey scale differences), normalising variances due to ambient conditions and simplifying image processing [Zuech 2000]

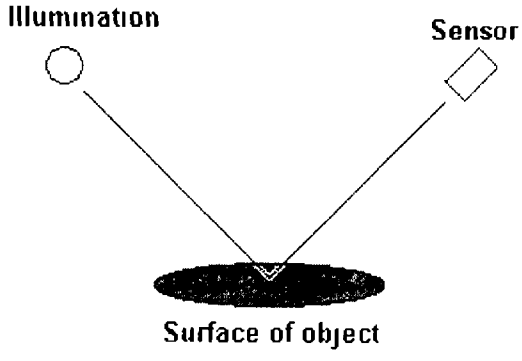


Figure 3 2 Lighting of an object

3 1 2 Focusing and Image Formation

In an imaging system the points located in the object plane are projected into the image plane as image points where these points are to be sensed by a sensor Like a human eye a sensing device like camera has a lens that translates the object points to image points (see Figure 3 3)

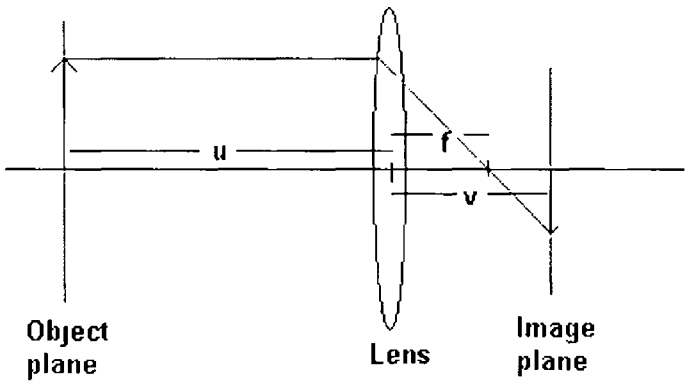


Figure 3 3 Image formation using a thin lens

In this figure

f focal length of the lens

u distance between the object planes and the lens

v distance between the image plane and the lens

where

$$\frac{1}{u}+\frac{1}{v}=\frac{1}{f}$$

Focusing means changing the distance of the lens and the object plane, u, in order to equate the v and the physical distance between the lens and the sensor's surface The image is sharp in this case Otherwise the image is blurred because the image is formed either in front of or behind the sensor's surface

3 1 3 Sensor

Modern camera sensors are made from *Charge Coupled Devices* (CCD), which use light sensitive materials to convert light photons to electrical charge In a matrix array, thousands of light

sensitive diodes are positioned accurately and shift registers transfer the charge from each pixel to form a video signal (see Figure 3 4)

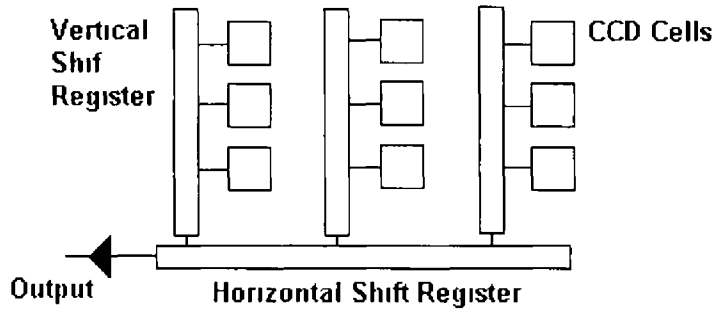


Figure 3 4 A CCD sensor system

3 1 4 Sampling

Temporal sampling is a function of the integration time period of the sensor In modern cameras, this integration period can be varied, and since it effectively determines the total number of photons per cell it can be used as an electronic shutter

3 1 5 Digitisation

In order to generate data suitable for computer analysis the video signal of the camera must first be sampled It normally needs a fast analogue-to-digital converter Selecting an appropriate sampling rate is related to the resolution The video signals are sampled both spatially and in amplitude Since most imaging devices generate a video signal in which the pixels are extracted sequentially, by sampling the signal at equally-spaced, discrete moments in time, spatial sampling can be achieved At these discrete moments the amplitude of the signal is measured

3 2 Hand Image

To capture an image of a hand, we used a CCD camera model JAI CV-M40 placed at about 2 5 meters from the subject's hand The data sheet of this camera is presented in Appendix A This monochrome camera is able to capture up to 233 partial frames per second We set the camera to work in 120 fps mode Since a hand gesture normally is performed in a fraction

of a second, a usual serial webcam working in 5 fps cannot provide enough information for real-time tracking and occlusion detection Also, since the training of the algorithms of gesture recognition require large amount of information a fast camera can provide this information in real-time However, given that the camera is working fast we will show that the processing times required by the algorithms do not allow the system to work faster than a regular 30 frames per second camera

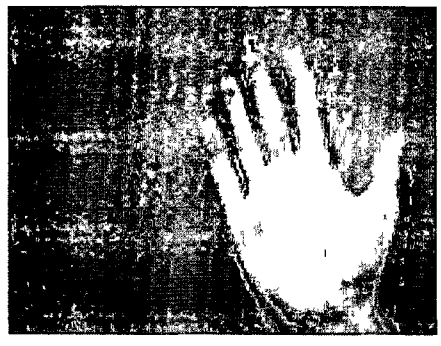


Figure 3 5 A hand image

An image of a hand (see Figure 3 5) is processed in order to extract the hand from the background (see Figure 3 6) using the grey-level detection A *blob analysis* algorithm called *Grassfire* [Pitas 1993] is used to search an image and find the connected regions with the same grey values as the hands The algorithm scans an image from left to right, top to bottom to find the pixels of connected regions with values belonging to the range of the hand's grey scale For the first pixel found in that range it searches around the pixel to find other pixels Therefore, by finding all the pixels belonging to a connected region the hand is extracted

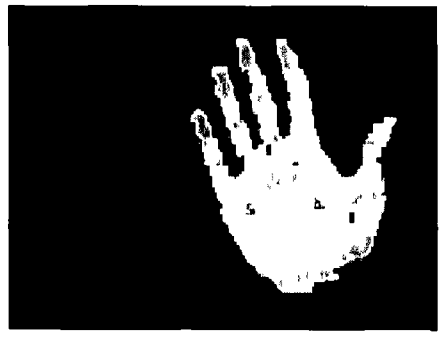


Figure 3 6 Hand extraction by pixel grey level detection

In this thesis we are, specifically, going to deal with rotation and changes in the hand shapes as well as the movement of the hand Other research projects have already dealt with the change in the angle and position of arm and hand in the room frame [Starner 1995a] But

they have totally ignored changes in the fingers and shape of hand, which is used for hand gestures, e g finger spelling in sign languages

In order to analyse the extracted shape we have to change the format of the image to a standard constant format A 32x32 image format was selected and all the hand shapes are mapped onto this format A scaling algorithm is used for mapping process as explained in the next section

3 2 1 Scaling

In this method first a rectangle around the hand is constructed (see Figure 3 7) Then the content of the rectangle, the hand, is mapped onto a big blank square in which the centre of the rectangle is positioned at the centre of the square (see Figure 3 8)

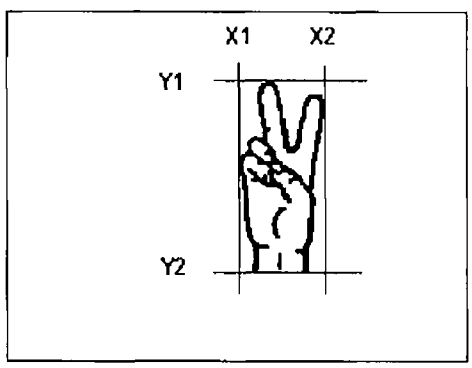


Figure 3 7 A rectangle around the hand is considered in which the x and y parameters are the coordinates on the horizontal and vertical axes of image plane representing the leftmost, rightmost top and bottom of the hand's blob in the image

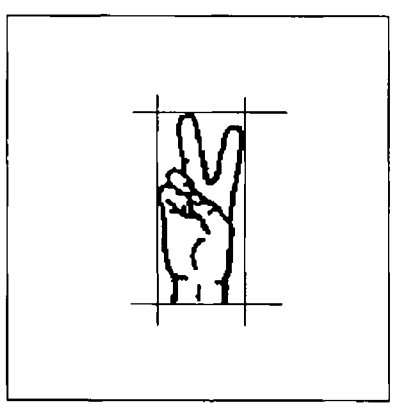


Figure 3 8 Mapping an extracted hand onto a blank square so that the centre of the rectangle is positioned on the centre of the blank square

The size of this square, which is a power of 2, is chosen so that the biggest hand shape (the hand shape that all the fingers are open showing the number five) is smaller than that. This size is a parameter that determines the scale factor of final shape in a 32×32 image. Then the big square is divided to 32×32 squares. Finally each small square is mapped to a point in a 32×32 frame (see Figure 3 9).

Calculating the mean value of all the pixels in a small square does this mapping process. Mathematically, first the X1, X2, Y1, and Y2 are extracted (see Figure 3 7). Then the size of big square is selected to be,

$$S = \alpha \mathrm{T}(\max(X2 - X1, Y2 - Y1)) \qquad (3\ 1)$$

where operator T can be defined as

$\mathrm{T}(a)$ the first power of 2 which is greater than or equal to a, and $\alpha \in N$ is the scaling factor

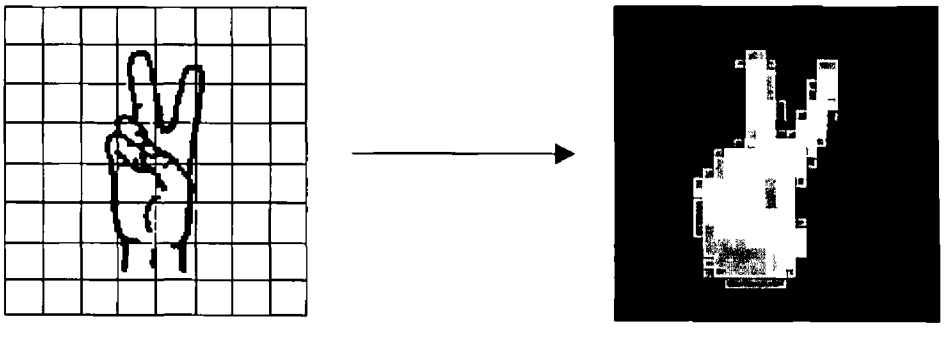


Figure 3 9 Scaling an image from a high resolution to a 32×32 pixel image

However, I must be constant for all the shapes. An estimation of the biggest possible hand shape can determine I. The big square is divided by 32 so that every small square has a size of s by s, where

$$s = \frac{S}{32}$$

For every small square in the area of $(X, X + s - 1)$ and $(\Psi, \Psi + s - 1)$ the mean value is calculated as,

$$\mu = \frac{1}{s^2} \sum_{\psi=\Psi}^{\Psi+s-1} \sum_{\chi=X}^{X+s-1} e_{\chi\,\psi} \qquad (3\,2)$$

where e is the energy level or the pixel value at the position χ and ψ

Finally, one can normalise the pixel values to be between zero and one For example, if the imaging system capture pixels within the range of 0 and 255 grey-level, all the values can be divided by 255 Although, there are other scaling methods, this simple fast method suffices our needs

3 2 2 Static Hand Shape Recognition

Now, every image has 32x32 (=1024) pixels that constitute a 1024-dimensional feature space To reduce the dimensionality of the feature space we use Principal Component Analysis (PCA) explained in the previous chapter By using PCA a new feature space called an *eigenspace* is formed where the projection of an image to this space is a point Due to the rotation of hand in the images and the presence of noise different examples of a shape form a cluster of points in the eigenspace (see Figure 3 10) For a number of different hand shapes all the examples of all the shapes can be employed to make a common eigenspace The projection of examples into this common subspace forms the clusters, each of which associated with a hand shape (see Figure 3 11)

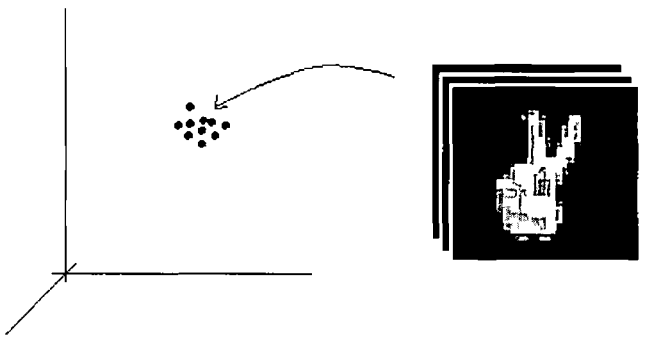


Figure 3 10 Cluster of points in a 3 dimensional eigenspace for a hand shape

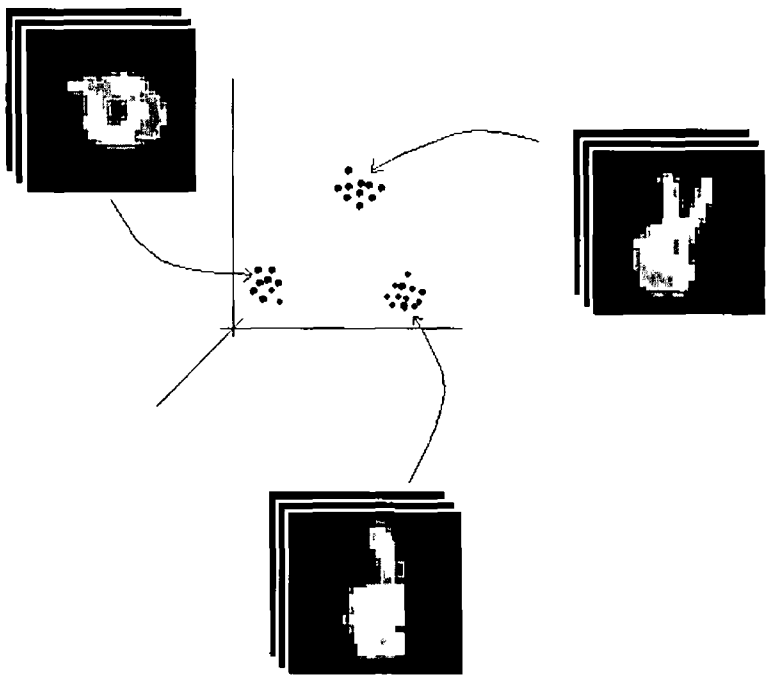


Figure 3 11 Clusters representing different hand shapes in the eigenspace

For a given hand shape, one can project it into this common space by

$$\mathbf{y} = \mathbf{A}^{\mathrm{T}}\mathbf{x}$$

where **A** is the matrix of selected eigenvectors and **x** is the image vector of the shape This gives a point in the subspace By using cluster analysis and nearest neighbour methods, the nearest cluster to the given hand shape can be identified (see Figure 3 12)

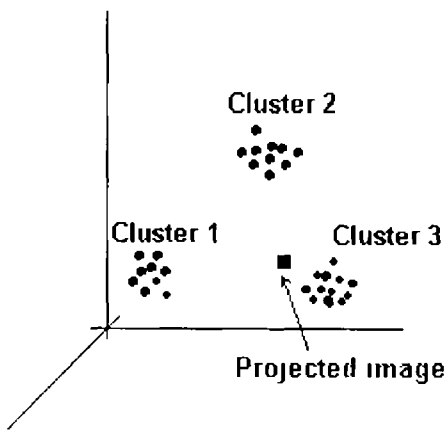


Figure 3 12 The projected image is identified as the third known shape by finding the nearest cluster

We developed an algorithm based on this method of static hand shape recognition By this algorithm we simulate all the functions of a mouse pointer in an MS Windows environment This enables us to use hand gestures to do regular mouse tasks like, clicking, moving, cutting, pasting, resizing, etc

3 3 A Human Computer Natural Interface

In order to interact with a computer without any physical contact we have developed a human computer natural interface This interface is based on an algorithm that controls the mouse cursor, and controls the computer just by hand gestures

A mouse system has a moving pointer on screen and two buttons The left button is normally used for selecting and opening icons and the right button has different functions defined in every program For controlling the mouse pointer a camera captures the hand gestures of a user By moving the hand in every 4 directions on a plane the mouse pointer moves And by changing the shape of hand the two buttons of the mouse are simulated For this purpose three hand shapes were defined as in Figure 3 13

For each shape we captured 350 to 450 examples to form the training set, and by using the algorithm described previously three clusters in the eigenspace are formed In the training set some rotations, for every shape are included The three-dimensional eigenspace and the clusters are shown in Figure 3 14

A move *B* left click *C* right click

Figure 3 13 Three hand shapes used in the mouse controlling system

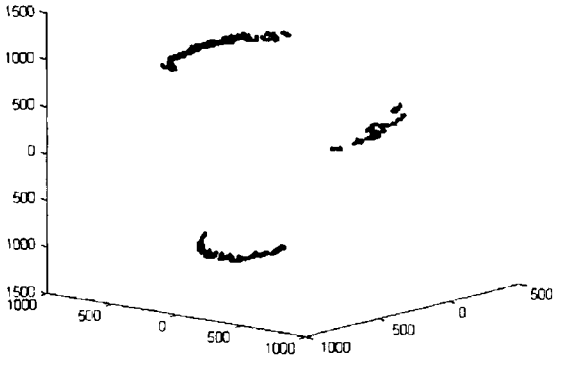


Figure 3 14 The clusters of the three hand shapes in the eigenspace

In the recognition phase by projecting an image into this eigenspace the hand shape is identified A state-based model is constructed for the mouse functions This model contains four states and connecting edges (see Figure 3 15)

The camera captures the hand shape continuously Different cases may happen in every state,

State 1 (default) In this state the hand is in shape *A* By moving the hand the mouse pointer moves over the screen While we are in State 1 three cases may happen

Case 1 (default) If the hand shape is *A* we remain in State 1, and the system just responds to the hand movements

Case 2 If the hand shape becomes *B* it jumps to State 2 through edge E_{12} This edge is defined as pressing the left button on mouse

Case 3 If the hand shape becomes *C* it jumps to State 3 through edge E_{13} This edge is defined as pressing the right button on mouse

State 2 In this state hand is in shape *B* Again three cases may happen

Case 1 If the hand shape becomes *A* it jumps to State 1 through edge E_{21} This edge is defined as releasing the left button

Case 2 (default) If the hand shape is *B* we remain in state 2, and the system just responds to the movements

Case 3 If the shape becomes *C* it jumps to state 4 through edge E_{24} This edge is defined as pressing the right button on mouse while holding the left button This is a proper choice Because during changing the hand from

shape *A* to shape *C* it may pass from near the shape *B* Therefore, the system first goes to state 2, and then moves to state 3

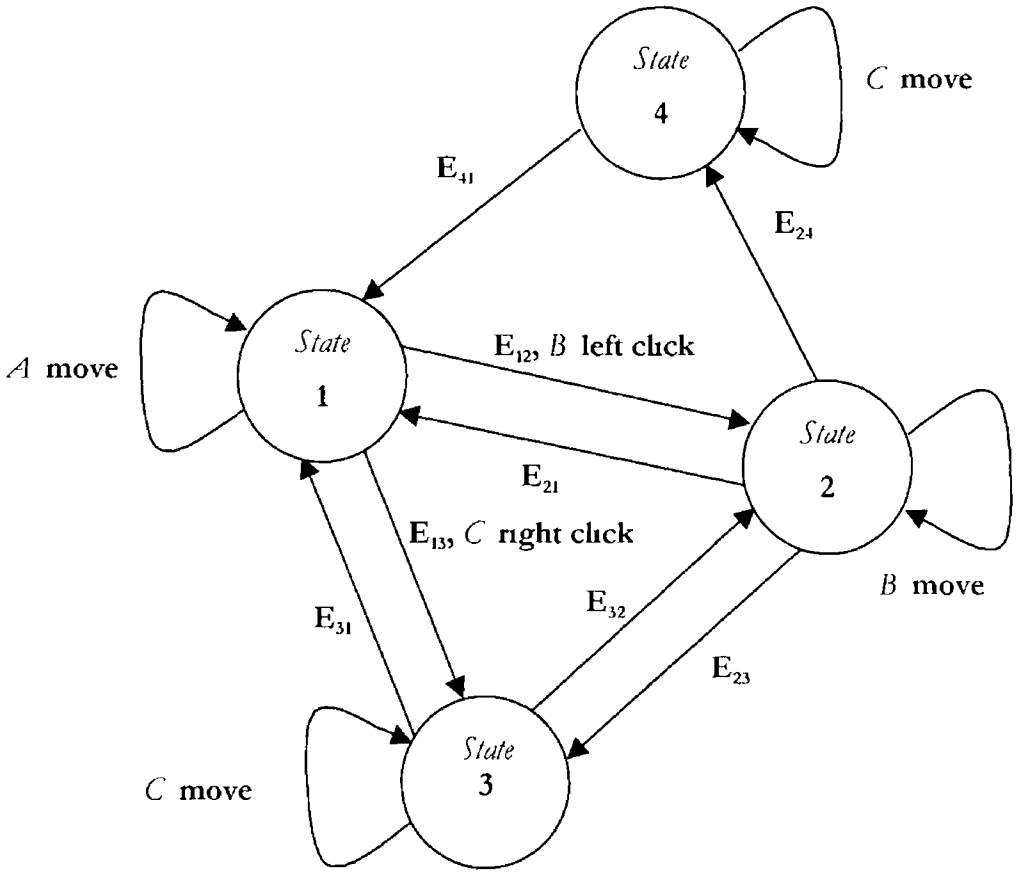


Figure 3 15 A four state model for mouse functions

State 3 In this state hand is in shape *C*

Case 1 If the shape becomes *A* it jumps to state 1 through edge E_{31} This edge is defined as releasing the right button

Case 2 If the shape becomes *B* it jumps to state 2 through edge E_{32} This edge is defined as releasing the right button and pressing the left button

Case 3 (default) If the hand shape is *C* we remain in state 3, and the system just responds to the hand movements

State 4 In this state hand is in shape *C*

Case 1 If the shape becomes *A* or *B* it jumps to state 1 through edge E_{41} This edge is defined as releasing the right button

Case 2 (default) If the shape is *C* we remain in state 4, and the system just responds to the hand movements

In order to keep the movement of mouse pointer smooth, the movement of the hand is processed in parallel with the shape recognition Choosing a point on the hand rectangle, which is invariant to a change in the hand shape, is important Since we have used the point and thumb fingers for the different shapes, the corner of the rectangle opposite to the head of the point finger is used The position of this point determines the position of the mouse pointer on screen Therefore, by moving the hand on a plane the mouse pointer moves But changing the shape of hand for pressing the buttons will not affect the position of the mouse pointer (see Figure 3 16)

We implemented the above algorithm as a real-time interface The algorithm responded in a high level of performance Using this application the user is able to interact with the computer with the hand gestures The interface is general so that the hand gestures are used to communicate with every program on the computer It translates the hand gestures to the functions defined in every application for the mouse A screen shot of the program is presented in Figure 3 17

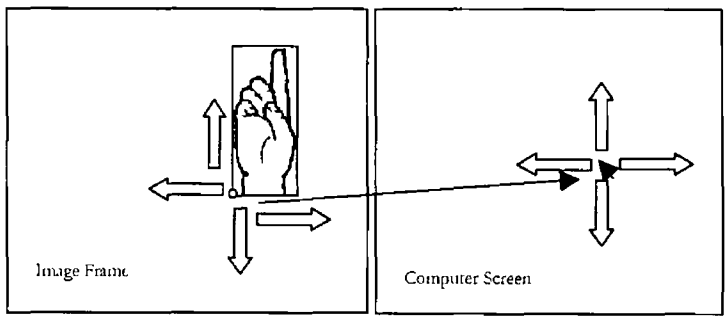


Figure 3 16 Moving the hand moves the mouse pointer on screen

Summary and Conclusion

In this chapter the basics of image formation and acquisition systems were presented We explained the illumination, the visual sensors and the way that images are formed on the sensors We also described the statistical pattern recognition techniques for hand shape extraction and scaling Then dimensionality reduction was presented in which the technique of Principal Component Analysis is used By using the methods of clustering and nearest neighbour we classified the hand shapes and recognised a static hand shape

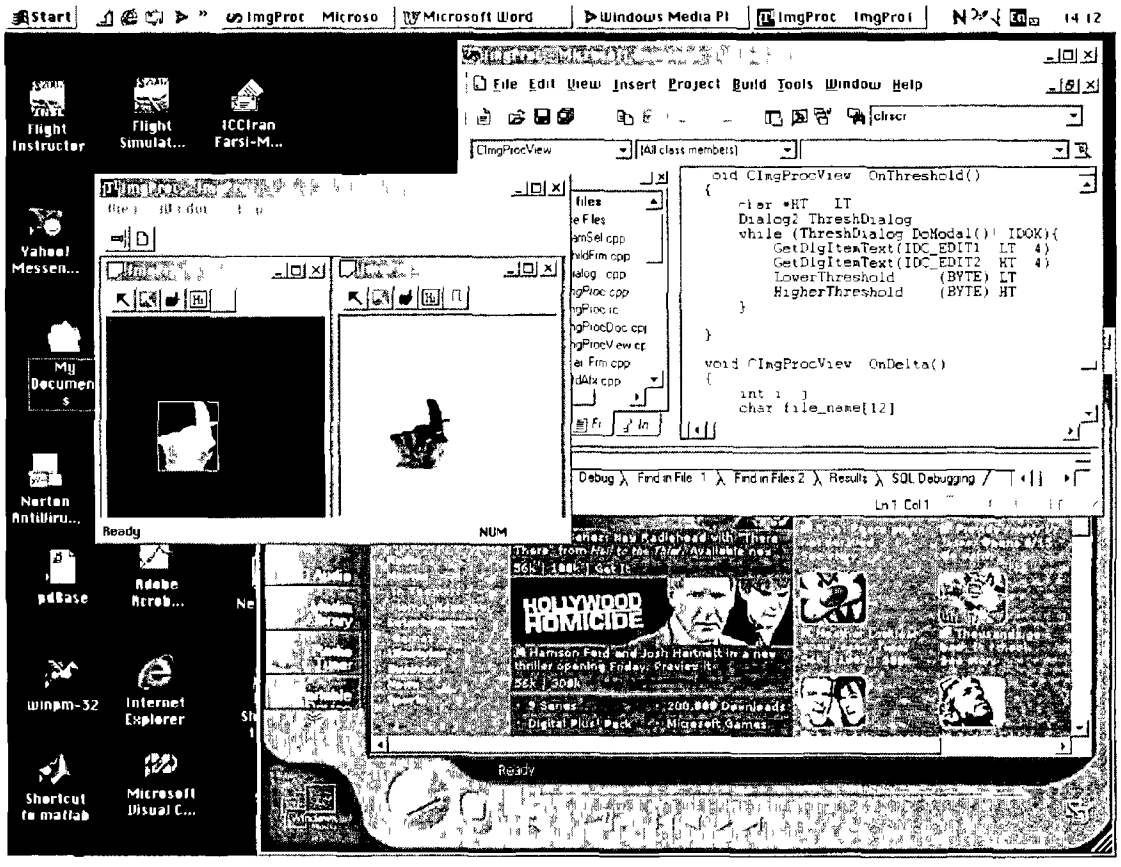


Figure 3.17 A screen shot of the gesture recognition application for mouse control

An algorithm was introduced in order to make a human computer natural interface. In this interface the functions of a mouse in a Graphical User Interface is simulated by the hand gestures. Using the statistical pattern recognition techniques we recognise the hand shapes and detect the changes. A state machine was introduced in which the hand gestures are mapped onto the mouse functions. We developed the interface as a real-time application and the ability of the interface to communicate with every application on the computer was demonstrated. As a future work we will do some usability testing by different users.

Chapter 4

DYNAMIC HAND GESTURES

A dynamic gesture is a movement of the hand in order to show something or imply a meaning. A few gestures from British Sign Language are shown in Figure 4.1. These gestures, representing numbers, involve hand movements.

Using machine vision techniques a dynamic gesture is represented by a sequence of images. In this chapter we discuss the recognition of a dynamic gesture appearing in a sequence of images taken by a CCD camera.

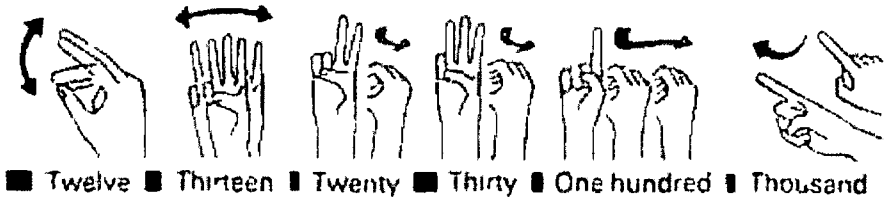
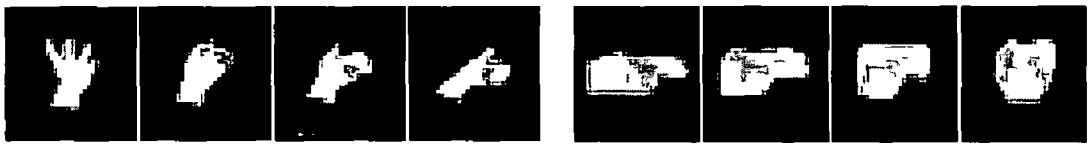


Figure 4.1 The hand movements to show the numbers in British Sign Language (source: A-Z Deafblindness website at www.deafblind.com/signs.html)

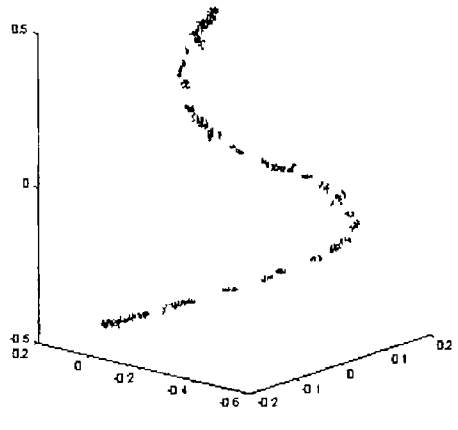
A dynamic gesture can be of many types. It can be a movement or rotation of the hand or a change in the hand shape (see Figure 4.2 (a)). In all cases the projection of the image sequence in the feature space is a trajectory (see Figure 4.2 (b)). As in Chapters 2 and 3, the feature spaces are the eigenspaces formed by the eigenvectors of the covariance matrix of the images using Principal Component Analysis. A trajectory in the eigenspace is a set of points each of which is the projection of an image (from the image sequence containing the hand gesture) into the eigenspace.

In the next section we study the trajectories of gestures in the feature spaces. An unsupervised clustering technique is presented for clustering the trajectories of the gestures. We use this technique in the following sections to make *HyperClasses* of the gestures each of which includes a set of multidimensional gaussian distributions. A new spatio-temporal

gesture matching algorithm is introduced that uses the HyperClasses to calculate the probability that a given unknown gesture matches one of the gestures in the vocabulary This algorithm is based on a Graph-Matching technique We present experimental results at the end of the chapter and will show that the presented algorithm works very well in recognising the dynamic hand gestures



(a)



(b)

Figure 4 2 (a) Different hand gestures appearing in the sequences of images (b) the trajectory of a gesture in the feature space

4 1 In Feature Space

In order to reduce the dimensionality of data and reduce noise we use the trajectories of gestures in the eigenspaces For this purpose, first an eigenspace[1] for a gesture must be established Performing a gesture several times and capturing (at least 1024) images provides the required data for constructing the (1024×1024) covariance matrix and calculating the eigenvalues Another advantage of capturing many samples is the involvement of variations By performing a gesture several times variations in the hand shapes and movements are recorded in the sequences This helps us to have a better view of the gesture and its variations

[1] As we will see in Section 4 4 based on an experiment we construct an eigenspace with 7 dimensions

By constructing the eigenspace and projecting the entire image sequences of the gesture into it, the manifold[2] of the gesture is formed (see Figure 4 3) The manifold has many points, each of which is related to an image Projection of many examples of a gesture in the feature space makes a large number of points Working with a large number of points is usually time consuming Therefore, we need a technique to reduce the number of points in a manifold

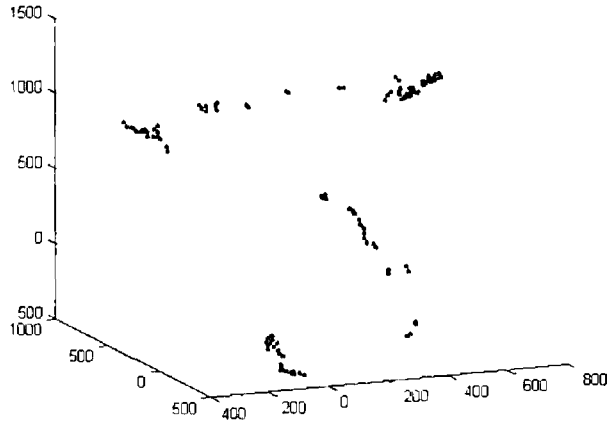


Figure 4 3 The manifold of a gesture in the eigenspace Two samples of a gesture projected in the eigenspace show variations in the trajectories

We can approximate a manifold by a small number of points each of which represents a group of the original points around it Having a manifold with a small number of points needs less memory space and makes the processing faster, while the main attributes of the original manifold are preserved

Since we do not have any prior information about grouping the points in a manifold we use an unsupervised clustering technique called Vector Quantisation for clustering the data points

4 2 Vector Quantisation

Vector Quantization (VQ) is a lossy data compression method It approximates the data by a method similar to rounding-off or the nearest integer A one-dimensional VQ is shown in Figure 4 4

We use the word trajectory for the trajectory of a gesture in the feature space and the word manifold for the set of trajectories of all the examples of a gesture in the feature space

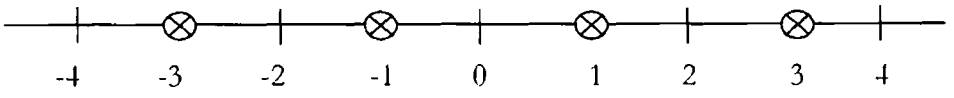


Figure 4 4 A one dimensional representation of Vector Quantization

In this figure, every number less than –2 is approximated by –3 Every number between –2 and 0 and between 0 and 2 is approximated by –1 and 1 respectively Finally, all the numbers greater than 2 are approximated by 3

An example of a two-dimensional VQ is shown in Figure 4 5

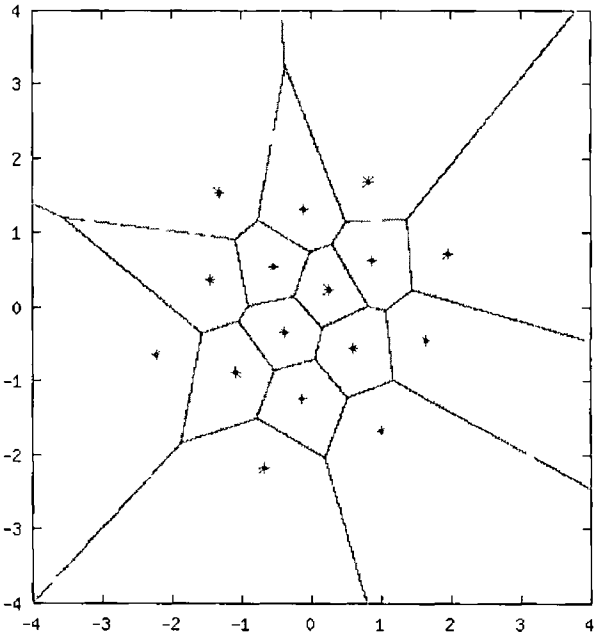


Figure 4 5 A two dimensional Vector Quantization

In a two-dimensional VQ every point with two coordinates falling in a particular region is approximated by a star associated with that region In Figure 4 5 there are 16 regions and 16 stars each of which can be uniquely represented by 4-bits Therefore, a large number of points in this 2-dimensional space can be approximated by 16 points These points (stars in the figure) are called codevectors, and the regions defined are called encoding regions The set of all codevectors is called a codebook

The Vector Quantization problem can be stated as

"Given a set of data and the number of required codevectors, find a codebook and a partition, which result in the smallest average distortion"

First we will give a definition of distortion In lossy data compression, it suffices for the decompressed data to have a reasonably close approximation of the original data A distortion measure is a mathematical entity, which specifies how close the approximation is For an original data x and its approximation x', $d(x,x')$ denotes the amount of distortion between x and x' where $d(x,x')\geq 0$

A squared-error one-dimensional distortion measure is defined as

$$d(x,x')=(x-x')^2 \tag{4 1}$$

In a training set-based Vector Quantization a training set of M vectors is given by $T=\{x_1,x_2,\quad ,x_M\}$ This training set is assumed to be sufficiently large so that all the statistical properties of the source of data are captured

We assume that the source vectors are k-dimensional

$$\mathbf{x}_m=\begin{bmatrix} x_{m1} \\ x_{m2} \\ \\ x_{mk} \end{bmatrix},\ m=1,2,\quad ,M \tag{4 2}$$

Let N be the number of codevectors and $C=\{c_1,c_2,\quad ,c_N\}$ represents the codebook Each codevector is k-dimensional

$$\mathbf{c}_n=\begin{bmatrix} c_{n1} \\ c_{n2} \\ \\ c_{nk} \end{bmatrix},\ n=1,2,\quad ,N \tag{4 3}$$

Let s_n be the encoding region associated with the codevector $\mathbf{c}_n$ and $P=\{s_1,s_2,\quad ,s_N\}$ denotes the partition of the space If the source vector $\mathbf{x}_m$ is in the encoding region s_n, then its approximation (denoted by $\mathbf{Q}(\mathbf{x}_m)$) is

$$\mathbf{Q}(\mathbf{x}_m) = \mathbf{c}_n \qquad if \ \mathbf{x}_m \in s_n \tag{4 4}$$

The average distortion is given by

$$D_{avg} = \frac{1}{M\,k}\sum_{m=1}^{M} \left\|\mathbf{x}_m - \mathbf{Q}(\mathbf{x}_m)\right\|^2 \tag{4 5}$$

where $\left\|\mathbf{x}_m - \mathbf{Q}(\mathbf{x}_m)\right\|$ is the Euclidean distance between $\mathbf{x}_m$ and $\mathbf{Q}(\mathbf{x}_m)$ defined as

$$\|\mathbf{d}\| = \sqrt{d_1^2 + d_2^2 + \quad + d_k^2} \quad \text{where} \quad \mathbf{d} = [d_1 \quad d_2 \qquad d_k]^T \tag{4 6}$$

Now, we find C and P such that D_{avg} is minimised The following two criteria must be satisfied if the C and P are a solution

1 Nearest Neighbour condition

$$s_n = \{\mathbf{x} \ \ \|\mathbf{x} - \mathbf{c}_n\|^2 \le \|\mathbf{x} - \mathbf{c}_{n'}\|^2 \ , \ n' = 1,2, \quad , N\} \tag{4 7}$$

2 Centroid condition

$$\mathbf{c}_n = \frac{\sum_{\mathbf{x}_m \in s_n} \mathbf{x}_m}{\sum_{\mathbf{x}_m \in s_n} 1} \ , \quad n = 1,2, \quad , N \tag{4 8}$$

In the first condition, s_n must contain all vectors that are closer to $\mathbf{c}_n$ than any other codevector The second condition says that the codevector $\mathbf{c}_n$ should be average of all the vectors in the encoding region s_n

The Linde, Buzo, Gray (LBG) Vector Quantization algorithm [Linde 1980] is an iterative algorithm, which solves the above two optimality criteria The algorithm needs an initial codebook $C^{(0)}$ which is obtained by a splitting method In this method first an initial codevector is found as the average of the entire data in the training set Then it is split into two The algorithm continues with these two codevectors as the initial codebook The final two codevectors are split into four and the process is repeated until the required number of codevectors is obtained A mathematical description of the algorithms is presented in Appendix B

By using Vector Quantisation we cluster the manifolds of the gestures in the feature space (eigenspace) Each cluster is represented by a codevector The set of codevectors of a gesture in the feature space represents the spatio-temporal variations of the hand gestures Then the manifolds are approximated by the extracted sets of codevectors These codevectors are represented as small solid circles in Figure 4 6

By using these codevectors we introduce a new gaussian Graph-Matching algorithm for the recognition of dynamic hand gestures

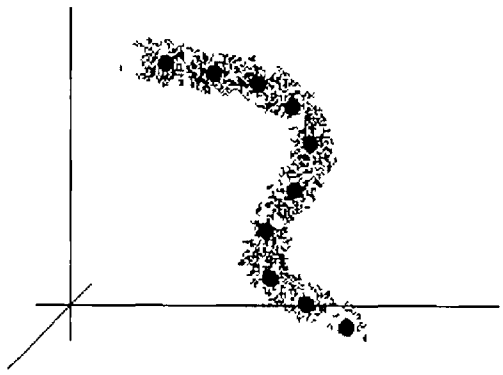


Figure 4 6 A manifold (the cloud) of a gesture in the eigenspace and the extracted codevectors (small solid circles)

4 3 A Spatio-temporal Pattern Matching Algorithm for Recognition of Dynamic Hand Gestures

In gesture recognition the problem is to find a gesture in a database that best matches a given gesture Different techniques such as neural networks [Su 1998 Lin 1998], position-based recognition [Ng 2000], and well-known Hidden Markov Models [Starner 1995a, Starner 1995b Lee 1999 Nam 1996 Wilson 1999] have been proposed in the literature for modelling and recognising hand gestures In [Bunke 2000], however, it has been stated that in computer vision and statistical pattern recognition there are a number of applications of *graph matching* that deserve attention

We introduce an algorithm based on graph matching for the recognition of dynamic gestures In this algorithm we construct the multidimensional gaussian distributions representing the variations of the gestures Then a given gesture is matched with the graphs representing these distributions in the eigenspaces In Chapter 6, we will compare the proposed algorithm with the widely used Hidden Markov Models in recognising a database of gestures

4 3 1 Constructing the Feature Space and the HyperClasses

For each gesture an eigenspace is formed using all the examples of the gesture We establish as many eigenspaces as individual gestures By projecting the examples of each gesture into its eigenspace a manifold is formed We call this the *main manifold*

Three main manifolds in the associated eigenspaces are shown in Figure 4 7 Each manifold represents the spatial and temporal variations of a gesture We divide each manifold into a set of classes with multidimensional gaussian distributions We call the whole set of classes of a main manifold a *HyperClass*

Therefore, every gesture is represented by a temporal sequence of spatially distributed classes An illustration of a HyperClass with 2-dimensional gaussian distributions is shown in Figure 4 8

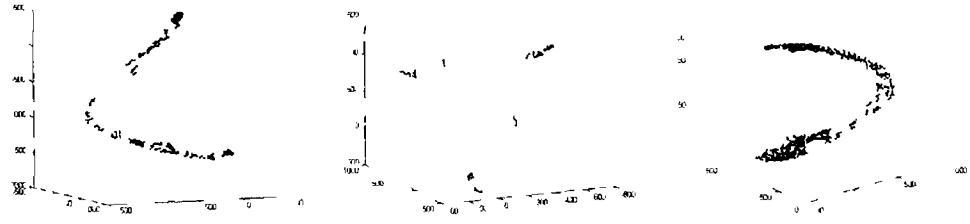


Figure 4 7 The main manifolds of three gestures in their three dimensional eigenspaces

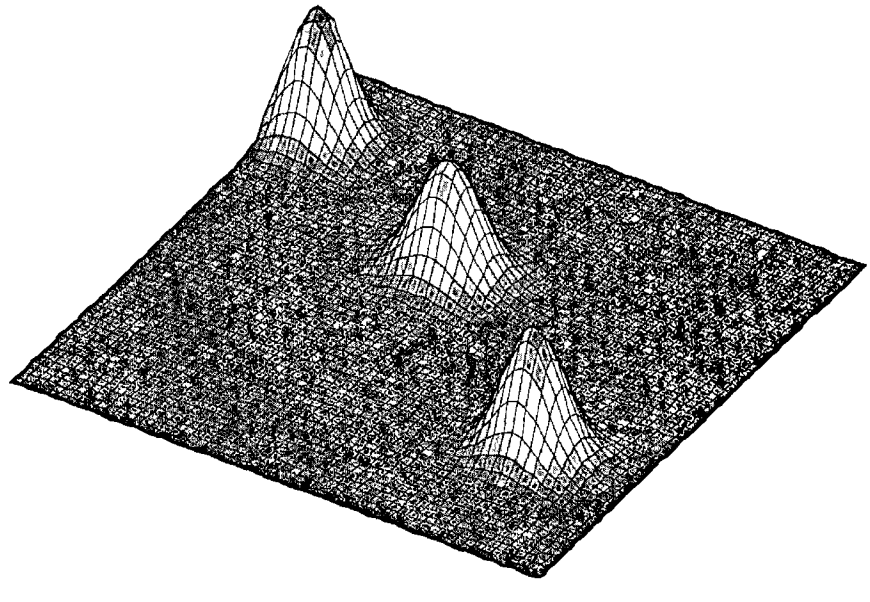


Figure 4 8 Every manifold in the eigenspace is represented by a sequence of classes each of which is described by a multi dimensional gaussian distribution

Each of the classes in a HyperClass represents a group of points in the eigenspace along the trajectory of the main manifold of the gesture To create these classes we must cluster each manifold in order to classify the data points belonging to each distribution We cluster the main manifolds into an equal number of clusters in all the eigenspaces

Since we have no prior information about the clustering of points on each manifold we should use an unsupervised clustering technique Therefore, the Vector Quantisation algorithm generates the clusters The clusters are approximated by the multi-dimensional

gaussian distributions A HyperClass must be trained with data points of the main manifold Each gaussian distribution in a HyperClass is fitted to the set of data points extracted in each cluster The HyperClass of each eigenspace is treated as a graph Individual distributions are the vertices of this graph (see Figure 4 9) We call them the main graph of each eigenspace

In order to recognise an input gesture it is projected into all the eigenspaces A new trajectory is formed in each eigenspace These trajectories should be approximated by graphs

Due to the fact that the Vector Quantization algorithm is a time consuming process we cannot use it in the recognition phase

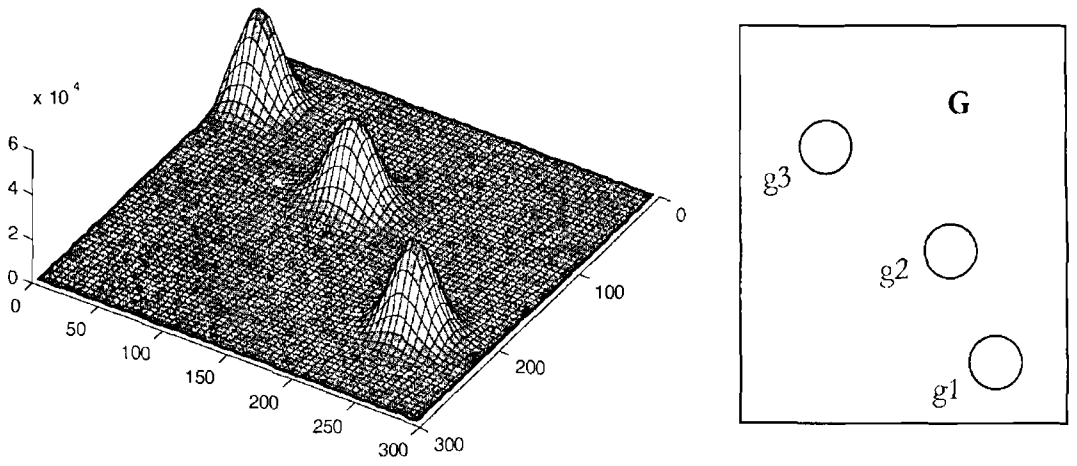


Figure 4 9 The graph of a HyperClass

The trajectory of the input gesture is divided into an equal number of clusters just based on the number of data points in the trajectory For every cluster the centre of gravity is calculated These central points are treated as the vertices of the new graphs (see Figure 4 10) Thus in each eigenspace two graphs are present The main graph and the graph of the input gesture We should find a match between the graphs in every eigenspace The best match represents the recognised gesture

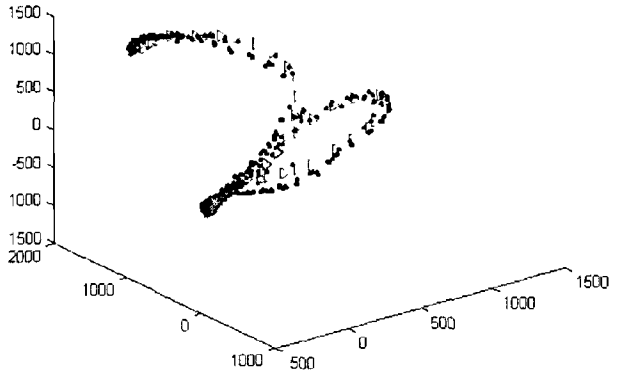


Figure 4 10 The trajectory of an input gesture (dots) is divided into the clusters The centre of gravity of the clusters form the graph of the input gesture (triangles)

4 3 2 Bipartite Graph Matching

"The bipartite graph matching problem is to find a set of pairwise disjoint edges of a bipartite graph based on a special characteristic of the edges" [Shamaie 2001] Due to NP-Completeness of the problem, finding the optimal solution requires exponential time However, a suboptimal or approximative solution can be satisfactory in some cases with polynomial processing time [Bunke 2000] The algorithm we introduce is based on the labelled edges of a complete bipartite graph

4 3 3 The Graphs

Let the main graph G_i of the i^{th} eigenspace be the set of vertices V_i and edges E_i,

$$G_i = (V_i, E_i) \tag{4 9}$$

In these graphs no edge connects any pair of nodes (vertices) Thus, $E_i \equiv \varnothing$ The graph of the input gesture in each eigenspace is also represented by a graph G'_i, $G'_i = (V'_i, E'_i)$ in which $E'_i \equiv \varnothing$

4 3 4 Matching the Graphs

For the two graphs of G_t and G_t in a subspace (eigenspace) we find a subgraph H_t. First, by using the graphs' vertices a complete bipartite graph is constructed (see Figure 4 11)

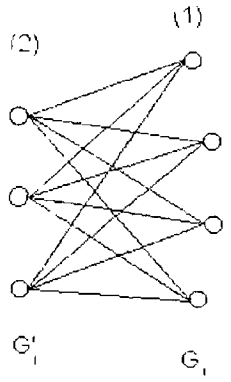


Figure 4 11 A complete bipartite graph

The edges of this graph are weighted by the probability of each vertex of the graph G_t in each class in the HyperClass,

$$e_{s_{t,k}s_{t,j}} = P(\mathbf{g}'_{t,k} \mid C_{t,j}), \quad \mathbf{g}'_{t,k} \in G'_t \tag{4 10}$$

where t is the eigenspace index, j stands for the j^{th} class and k stands for the k^{th} vertex in the graph G_t. The probability of a vertex $\mathbf{g}$ in a class of data C_t is given by the Mahalanobis distance and the gaussian probability density function,

$$P(\mathbf{g} \mid C_t) = \prod_{j=1}^{m} \frac{1}{\sigma_{t,j}\sqrt{2\pi}} e^{-\left(\frac{(g_j - \mu_{t,j})^2}{2\sigma_{t,j}^2}\right)} \tag{4 11}$$

where m is the number of dimensions of the eigenspace, $\sigma_{i,j}$ stands for the standard deviation of the distribution on the j^{th} principal axis of the i^{th} class, $\mu_{i,j}$ is the mean of the distribution on the j^{th} principal axis of the i^{th} class and g_j stands for the component of point g projected on the j^{th} principal axis of the distribution. The parameters of the probability density functions of each distribution are extracted by using Principal Component Analysis.

In the second set of vertices of the bipartite graph, the set belonging to the graph of the input gesture, for every vertex we find the incident edge with highest probability and eliminate the other edges. At the end of this stage the vertices of the first set, the set belonging to the main graph of the eigenspace, with no incident edge are removed (see Figure 4.12).

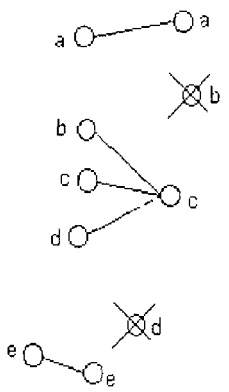


Figure 4.12. The edges with highest probabilities are found. The vertices with no incident edge are removed.

We get two sets of nodes (vertices) V_t and Γ_t where $\Gamma_t \subseteq V_t$ and $\Omega_t \subseteq G_t$. Γ_t is a subset of V_t because the number of vertices of the first set, G_t at the end of this stage is smaller than or equal to the order of G_t.[3] As can be seen in Figure 4.12, a vertex can be adjacent with many vertices in graph G'_t.

In the second stage in the first set of vertices, for every vertex we find the incident edge with highest probability (the probabilities that the vertices are labelled with in the first stage). At

[3] The order of a graph G is defined as the number of vertices in the graph $v = |G|$

the end of this stage the vertices of G'_t with no incident edge are removed Therefore, a set of nodes Γ'_t which is a subset of second graph ($\Gamma'_t \subseteq V'_t$) and $\Omega'_t \subseteq G'_t$ is obtained The remainder is a bipartite graph, which is a subgraph of the bipartite graph at the beginning The orders of Ω_t and Ω'_t are equal at the end

The algorithm reduces the number of nodes (vertices) while it is finding the match The decision is made based on the order of the remained bipartite graph The graph of the eigenspace with highest order is chosen as the best match In other word, we recognise the input gesture as the gesture in the vocabulary with the best matched graph (the bipartite graph with highest order)

The algorithm is summarised as follow

4 3 5 The Complete Algorithm of Gaussian Graph-Matching

In the following algorithm we refer to v_t and e_t as a vertex and edge of a graph respectively

1 Training Phase

a Several examples of a gesture are captured

b Step **a** is repeated for all gestures (N gestures) in the vocabulary

c By using PCA a subspace is made for each gesture and each gesture is projected into its own eigenspace to form the main manifold,

$$z_t = A'_t a_t \qquad t = 1,2, \quad ,N$$

where

A'_t is the orthogonal matrix whose k^{th} column is the k^{th} eigenvector of covariance matrix of a sequence of images represented as a_t, and z_t is the projection of image sequence a_t into its subspace

d The main manifold of each eigenspace is clustered by the Vector Quantization algorithm, the gaussian distributions are trained and the main graph is extracted

$$G_t = (V_t, E_t) \qquad t = 1,2, \quad ,N$$

where

$V_t = \{v_{t1}, v_{t2}, \quad , v_{tm}\}$

v_{tk} = the k th codevector

2 Recognition Phase

a A gesture b is captured

b b is projected into all the subspaces

$$t_i = A_i' b \qquad i = 1,2, \quad ,N$$

where

t_i is the projection of the image sequence b in the i^{th} subspace

c The graphs of gesture b are constructed in every subspace,

$$G_i' = (V_i', E_i') \qquad i = 1,2, \quad ,N$$

d The complete bipartite graph is formed with the two sets of vertices

$$K_i = (V_i'', E_i'') \qquad i = 1,2, \quad ,N$$

where

$V_i'' = V_i \cup V_i'$

and every edge in E_i'' labled with the probabilities

e The subgraph of the complete bipartite graph is found,

e 1 Starting from a set of vertices V_i', the edge with the largest label incident with a vertex of the set V_i' is found

e 2 The edge with the largest label is kept and the other incident edges for each vertex are removed

e 3 Step **e 2** is repeated for all the vertices in the set V_i'

e 4 The vertices of the set V_i with no incident edge are removed and the subset Γ_i ($\Gamma_i \subseteq V_i$) is obtained

e 5 Steps **e 1** to **e 4** are repeated with the set of vertices V_i of the main manifold, and the subset Γ_i' ($\Gamma_i' \subseteq V_i'$) is obtained The results are the matched bipartite subgraph H_i, for each subspace

f The matched subgraph H_i with the largest number of vertices, between the matched subgraphs of all the subspaces, represents the most similar gesture in the training set to the given gesture

The following likelihood is defined,

$$L_i = \frac{n_i(m_i + 1)}{2Y} \qquad (4.12)$$

where Y is the number of vertices of every main graph (usually the main graphs have the same number of vertices,) n_i denotes the number of vertices of the bipartite subgraph after the matching process, and m_i is the mean of the probabilities of the connected vertices in the final subgraph. This likelihood gives us a better measurement to find the best match because not only the number of matched vertices but also the average probability of the vertices is involved. The largest likelihood can be selected as the best match.

4.4 Experimental Results

Since only a few eigenvectors and principal components carry actual data and the rest are noisy we should choose the non-noisy principal components corresponding to the largest eigenvalues of the covariance matrix.

In order to choose a reasonable number of eigenvectors to form the eigenspaces we did an experiment with 10 gestures and a simplified version of the Graph-Matching algorithm. In this version of the algorithm the distributions were replaced by the centre of gravity of each cluster of points and the probabilities by the Euclidean distances on the edges of the bipartite graph. A few frames of the 10 gestures are shown in Figure 4.13. Given that the examples of the gestures had very little variations (spatially and temporally) the recognition rate of the simplified algorithm is plotted in Figure 4.14. This figure shows the recognition rate of the simplified algorithm versus the number of principal components. In other words, different numbers of dimensions in the eigenspaces results in different recognition rates. Initially as the number of principal components increases better recognition rates are observed. The best recognition rate was obtained with 7 principal components. By increasing the number of principal components further the recognition rate falls. This shows that from the seventh principal component onward the noisy principal components are involved that reduce the recognition rate of the algorithm. Therefore, we use 7-dimensional eigenspaces in the rest of our experiments and throughout the whole thesis. In [Sharifi 2003], however, it has been shown that the number of noisy eigenvalues and eigenvectors of a covariance matrix is independent of the amount of noise in the data. Therefore, by increasing or decreasing noise in the data the number of non-noisy principal components is almost constant.

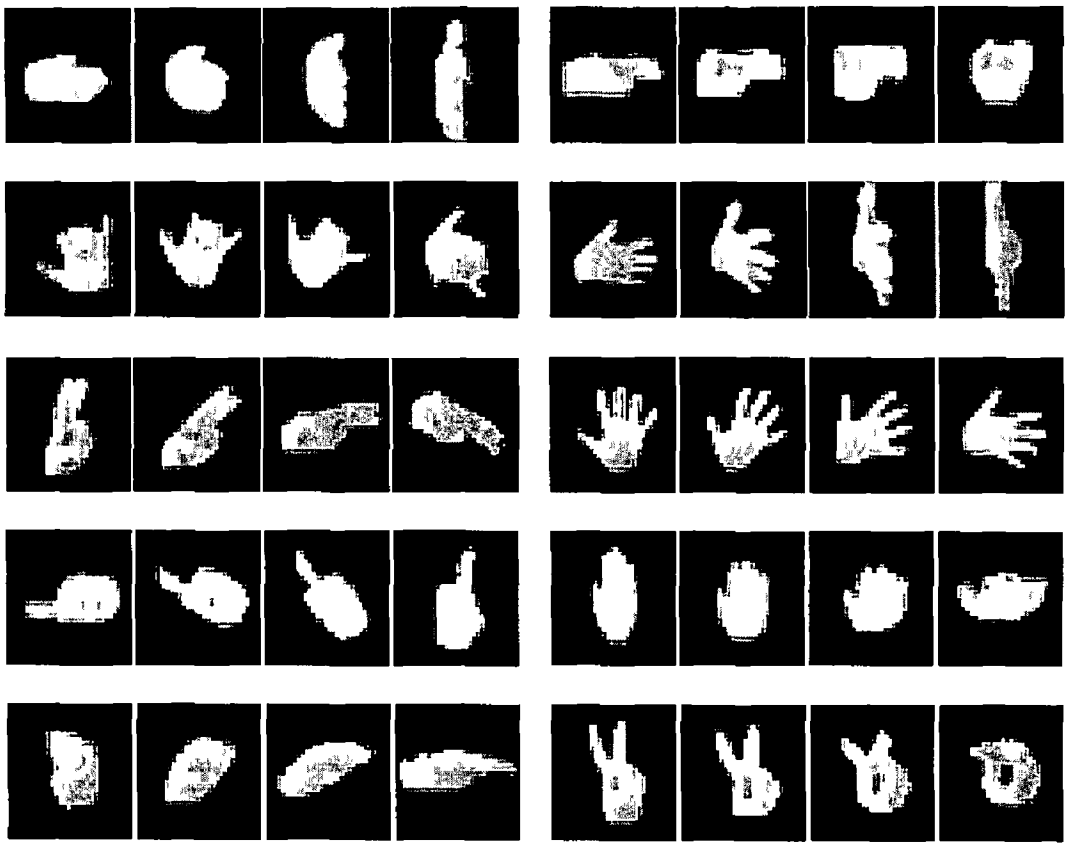


Figure 4 13 A few frames of the 10 gestures used in the first experiment

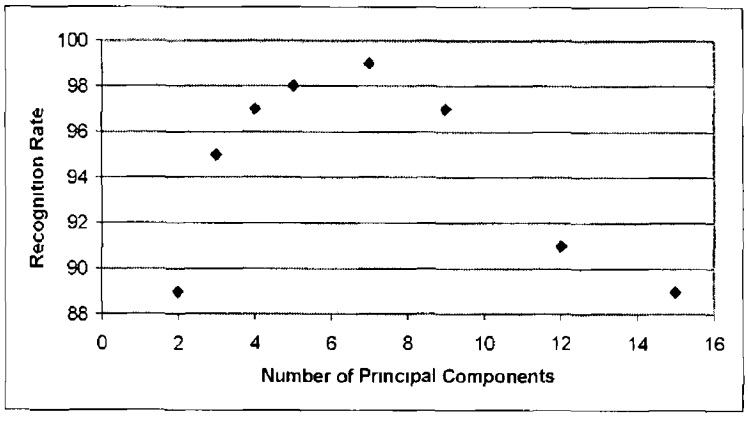


Figure 4 14 The recognition rate versus the number of principal components

In the following, we did some experiments with a vocabulary of 100 gestures with the original algorithm, the algorithm with gaussian distributions and the trained HyperClasses Here we present a couple of experiments to demonstrate the way that Graph-Matching algorithm works in real problems

Figure 4 15 shows a two dimensional representation of a 7-dimensional HyperClass of one of the experiments For a given gesture which is the same gesture as the gestures used to make this eigenspace and train this model the results of graph matching is as following,

Order of matched graph 22

which is a bipartite graph with 11 nodes on each partition

The likelihood of Equation 4 12 0 6874

A part of the trajectory of the given gesture in this eigenspace is shown in Figure 4 16, (the trajectory has been moved a bit up on the probability axis not to let it be hidden under the meshes) Obviously, the trajectory matches the gaussian distributions along its path

Another gesture is projected into this eigenspace and its trajectory is shown in Figure 4 17

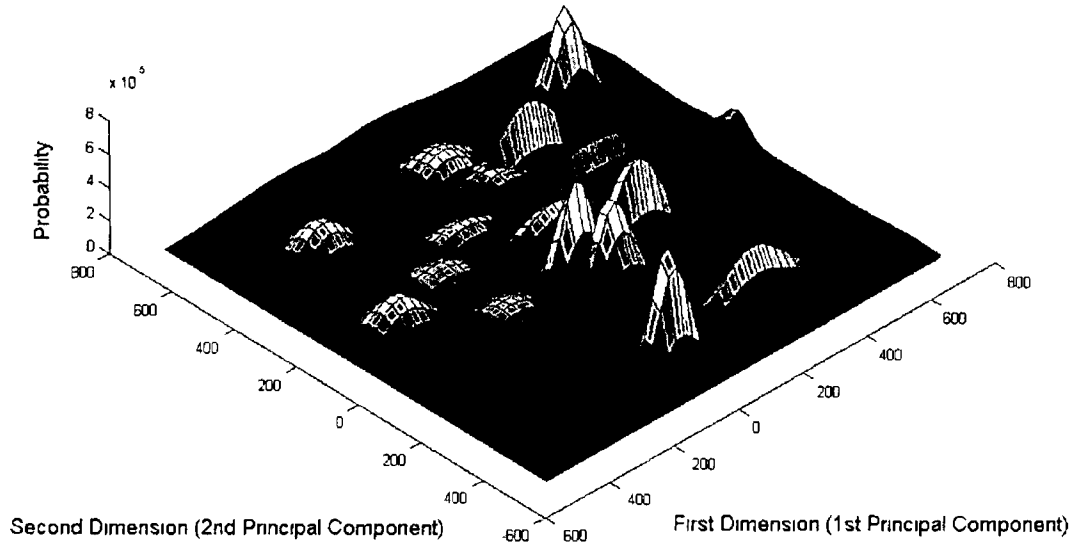


Figure 4 15 The HyperClass of a gesture in an eigenspace

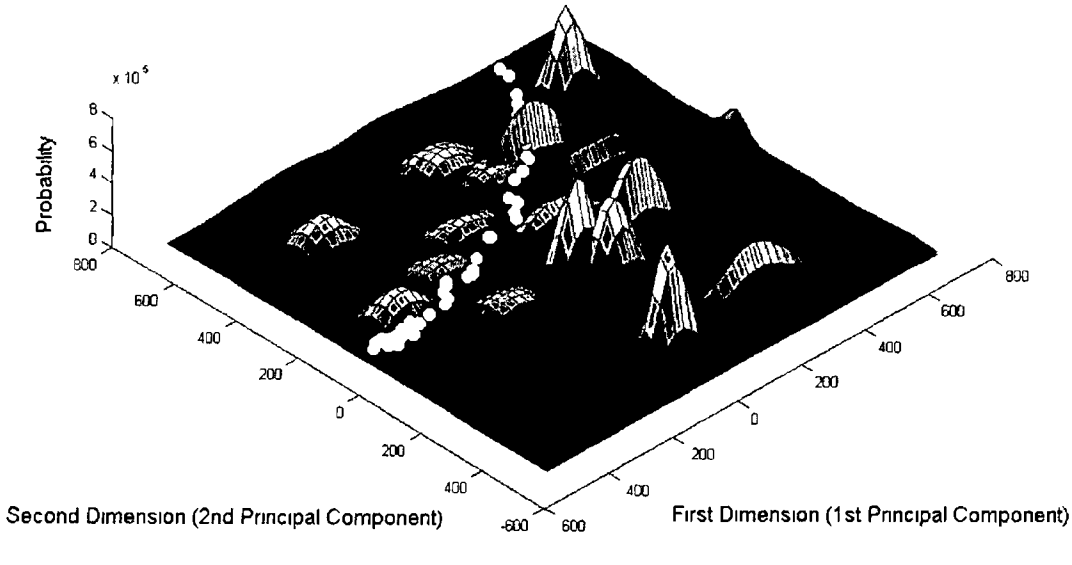


Figure 4 16 The 2 dimensional trajectory of a correct gesture in the eigenspace

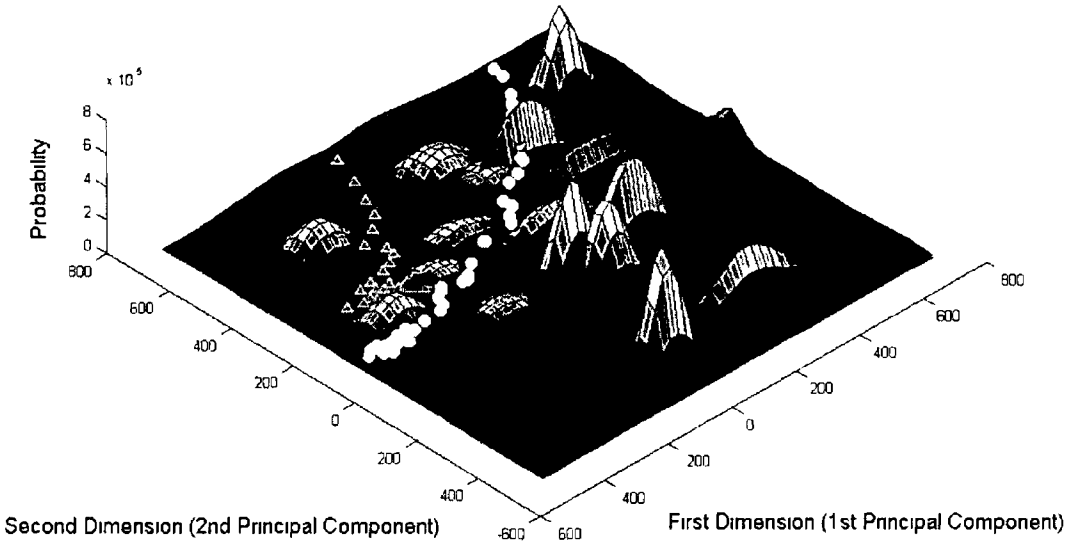


Figure 4 17 The trajectory of mother gesture in the eigenspace (the triangles)

In this figure most of the points (the triangles) are concentrated in a small part of the space and the other points are placed mostly in the valleys that do not match many of the gaussian distributions The results of matching this gesture in this eigenspace is as following;

Order of matched graph 2

which is a bipartite graph with 1 nodes on each partition

The likelihood of Equation 4 12 0 0625

In a second experiment we change the eigenspace to the eigenspace of the second gesture The trajectories of the first gesture and the second gesture in this eigenspace are shown in Figure 4 18 As can be seen the trajectory of the second gesture matches more gaussian distributions than the first one The results of matching the first gesture is as following,

Order of matched graph 8

which is a bipartite graph with 4 nodes on each partition

The likelihood of Equation 4 12 0 25

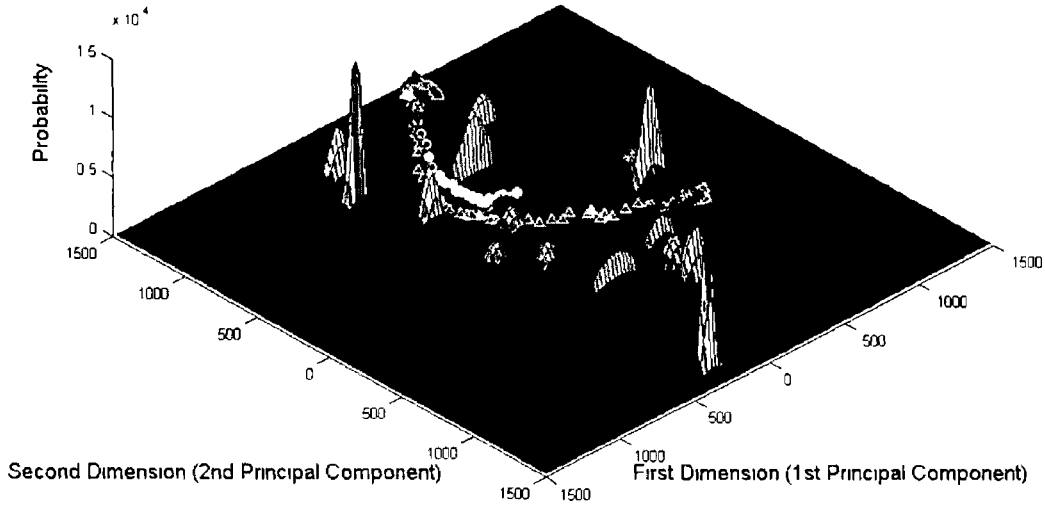


Figure 4 18 The trajectories of the first gesture (circles) and the second gesture (triangles) in the second eigenspace

However, the second gesture in the second eigenspace should result in a better match and likelihood The results are as follows,

Order of matched graph 14

which is a bipartite graph with 7 nodes on each partition

The likelihood of Equation 4 12 0 4375

Since the number of gaussian distributions in all the HyperClasses is the same the denominator of the Equation 4 12 gives an equal value for all the HyperClasses This denominator is actually used as a normalising factor It is shown in the experiments that a better match results in higher order of matched bipartite graph

However, in calculating the likelihoods, Equation 4 12, the mean of matched vertices probabilities is also involved This parameter is useful when the algorithm results in the same order of graphs for more than one HyperClass Therefore, the higher mean of the probabilities, the higher likelihood The results of the experiments show that the better match gives a higher likelihood

As can be seen the algorithm works very well in recognising the gestures In the next chapters we measure the recognition rate of the algorithm with a large database of gestures and compare the results with another algorithm

Summary and Conclusion

In this chapter we investigated the problem of dynamic gesture recognition An unsupervised clustering technique called Vector Quantisation was presented which is used in our algorithm for clustering the trajectories and manifolds of the gestures in the eigenspaces We also use this technique in future for the algorithms based on Hidden Markov Models

A new spatio-temporal pattern matching algorithm for dynamic gesture recognition was introduced In this algorithm for every gesture an eigenspace is constructed and a HyperClass of many multi-dimensional gaussian distributions is formed In order to recognise an unknown gesture we presented a Graph-Matching technique This technique matches the graph of a given gesture with the graphs of all the gestures in the vocabulary We showed that the algorithm works very well in recognising the dynamic gestures We also presented an experiment in which the best number of eigenvectors to form the eigenspaces was estimated

In the next chapters we look into the problem of dynamic gesture recognition in more detail with the large vocabulary of gestures We will compare the recognition rate and processing speed of the gaussian Graph-Matching algorithm with another algorithm using Hidden Markov Models

Chapter 5

MOTION TRACKING AND A DYNAMIC MODEL

In a bimanual movement recognition system when the two hands are moving together an important problem is to track the motion of each hand individually. Detecting occlusion is an important problem in hand tracking. By tracking the hands the system will be able to detect hand occlusions and get the occlusion parts of a bimanual movement under control. Otherwise, the occlusions may happen at any time and the system cannot recognise the occluded hand shapes from non-occluded ones. On the other side, a hand may leave the scene by leaving the image frame or hiding behind a part of body. In that case, only one hand shape is detected in the images.

Therefore, we need a tracking system to keep eyes on the motions of the hands and recognise the hand occlusions and the other types of movements a hand can do. For this we use an algorithm called Kalman filter and a dynamic model for motion tracking.

We use this model in tracking single-hand motions. Based on this model we are able to approximately detect the beginning of a gesture belonging to the meaningful part of a movement. In Chapter 7 we employ the dynamic model to track the movements of the two hands and detect the hand occlusions for correctly tracking the hands during a bimanual movement.

First, the Kalman filtering algorithm is presented very briefly. Based on the Kinematic equations of movement we present the dynamic model for tracking a hand movement. By using this model we detect the beginning of a gesture by looking at the velocity changes of the hand. We present some experimental results at the end of chapter to demonstrate the performance of the proposed dynamic model in correctly tracking different hand movements.

5 1 Kalman Filter

"Kalman filtering is an optimal state estimation process applied to a dynamic system that involves random perturbations" [Chui 1999] The Kalman filter gives a linear, unbiased, and minimum error variance recursion algorithm to optimally estimate the unknown state of a dynamic system from noisy data taken at discrete real-time

Kalman filter is used in many areas in science and engineering Real-time tracking of a flying object, estimating the planar orbit of a satellite, target tracking [Chui 1999] and Global Positioning System [Brown 1997] are well-known applications of Kalman filter

We use Kalman filter to track the correct position of the hand in a sequence of images Not only the position of the hand but also the velocity and acceleration are the important parameters in our hand tracking problem Using the Kalman filter we track the position, velocity and acceleration of a hand in an image sequence in order to detect hand pauses, collisions, occlusions and appearance and disappearance of the hands

The complete description of the Kalman filter algorithm is presented in Appendix C, Section C 1 The algorithm is briefly shown in Figure 5 1

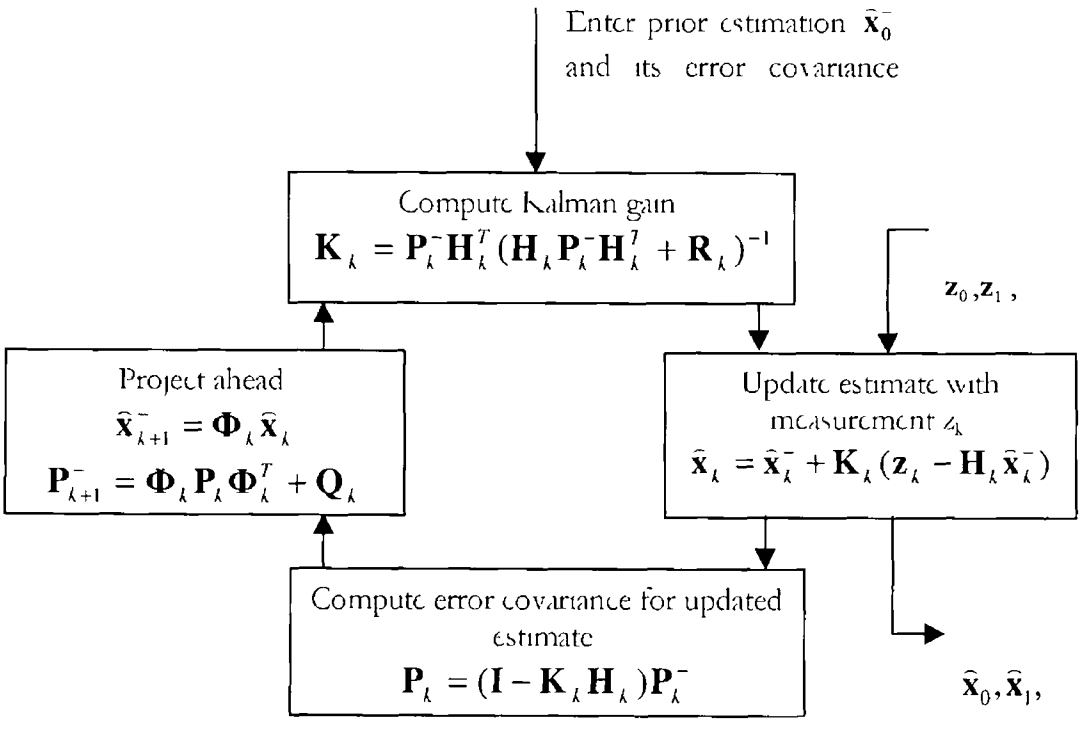


Figure 5 1 An illustration of Kalman filtering algorithm from [Brown 1997]

5 2 Hand Motion Tracking

Researchers have used different methods for tracking hands in an image sequence Some of them model the hands by some geometrical descriptions such as circles and cylinders [McAllister 2002 Davis 1999] Others have used other techniques such as Bayesian Networks and CONDENSATION to track the hands [Gong 2002 Isard 1998a] Kalman filter, however, has been given great attention of researchers for tracking objects such as airplanes, satelites, human, etc [Brown 1997 Kohler 1997 Chui 1999 Bowden 2000, Dockstader 2000] Important factors of Kalman filter are the fast processing and flexibility to accept different dynamic models Therefore, we use this technique for hand tracking

In order to track the hand we use a model in which the position, velocity and acceleration of hand are modelled by a Kalman filtering process This model is based on the Kinematic equations of motion Chen et al [Chen 2003] and Zieren et al [Zieren 2002] have used this model to track the fingers and hands in surgical and frontal view gesture recognition applications

Let $v_{(t)}$ denote the trajectory of hand movement where t is the time variable This function is discretised by sampling with $f = \frac{1}{h}$, $h > 0$ where f is sampling rate, and h is the sample interval Therefore,

$$x_k = v_{(kh)} \quad k = 0,1,$$

$x_{(t)}$ can be assumed to have continuous first and second order derivatives Where the $x_{(t)}$ is position, the first and second derivatives of $x_{(t)}$ are the velocity and acceleration respectively For small values of h the position and velocity vectors are calculated by,

$$x_{k+1} = x_k + hv_k + \frac{1}{2}h^2 a_k \tag{5 1}$$

$$v_{k+1} = v_k + ha_k \tag{5 2}$$

where

v_k velocity – the first derivative

a_k acceleration – the second derivative

$x_k = v_{(kh)} \qquad k = 0,1,$

$x_k = a_{(kh)} \qquad k = 0,1,$

We define the position of hand by the centre of the rectangle around it (see Figure 5 2) This rectangle represents the leftmost, rightmost, top and bottom of the hand's blob in the image Therefore, the position is denoted by the vector,

$$\mathbf{x}_k = \begin{bmatrix} x_k^h \\ x_k^v \end{bmatrix} \tag{5 3}$$

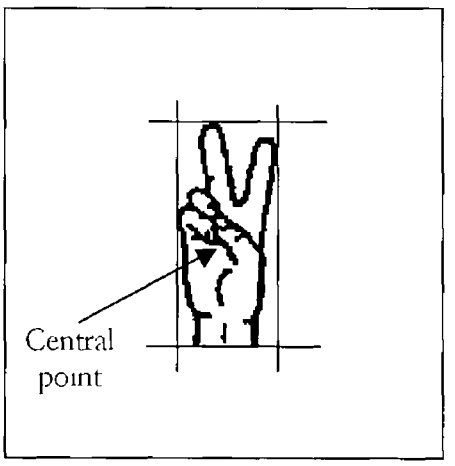


Figure 5 2 Position of the hand is defined by the centre of the rectangle

where

x_k^h the horizontal coordinate of the hand centre,

x_k^v the vertical coordinate of the hand centre

However, we can only observe the position of the hand in an image while the other parameters, velocity and acceleration, are not observable Therefore, the matrix $\mathbf{H}$ in Equation C 2 (Appendix C) is defined as,

$$\begin{aligned} \mathbf{H} &= [1 \quad 0 \quad 0] \\ \mathbf{z}_k &= H\,\mathbf{x}_k \end{aligned} \tag{5 4}$$

The hand-tracking model takes on the following linear stochastic description,

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{w}_k \\ \mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \end{cases} \tag{5.5}$$

In this system, the parameters are given by,

$$\mathbf{x}_k = \begin{bmatrix} x_k^h \\ \dot{x}_k^h \\ \ddot{x}_k^h \\ x_k^v \\ \dot{x}_k^v \\ \ddot{x}_k^v \end{bmatrix}, \qquad \mathbf{\Phi} = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 & & & \\ 0 & 1 & h & & & \\ 0 & 0 & 1 & & & \\ & & & 1 & h & \frac{1}{2}h^2 \\ & & & 0 & 1 & h \\ & & & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

and $\mathbf{v}_k$ and $\mathbf{w}_k$ are independent zero-mean Gaussian *white noise*[1] sequences with the following covariances,

$$E[\mathbf{w}_k \mathbf{w}_l^T] = \begin{cases} \mathbf{Q}_k & l = k \\ 0 & l \neq k \end{cases}$$

$$E[\mathbf{v}_k \mathbf{v}_l^T] = \begin{cases} \mathbf{R}_k & l = k \\ 0 & l \neq k \end{cases}$$

We assume that

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_k^1 & 0 \\ 0 & \mathbf{Q}_k^2 \end{bmatrix}, \qquad \mathbf{R}_k = \begin{bmatrix} \mathbf{R}_k^1 & 0 \\ 0 & \mathbf{R}_k^2 \end{bmatrix}$$

[1] White noise is defined to be a stationary random process with constant spectral density function,

$$S_{white-noise}(j\omega) = A$$

where A is the spectral amplitude of the white noise [Brown 1997]

Let

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^1 \\ \mathbf{x}_k^2 \end{bmatrix}$$

where $\mathbf{x}_k^1$ and $\mathbf{x}_k^2$ are defined to be the vectors of the horizontal and vertical attributes of the hand's central point with the following definitions,

$$\mathbf{x}_k^1 = \begin{bmatrix} x_k^h \\ x_k^h \\ x_k^h \end{bmatrix}, \qquad \mathbf{x}_k^2 = \begin{bmatrix} x_k^1 \\ x_k^v \\ x_k^v \end{bmatrix}$$

$$\mathbf{w}_k = \begin{bmatrix} \mathbf{w}_k^1 \\ \mathbf{w}_k^2 \end{bmatrix}, \qquad \mathbf{v}_k = \begin{bmatrix} v_k^1 \\ v_k^2 \end{bmatrix}$$

where $\mathbf{w}_k^1$, $\mathbf{w}_k^2$, v_k^1, and v_k^2 are zero-mean Gaussian white noise sequences with the following covariances,

$$E[\mathbf{w}_k^1 \mathbf{w}_i^{1T}] = \begin{cases} \mathbf{Q}_k^1 & i = k \\ 0 & i \neq k \end{cases} \qquad E[\mathbf{w}_k^2 \mathbf{w}_i^{2T}] = \begin{cases} \mathbf{Q}_k^2 & i = k \\ 0 & i \neq k \end{cases}$$

$$E[v_k^1 v_i^{1T}] = \begin{cases} R_k^1 & i = k \\ 0 & i \neq k \end{cases} \qquad E[v_k^2 v_i^{2T}] = \begin{cases} R_k^2 & i = k \\ 0 & i \neq k \end{cases}$$

Let,

$$\mathbf{A} = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{z}_k = \begin{bmatrix} z_k^1 \\ z_k^2 \end{bmatrix}$$

$$\mathbf{C} = [1 \quad 0 \quad 0]$$

Then the system of 5 5 can be decomposed into two subsystems,

$$\begin{cases} \mathbf{x}_{k+1}^{j} = \mathbf{A}\,\mathbf{x}_{k}^{j} + \mathbf{w}_{k}^{j} \\ z_{k}^{j} = \mathbf{C}\,\mathbf{x}_{k}^{j} + v_{k}^{j} \end{cases} \qquad j = 1,2 \tag{5 6}$$

where $\mathbf{w}_k^j$ and $\mathbf{x}_k^j$ are 3-vectors and z_k^j and v_k^j are scalars, $\mathbf{Q}_k^j$ is a 3x3 non-negative definite symmetric matrix and $R_k^j > 0$ is a scalar

Each subsystem is employed to model the hand movement in a direction, horizontal or vertical Assuming,

$$\begin{bmatrix} x_k^d(1) \\ x_k^d(2) \\ x_k^d(3) \end{bmatrix} = \begin{bmatrix} x_k^d \\ v_k^d \\ a_k^d \end{bmatrix}$$

where d is the horizontal or vertical direction ($d = h, v$), for one of the subsystems the tracking model is described as,

$$\begin{cases} \begin{bmatrix} x_{k+1}(1) \\ x_{k+1}(2) \\ x_{k+1}(3) \end{bmatrix} = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k(1) \\ x_k(2) \\ x_k(3) \end{bmatrix} + \mathbf{w}_k \\ z_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k(1) \\ x_k(2) \\ x_k(3) \end{bmatrix} + v_k \end{cases} \tag{5 7}$$

where we have suppressed the superscript d

If the initial condition $E(x_0)$ and $Variance(x_0)$ are given, the Kalman filtering algorithm for this model can be obtained Details of the calculations are presented in Appendix C, Section C 2

The dynamic model presented here is used in two forms Here in, we use this model to track the movement of a hand in a sequence of images in order to track the position, velocity and acceleration of hand In Chapter 7 we use this model to track the rectangle sides around the hands in order to detect the hand pauses and collisions during hand occlusions

By tracking the movement of a hand we can detect the moments that the hand has a pause in each of the horizontal or vertical directions By detecting the first pause the beginning of a gesture can be detected First we assume that a hand movement starts from a neutral position This condition can be the hand hanging by the side or the hand on lap We consider this as the hand out of camera frame When a person performs a hand gesture he/she moves the hand up and starts the gesture (see Figure 5 3)

Before the gesture starts the hand should move into the Region of Interest (ROI) Every gesture starts with a particular shape of hand So, normally, from the neutral position to the start point of the gesture the position and the shape of hand is changed rapidly toward the beginning position and shape of the gesture (see Figure 5 3) At a moment when the hand reach the correct position and shape to start the gesture the hand has a short pause at least in one of the directions, horizontal or vertical, but usually in both, and then it moves toward the end of gesture In order to detect the moment that the hand has a short pause to start the gesture two different ways are introduced

(a) (b) (c) (d)

Figure 5 3 (a) Hand in a rest position, (b) moving the hand toward the beginning of the gesture (c) the beginning of gesture (d) the end of gesture

1 The distance between the current position of hand, $\bar{x}_k(1)$ in Equation 5 7 and the previous position, $\bar{x}_{k-1}(1)$ gives a good approximation of the hand movement In a two-dimensional image frame the Euclidean distance between the previous position and the consequent one is a good approximation,

$$d = \sqrt{\sum_{i=1}^{2} (\hat{x}_k^i(1) - \hat{x}_{k-1}^i(1))^2} \qquad (5\,8)$$

If this distance is equal to zero we can conclude that the hand has a pause However, due to noise and the fact that in the low speed cameras the sampling rate of the camera may

prevent the capturing of two consequent frames with the same position of hand this way cannot be robust in real-time applications

2 A better detection factor is the velocity of the hand When the hand pauses the velocity reaches zero However, in fact, a well-chosen small threshold gives appropriate detection accuracy In a two-dimensional image frame,

$$v_d = \sqrt{v_h^2 + v_v^2} \qquad (5\,9)$$

where,

v_h horizontal velocity

v_v vertical velocity

For a small chosen $\varepsilon > 0$ if $v_d < \varepsilon$ we conclude that the hand has paused In most of the gestures even a pause in one of the directions is enough to detect the beginning of a gesture Because when the hand moves toward the beginning of the gesture it moves along the shortest diagonal path while the hand shape is changing to the beginning shape Therefore, at the beginning of the gesture at least in one of the directions, horizontal or vertical, the hand has a pause[2] From the pause point the system records the beginning of the hand gesture

5 3 Experimental Results

In order to investigate the introduced dynamic model some experimental results are presented In these experiments the velocity and the acceleration of the hand are of interest We demonstrate the performance of the dynamic model in tracking the velocity and acceleration of the hand

In the first example a normal hand movement is performed to detect the hand pauses in both horizontal and vertical directions In the next example, a circular movement of hand is performed to investigate the performance of the model

The distance of the camera and the hand is about 250cm The camera is filming a 70-by-70cm area approximately The resolution of the camera is set to 400-by-200 pixels Each

[2] This is a simplification assumption In many gestures a short pause happens but this does not include all gestures

pixel on the vertical axis corresponds to 3 5mm and on horizontal axis to 1 75mm approximately on the filming area The camera is set to work in 30 frames per second Thus, the time unit between two consecutive images is 33 3 milliseconds,

$$u = 33\ 3\ ms$$

This means that a velocity of 1 pixel per time unit (33 3 ms) is equal to 30 pixels per second On the vertical axis, 30 pixel per second is equal to 105 mm (10 5 cm) per second On the horizontal axis 30 pixel per second is equal to 52 5 mm (5 25 cm) per second

In the following experiments, the measurements are based on the time unit (u) and pixel per time unit

5 3 1 Experiment 1

In this experiment the hand starts from its position on lap and moves up and does the gesture (see Figure 5 4) Figure 5 5 shows the graph of the horizontal velocity of the hand

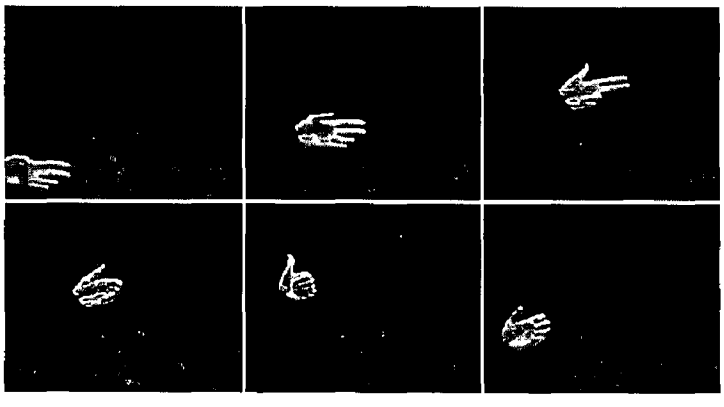


Figure 5 4 A few frames of the gesture in the Experiment 1 In this example the hand moves into the scene has a short pause at the beginning of the gesture and then moves toward the end of gesture

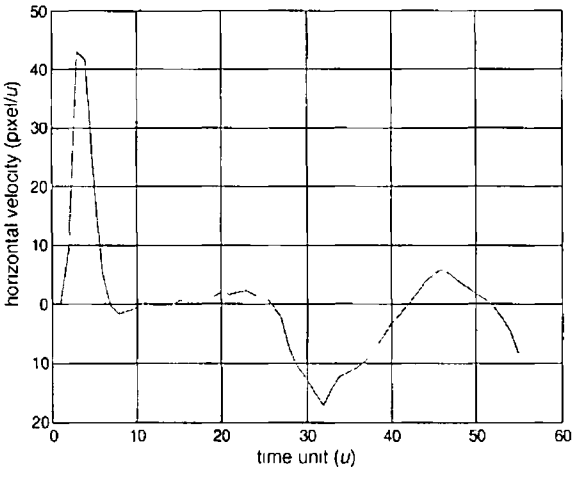


Figure 5 5 The graph of the horizontal velocity of the hand during the movement of Experiment 1

In this figure, somewhere between the 10^{th} to 20^{th} time unit (frame) the velocity goes to zero for the first time We do not consider the velocity of hand in the first frame because it always starts from zero The interesting point in this figure is the negative velocity When the hand moves in the decreasing direction of the horizontal axis the horizontal velocity is negative

The graph of the vertical velocity of the hand is shown in Figure 5 6 The negative velocity at the beginning of movement in this figure shows that, the hand has had a movement in the decreasing direction of the vertical axis (up direction)

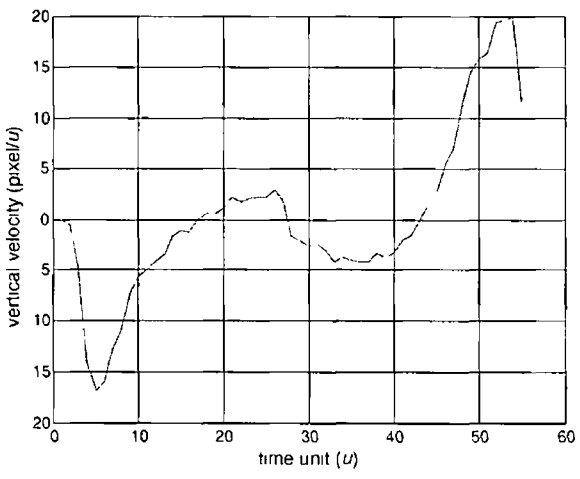


Figure 5 6 The graph of the vertical velocity of the hand during the movement of Experiment 1

Equation 5 9 finds the exact point that the velocity of hand in both directions, horizontal and vertical, becomes zero The graph of this equation is shown in Figure 5 7 This graph shows that at the 17^{th} time unit (frame) v_d becomes almost zero This point is detected as the beginning of the gesture

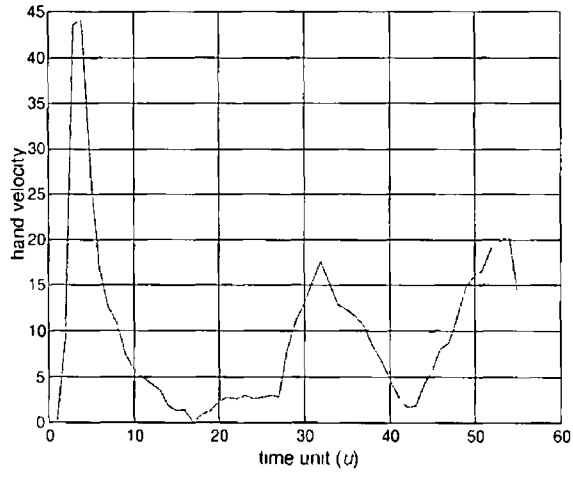


Figure 5 7 The graph of Equation 5 9 for the first experiment

5 3 2 Experiment 2

In this experiment we demonstrate the performance of the proposed model in tracking the hand velocity and acceleration

First the hand has a circular movement It circulates for several times and then stops and does the main gesture Figure 5 8 shows a few frames of this gesture

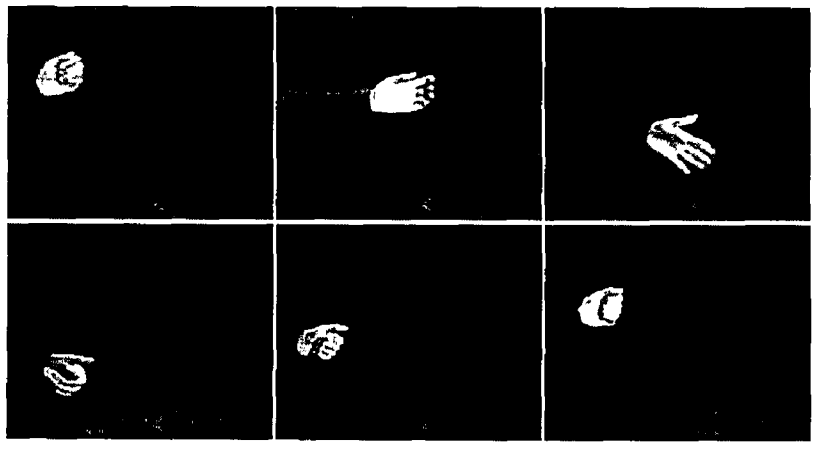
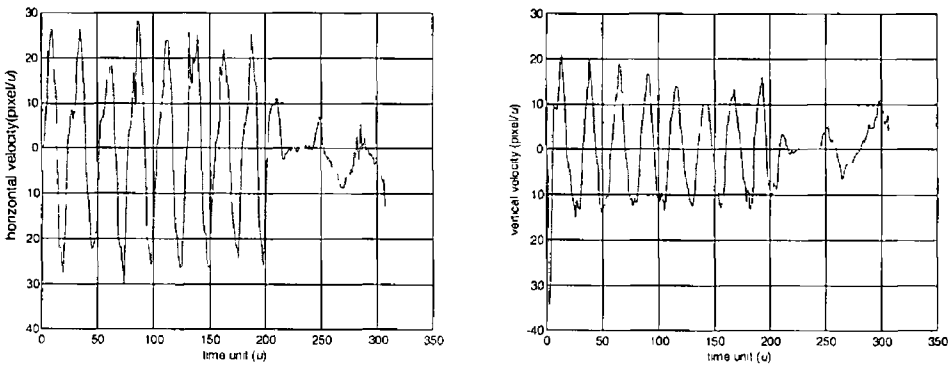


Figure 5 8 A few images of the movement of hand in a circular fashion

As long as the hand is circulating the horizontal and vertical velocities are oscillating However, at the pause point both become zero and the algorithm detects the pause Figure 5 9 shows the graph of horizontal and vertical velocities for the hand An accurate look at the horizontal and the vertical velocities shows that because of the circular movement of the hand, at the points that the horizontal velocity is at one of the minima or maxima the vertical velocity is passing through zero and vice versa (see Figure 5 10) This is a 90-degree phase difference that causes an oscillation with double frequency of oscillation in the Equation 5 9 However, it does not go to zero during the oscillations In Figure 5 11 the graph of Equation 5 9 is sketched The double frequency of oscillation is clearly visible in this graph



(a) (b)

Figure 5 9 Graphs of (a) the horizontal and (b) the vertical velocities of the circulating hand

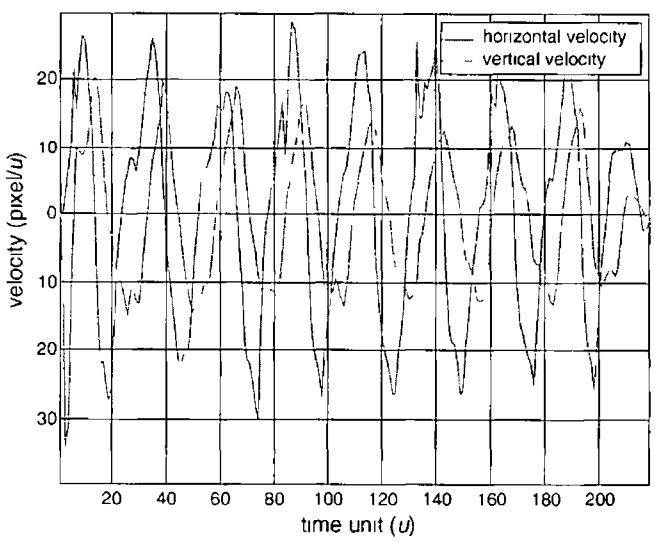


Figure 5 10 Graphs of the horizontal and vertical velocities of circulating hand 90 degree phase difference is observable

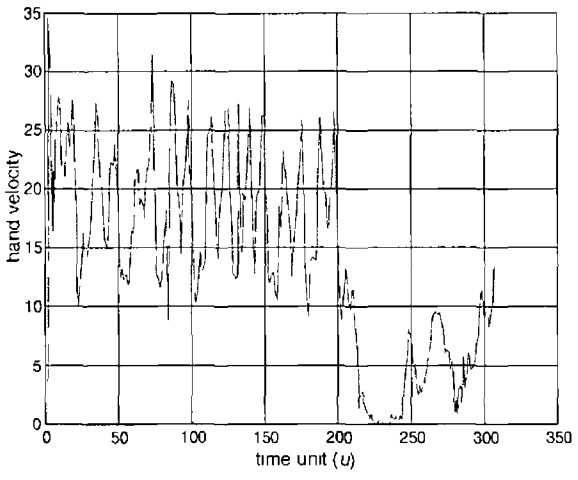


Figure 5 11 Graph of Equation 5 9 for the circulating hand

In this experiment, the calculated speed in Equation 5 9 becomes zero at the 236th time unit (frame) The accelerations of the hand are plotted in Figure 5 12 These graphs show that the model tracks the acceleration of the hand correctly in both directions

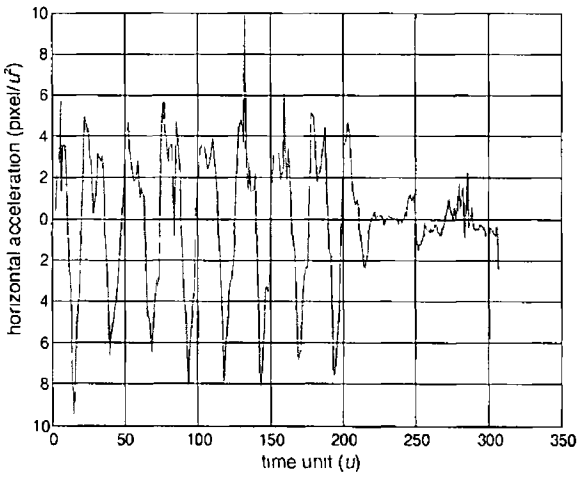
We demonstrated that the proposed model tracks the hand, particularly the velocity and the acceleration accurately in order to detect the hand pauses in horizontal and/or vertical direction This model is the basis of a model, which is introduced in Chapter 7 for hand occlusion detection and hand tracking in the presence of occlusion

Summary and Conclusion

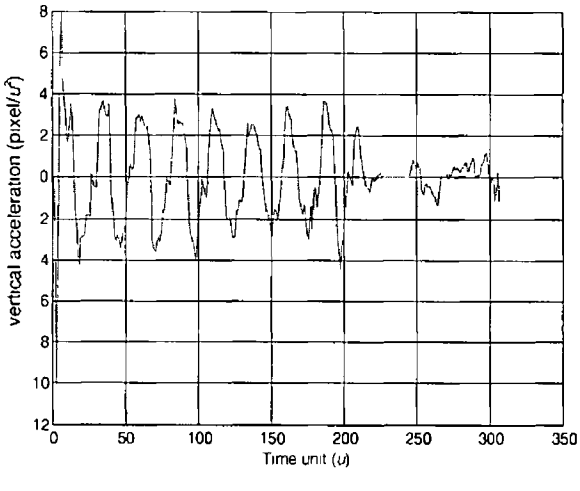
In this chapter we proposed a dynamic model based on the Kinematics equations of motion for hand tracking First we stated the general theory of Kalman filtering Then the details of the proposed dynamic model were presented Based on this model the position, velocity and the acceleration of the hand are tracked in the sequences of images containing a hand movement

Using the model, we are able to detect the hand pauses in both horizontal and vertical directions We detect the hand pauses to detect the beginning of a gesture during a hand movement Some experimental results were presented to demonstrate the effectiveness of the algorithm in correct tracking the velocities and accelerations

In the following chapter, we will use this model to detect the beginning of a gesture in order to extract the beginning shape of the gesture for recognising a large number of dynamic hand gestures



(a)



(b)

Figure 5.12 The horizontal and vertical acceleration of the circulating hand

Chapter 6

RECOGNITION OF A LARGE NUMBER OF HAND GESTURES

Dynamic gesture recognition as a spatio-temporal pattern recognition problem was presented in Chapter 4. An important parameter is the number of gestures in the vocabulary. A vocabulary containing a large number of gestures can cause a long searching and matching time. It is crucial to reduce the processing time of the algorithms while keeping the recognition rate as high as possible.

In this chapter we look into the problem of recognition of a large number of hand gestures. We introduce two hierarchical algorithms for the recognition of canonical hand gestures. First we review some statistical techniques addressed in the literature for the problem of recognition. Hidden Markov Models are a very well known statistical technique for modelling temporal events. We review the theory of this model. Then an algorithm based on this model is presented for the recognition of large number of canonical gestures. We replace the Hidden Markov Models in the algorithm with the gaussian Graph-Matching technique of Chapter 4 and present a new algorithm. At the end of chapter some experimental results are presented in which the recognition rate of both the algorithms are measured and compared. We will also compare the relative processing time of the algorithms.

6.1 Markov Chain

"A Markov chain deals with a group of random processes that incorporate a minimum amount of memory without actually being memoryless" [Jelinek 1997]. In this thesis we deal with discrete random variables. The values of these variables are defined in a finite alphabet $H = \{1, 2, \quad , M\}$.

The probability of observing $X_1, X_2, \quad , X_n$ is defined by the Bayes' rule as,

$$P(X_1, X_2, \quad , X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, X_2, \quad , X_{i-1}) \tag{6 1}$$

However, if a variable depends only on the value of the previous step,

$$P(X_i \mid X_1, X_2, \quad , X_{i-1}) = P(X_i \mid X_{i-1}) \tag{6 2}$$

the random variables are said to form a Markov chain Therefore, for a Markov chain,

$$P(X_1, X_2, \quad , X_n) = \prod_{i=2}^{n} P(X_i \mid X_{i-1}) \tag{6 3}$$

This process only has one step memory A Markov chain is time invariant (stationary) if regardless of the value of the time index i,

$$P(X_i = x' \mid X_{i-1} = x) = p(x' \mid x) \quad for all \ \ x, x' \in H \tag{6 4}$$

where

$$\sum_{x \in H} p(x' \mid x) = 1, \quad p(x' \mid x) \geq 0$$

$x' \in H$

$p(x' \mid x)$ is called the transition function

If we think of the values of X_i as states then the Markov chain is a finite state process with transition between states specified by the function $p(x' \mid x)$ Figure 6 1 shows a three-state Markov chain In this figure, the arrows show transitions between states and their probabilities The missing arrows imply that $p(1 \mid 2) = p(2 \mid 3) = 0$

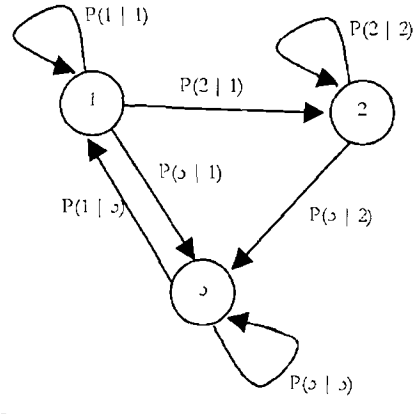


Figure 6 1 A transition diagram for a Markov chain

6 2 Hidden Markov Models

A Hidden Markov Model (HMM) is a finite set of states each of which is associated with a probability distribution In the other words, a Hidden Markov Model is a tool for representing probability distributions over a sequence of observations [Ghahramani 2001] We denote the observation at time t by the variable y_t This variable can take values from a finite discrete alphabet, set of real values, or any other set as long as we can define a probability distribution over it In a discrete model the observations are sampled at discrete, equally spaced time intervals

The term "hidden" in Hidden Markov Models refers to this assumption that the observation at time t was generated by some process whose state S_t is hidden from the observer Further, we assume that the state of this hidden process satisfies the Markov property, Equation 6 3

Mathematically, for an output alphabet $y = \{0,1, \quad, M-1\}$ a state space $S = \{0,1, \quad, N\}$ with a unique starting state s_0 a probability distribution of transitions between states $p(s' \mid s)$ and an output probability distribution $p(y \mid s)$ associated with state s, the probability of observing an HMM output string $y_1, y_2, \quad, y_k$ is given by,

$$P(y_1, y_2, \quad, y_k) = \sum_{s_1 \quad s_k} \prod_{i=1}^{k} p(s_i \mid s_{i-1})\, p(y_i \mid s_i) \qquad (6\ 5)$$

The initial probabilities are also involved in calculating the probability of observations and will be described later on in the description of the technique

An example of an HMM with $N=3$ is shown in Figure 6 2

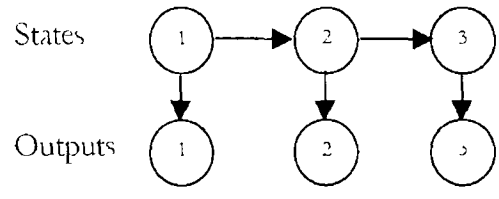


Figure 6 2 A three state Hidden Markov Model Source [Ghahramani 2001]

In addition to the previous two assumptions, Markov and stationarity, we assume that current output (observation) is statistically independent of the previous outputs For a sequence of observations $Y = y_1, y_2, \quad , y_T$ the probability of the sequence is given by,

$$p(Y \mid s_1, s_2, \quad , s_T) = \prod_{t=1}^{T} p(y_t \mid s_t) \tag{6 6}$$

However, this assumption has a very limited validity In some cases this assumption is not fair enough and therefore become a severe weakness of the HMMs We define an HMM completely by $\lambda = (A, B, \pi)$ where $A = \{a_{ij}\}$ is a set of state transition probabilities,

$$a_{ij} = p(s_{t+1} = j \mid s_t = i) \qquad 1 \le i, j \le N \tag{6 7}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad 1 \le i \le N$$

where N is the number of states of the model and s_t denotes the current state, $B = \{b_j(k)\}$ is the probability distribution in each state,

$$b_j(k) = p(y_k \mid s_t = j) \qquad 1 \le k \le M \tag{6 8}$$

$$b_j(k) \ge 0, \quad \sum_{k=1}^{M} b_j(k) = 1 \qquad 1 \le j \le N$$

where y_k is the k^{th} observation symbol in the alphabet $\pi = \{\pi_i\}$ is the initial state distribution,

$$\pi = p(s_1 = i) \qquad 1 \le i \le N \tag{6 9}$$

The three basic problems of HMMs are

1 The evaluation problem

Given an HMM λ and a sequence of observations $Y = y_1, y_2, \ldots, y_T$, what is the likelihood that Y is generated by the model In other words, what is the probability of Y given HMM λ,

$$P(Y \mid \lambda) = ?$$

Calculation of this probability requires a number of operations at the order of N^T This is a very time consuming process However, there exists a method called the *Forward/Backward* algorithm with considerably lower complexity This algorithm gives the probabilities to all possible state change sequences in an HMM Regardless of the most probable state sequence it finds the likelihood with respect to the probabilities along all possible paths Based on [Ghahramani 2001], we use this technique to calculate the likelihoods that the different HMMs of gestures in our database generate the given sequence of observations without extracting the sequence of state changes A description of the algorithm is presented in Appendix D

2 The decoding problem

Given a model λ and a sequence of observations $Y = y_1, y_2, \ldots, y_T$, what is the most likely state sequence in the model that produced Y In other words, find a state sequence that maximises the numerator of the right-hand side of the following equation

$$P(s_1, s_2, \ldots, s_T \mid y_1, y_2, \ldots, y_T, s_0) = \frac{P(s_1, s_2, \ldots, s_T, y_1, y_2, \ldots, y_T \mid s_0)}{P(y_1, y_2, \ldots, y_T \mid s_0)}$$

The *Viterbi* algorithm is a technique to find the most probable state sequence By this algorithm we can find the maximum probability state sequence of an HMM given the sequence of observations A complete description of the Viterbi algorithm is presented in [Jelinek 1997]

Although, the evaluation and decoding in HMMs can be used interchangeably in many applications, we only use the evaluation problem in order to calculate the likelihood that a given HMM generates a sequence of observations [Ghahramani 2001]

3 The learning problem

Given a model λ and a sequence of observations $Y = y_1, y_2, \ldots, y_T$, how should the model parameters $\{A, B, \pi\}$ be adjusted in order to maximise $P(Y \mid \lambda)$ There are many techniques proposed in the literature for estimating the statistical parameters of the Hidden Markov Models [Jelinek 1997] *Expectation Maximisation* and *Maximum Entropy* are well-known methods in this case We use an algorithm based on Expectation Maximisation called *Baum Welch* Details of this algorithm are presented in Appendix D

The evaluation problem is used for isolated recognition in speech and gesture recognition The decoding problem is related to continuous recognition as well as to segmentation "Learning" is to adjust the HMM parameters for the recognition task

In this thesis we only deal with evaluation and learning problems, which are directly related to our work

6 3 Algorithms for Recognition of a Large Database of Hand Gestures

As we stated earlier in Chapters 2 and 4, many researchers have used Hidden Markov Models (HMM) for modelling a temporal sequence HMMs are widely used in speesh recognition [Jelinek 1997] as well as gesture recognition [Starner 1995a Starner 1995b Lee 1999 Nam 1996 Wilson 1999] Its wide applications and great performance in dealing with variations in data encourage us to use this technique for recognising a large set of hand gestures

HMMs form distributions over observations in every state Because of this similarity between HMMs and the gaussian Graph-Matching algorithm of Chapter 4, we compare the two techniques in this chapter to find the advantages and disadvantages of each

As in Chapter 4, by using pixel grey-level detection and segmentation one can extract the hand from background in an image Herein, it is assumed that in the sequence of input

images the hand has been segmented from the background This leads to the following definition of the problem

" Given a hand gesture appearing in a sequence of images, find a gesture in a large database of predefined gestures which is the most similar to that "

6 3 1 Special Considerations

Since we have a large database the similarity between the gestures is high Therefore, variations in the gestures may change the result of recognition The algorithms should be able to deal with both similarities and variations together This makes our job more difficult Also working with a large database of gestures involves extensive computation and long processing time

We introduce a hierarchical algorithm in which the Hidden Markov Models are used to deal with both variations and similarities The hierarchical nature of the algorithm makes the processing time shorter

6 3 2 A Quick Review of the Algorithm

In this algorithm we have a hierarchical recognition process that uses a multilevel trained model By using the dynamic model and Kalman filtering of Chapter 5 we are able to recognise the beginning of a gesture The first levels of the training and recognition phases of the algorithm are based on the beginning hand shape of the gestures At this level, depending on the beginning shape of an input gesture a group of gestures within the database is selected This group is forwarded to the second level At the second level, by using either Hidden Markov Models or the gaussian Graph-Matching algorithm the best match between the input gesture and the forwarded gestures is found

By this hierarchy we are able to find the best match in a short time We use Principal Component Analysis to reduce the dimensionality of data and get rid of noise This reduces the running time too Hidden Markov Models are powerful in dealing with large vocabularies with great variations Graph-Matching is fast in dealing with small number of gestures We compare the advantages and disadvantages of the two algorithms in the second level

6 3 3 Training Phase

By extracting a few beginning hand shapes of all the gestures a common eigenspace is constructed for the beginning shapes The projection of the beginning shapes into this eigenspace forms clusters of data points Each cluster represents a group of gestures starting with similar hand shapes Therefore, we get as many clusters as different shapes If we do not know the beginning shapes of gestures a clustering technique like Vector Quantisation can cluster the data points But, if we know that each gesture starts with what hand shape the clustering is straightforward Figure 6 3 shows clusters of points corresponding to the hand shapes for the English letters in the Irish Sign Language We use each cluster of points to form a multidimensional gaussian distribution The seven-dimensional data points in the common eigenspace are employed to train the gaussian distributions

In the second level we have two choices, the HMMs and the gaussian Graph-Matching

- **The Hidden Markov Models**

A common eigenspace is formed by using the full-length image sequences of all the gestures This eigenspace is constituted by the seven eigenvectors of the covariance matrix made from the whole training set image sequences The projection of each gesture into this subspace (eigenspace) forms a trajectory The projection of all the gestures into this subspace looks like a cloud of points In order to employ the HMMs we need to allocate codewords (codevectors) to the groups of points in this cloud Vector Quantisation algorithm extracts the required codewords

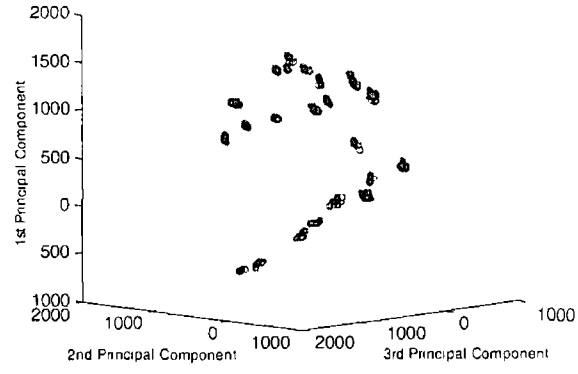


Figure 6 3 A 3 dimensional illustration of the eigenspace and the 26 clusters of hand shapes

A well-chosen number of codewords can represent all the points so that the gestures are exclusively recognisable by unique sequences of the codewords However, since the Vector Quantisation algorithm is a time consuming process, especially in the case of a very large number of data points, one can extract the codewords for the trajectory of each gesture separately as opposed to treating the whole data at once Then, by combining the extracted codewords and applying a second stage of Vector Quantisation a reasonable number of codewords is extracted Figure 6 4 shows the manifold and the extracted codewords for a gesture in a 3-dimensional representation of the common eigenspace A left-to-right Hidden Markov Model with four states is trained for every gesture (see Figure 6 5) The number of states in the HMMs is selected based on the shapes of the manifolds and the number of gestures Since the number of gestures is large the differences between the gestures should be considered Therefore, for example a 2-state HMM cannot distinguish different manifolds very well With respect to the shapes of the manifold (e g see Figure 6 4) a 4-states HMM seems suitable to model the gestures in this problem As a future work we can vary the number of states and look at the changes in the recognition rate of the algorithm

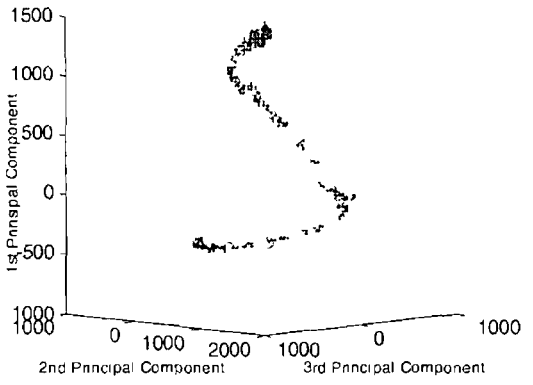


Figure 6 4 A 3D illustration of the manifold (dots) and the extracted codewords (small circles) in the common eigenspace

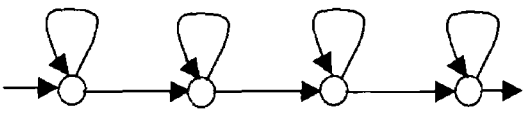


Figure 6 5 A left to right HMM is trained for every gesture

This is done by extracting the sequence of the symbols (codewords) constituting the trajectory of a gesture in the common eigenspace The set of codewords for all the HMMs are the same However, based on the sequences of codewords associated with each gesture in the training set the HMMs are trained An individual HMM is trained by the sequences extracted for all the examples of a gesture in the training set Therefore, at the end we have a trained HMM for every gesture in the vocabulary

- **The Gaussian Graph-Matching Algorithm**

An individual eigenspace is constructed for every gesture using the examples of the gesture By projecting the gestures into their own subspace the main manifolds of the subspaces are formed Then by using the Vector Quantisation and clustering the main manifolds the HyperClass of every eigenspace is formed and trained The algorithm is the same as the one in Chapter 4

6 3 4 Recognition Phase

In order to recognise an unknown gesture a 2-level hierarchy is used At the first level, a subset of gestures is selected to be passed to the second level (see Figure 6 6)

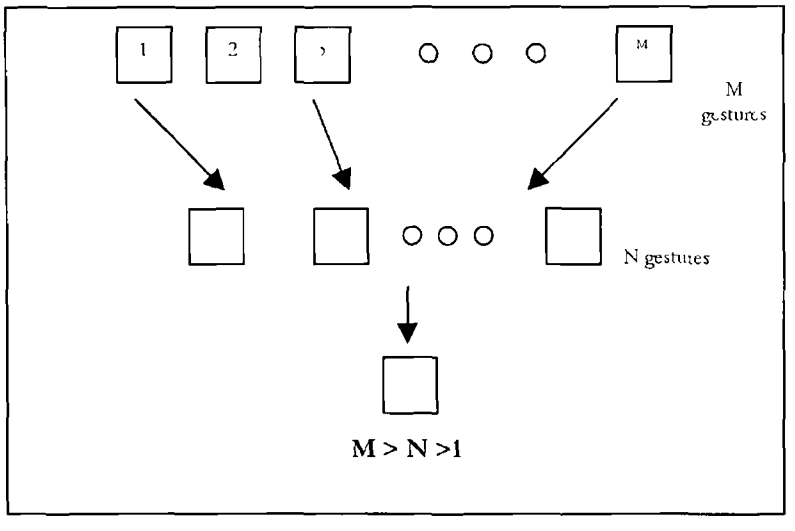


Figure 6 6 Hierarchy of selections in the recognition phase

Level 1

For an unknown gesture appearing in a sequence of images the hand shapes within the first few frames are extracted By using the tracking algorithm of Chapter 5 the beginning of a gesture can be detected By extracting a few initial hand shapes of a gesture and projecting them into the common eigenspace made in the first level of the training phase a few points are formed The centre of gravity of these points is calculated The probability that the central point belongs to each of the multidimensional gaussian distributions is calculated The point belongs to the class of shapes with the highest probability

However, due to variations and noise this is not always the best estimate and one should consider more classes Therefore, a list of all classes sorted descendingly based on the probabilities is constructed By taking α classes from the top of the list a group of gestures starting with the associated hand shapes are forwarded to the second level Since many gestures may start with the same shape of hand, the number of forwarded gestures is usually larger than α

Level 2

- **The Algorithm with Hidden Markov Models**

By projecting the input gesture into the common eigenspace formed in the second level of the training phase a sequence of symbols (codewords) is extracted The trained HMMs of the forwarded gestures are employed to calculate the likelihood of the extracted sequence

The HMM that results in the largest likelihood is the best match In other words, the gesture whose HMM results in the largest likelihood is the best match to the input gesture

- **The Algorithm with Graph-Matching**

By projecting the input gesture into the eigenspace of each of the forwarded gestures, formed in the second level of training, the trajectory of the gesture in each subspace is extracted As in Chapter 4 the trajectory is divided into an equal number of clusters and the graph of the trajectory is extracted in the eigenspaces By matching the graphs of the input gesture with the HyperClass of each gesture the best match is found

6 4 Exprimental Results

100 gestures were created by a combination of about 35 hand shapes mostly selected from the sign language alphabet The gestures start from a shape and end in another shape In Figure 6 7 a few gestures are shown For every gesture 10 examples were captured Half of the examples (500) were used as training set and the rest as the test set

The gestures start from one of the shapes defined as an English letter in the sign language For every gesture, the first 5 consecutive hand shapes were extracted and a common eigenspace was constructed using all the extracted hand shapes A seven-dimensional gaussian distribution was assigned to the similar hand shapes and trained 26 distributions for the set of beginning hand shapes associated with the 26 english letters were formed (see Figure 6 8)

In the second level of the training phase the 500 examples in the training set were employed to form a common eigenspace By projecting the training samples into this subspace (see Figure 6 9) and using Vector Quantisation at the first stage 3200 codewords were extracted, 32 for each gesture (see Figure 6 10)

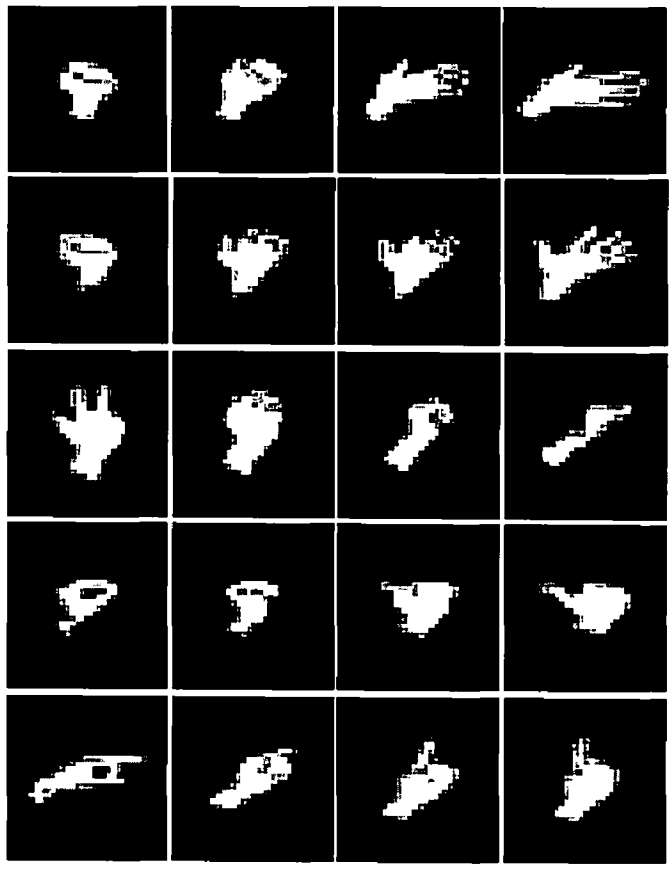


Figure 6 7 A few images of some of the created hand gestures

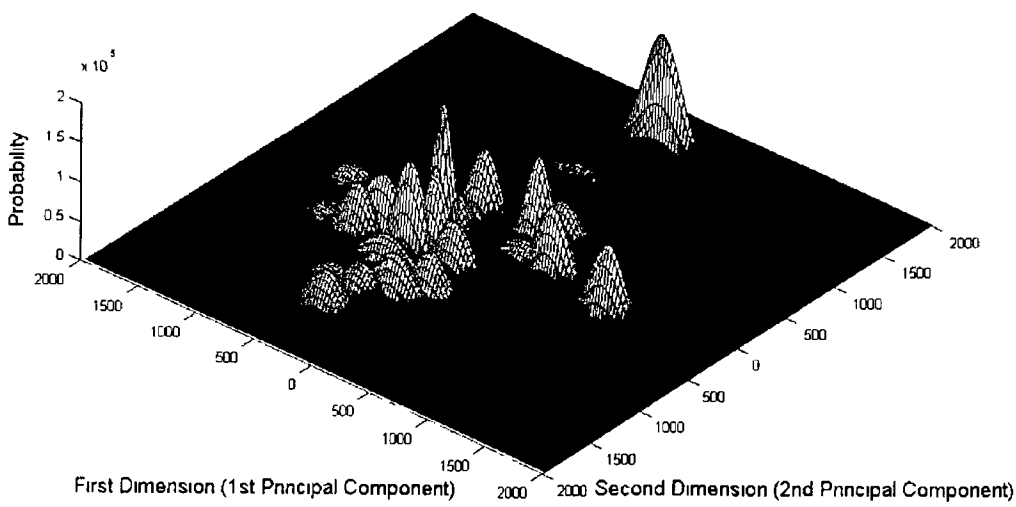


Figure 6 8 A 3 dimensional illustration of the 2D gaussian distributions of the beginning hand shapes

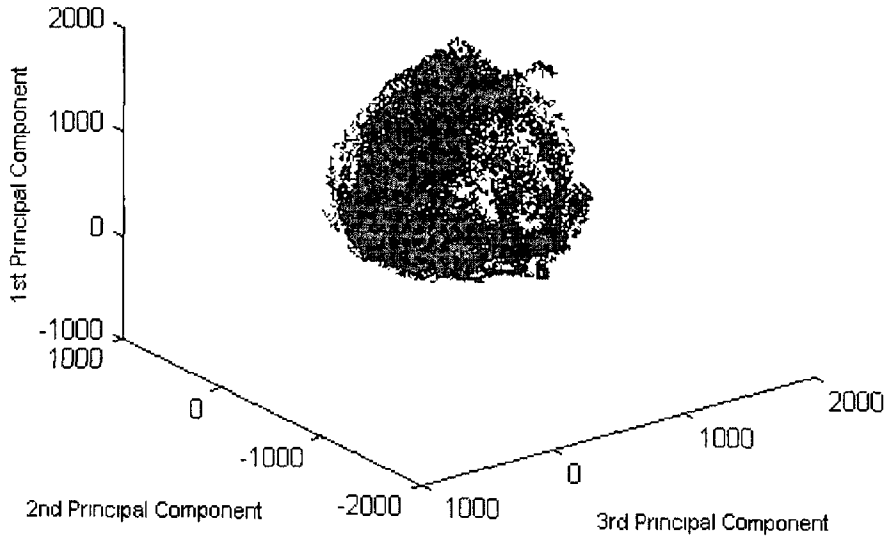


Figure 6 9 Projection of the training set (images) into the common eigenspace

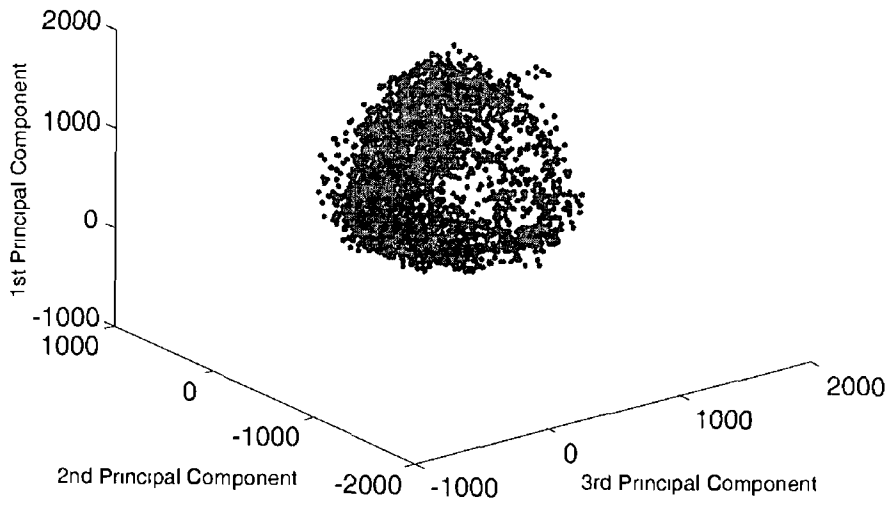


Figure 6 10 3200 extracted codewords for the projection of the images of the training set gestures

By applying the second stage of Vector Quantisation, 1024 codewords were extracted (see Figure 6 11) We have chosen this number of codewords based on the variation in the data appearing in the manifolds of the gestures and the processing speed[1] An example of variation in the examples of a gesture will be presented later Figure 6 12 shows the trajectory of a gesture projected into the eigenspace

Although we did not deliberately vary the samples of each gesture, the trajectories of the gestures show significant variations in the samples of a gesture (see Figure 6 13) In this figure, 5 examples of a gesture are projected into the eigenspace Four different paths shown in the figure demonstrate the great variations in the samples

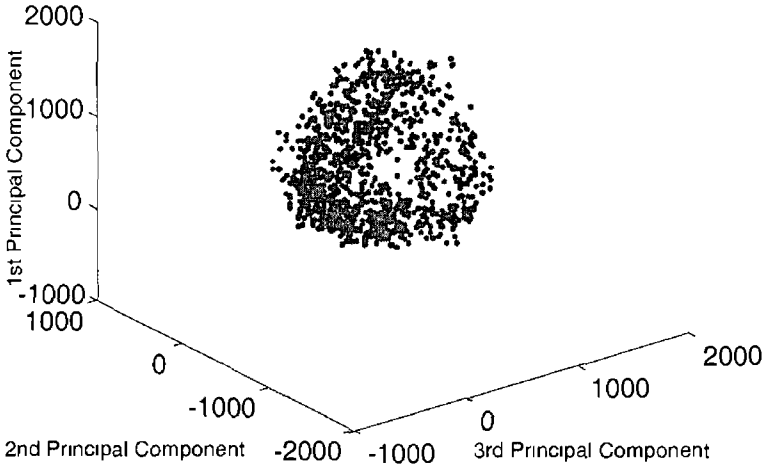


Figure 6 11 1024 extracted codewords for the projection of the images of the training set in the common eigenspace

[1] The number of codewords must be large enough to capture variation in data and small enough so that the trajectories of the gestures are extracted fast The codewords in speech recognition are chosen based on the number of phonemes However since a canonical gesture is just a continuous movement of hand we cannot interpret the codewords as meaningful entities We have chosen one codeword for approximately 100 data points in the training set This number has been selected by trial and error In [Shamaie 2003] it has been shown that by increasing the number of codewords from 1024 to 3200 the recognition rate of the algorithm increases only 1% while the processing time is nearly three times longer By LBG Vector Quantization algorithm we can extract powers of-two number of codewords Therefore, 1024 is the nearest power of two to the 1/100 of the number of data points in the training set As a future work we can find the optimum number of codewords in this problem

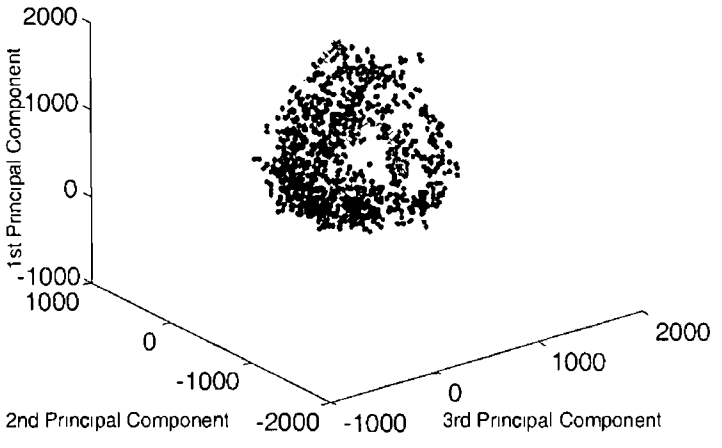


Figure 6 12 The projection of a gesture into the common eigenspace (the trajectory with x points) and the extracted codewords (the dots)

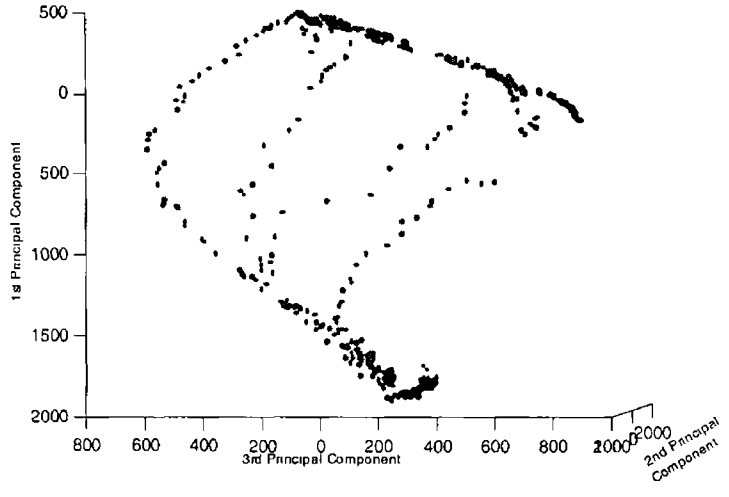


Figure 6 13 Variations in the examples of a gesture appearing in the trajectories in the eigenspace

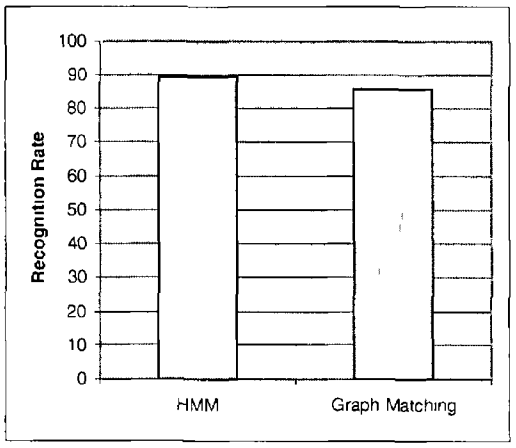
By using the extracted sequence of symbols for the samples of each gesture, a Hidden Markov Model was trained for every gesture Therefore, 100 HMMs were trained for the 100 gestures each with 5 samples of each gesture in the training set

Also, an individual eigenspace was formed by the 5 samples of each gesture in the training set In the 100 constructed eigenspaces the HyperClasses were formed and trained These eigenspaces and the HyperClasses are used in the algorithm with the gaussian Graph-Matching in the second level

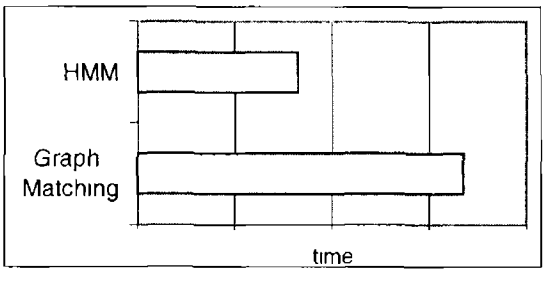
6 4 1 First Experiments

In order to measure the recognition rate of the two algorithms we used the 500 gestures in the test set In these experiments we forwarded all the gestures of the vocabulary to the second level In other words, we bypassed the first level of the algorithm in which the beginning hand shape of a gesture is recognised We are going to estimate the recognition power of the Hidden Markov Models and the gaussian Graph-Matching algorithm in a large database of gestures and compare them together Using the algorithm with HMMs 447 out of 500 gestures of the test set were recognised correctly In other words, **89 4%** recognition rate was obtained for the algorithm with HMMs

Using the algorithm with Graph-Matching, 428 out of 500 gestures of the test set were recognised correctly This means that the Graph-Matching algorithm was able to recognise **85 6%** of the gestures Also the Graph-Matching algorithm takes 2 03 times longer than the HMMs to calculate the best match The graph of the recognition rates and the processing times are shown in Figure 6 14 Obviously the Hidden Markov Models work better than the Graph-Matching in the large vocabularies



(a)



(b)

Figure 6 14 Comparisons of (a) the recognition rate of the algorithms (b) the processing time of the algorithms in recognising a database of 100 canonical gestures

6 4 2 Second Experiments

In this set of experiments we set the $\alpha = 2$ in the first level of the recognition phase Therefore, 2 groups each of which containing 4 gestures are forwarded to the second level of recognition In this case, the algorithms have to recognise the input gesture given 8 forwarded gestures from the first level

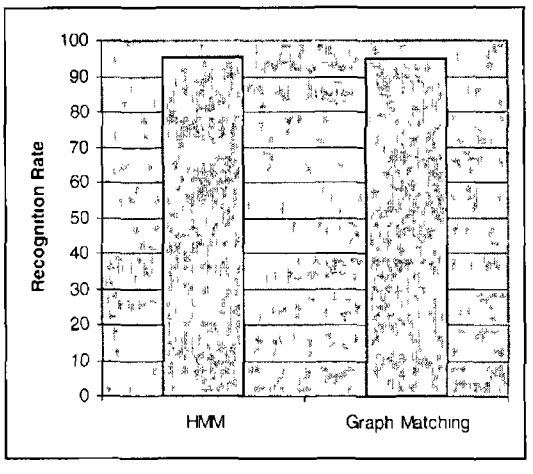
Using the algorithm with HMMs 472 out of 500 gestures in the test set were recognised correctly This is equal to **95 4%** recognition rate By using the gaussian Graph-Matching algorithm, 470 out of 500 gestures in the test set were correctly recognised In other words, the Graph-Matching algorithm was able to recognise **95%** of the gestures

Now we should compare the algorithms from computation time point of view For a gesture in the test set the algorithm with HMMs takes 6 times longer than the Graph-Matching In other words, the Graph-Matching algorithm is 6 times faster than the HMMs which is a great advantage Why is this so?

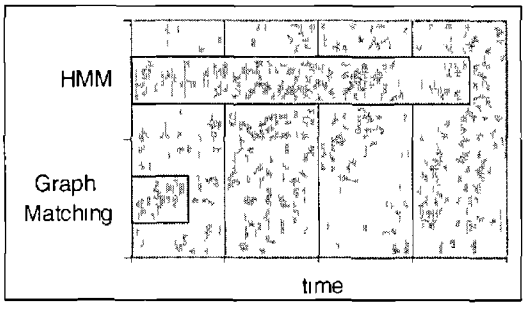
Since in the algorithm with HMMs we have to extract the sequence of codewords for the input gesture, the large space of the codewords constructed for the 100 gestures have to be searched This searching process takes a long time But, calculating the HMM likelihoods is not time consuming However, in the case of the gaussian Graph-Matchings the algorithm has to match the graph of the input gesture with only a small number of HyperClasses (8 in these experimens) Therefore, the matching space is small and the process runs faster The recognition rates and comparison of the processing times are shown in Figure 6 15

As we saw in the first set of experiments the Graph-Matching algorithm had to match 100 graphs each of which was a 32-by-32 bipartite graph This process took twice longer than the HMMs to find the best match Although, the reduction of the number of vertices in the graphs from 32 to some smaller numbers can result in a faster processing, it may affect the recognition rate of the algorithm

Given that the Graph-Matching algorithm has a bit lower recognition rate than the Hidden Markov Models it finds the best match in a fraction of time needed by the Hidden Markov Models At the end, we should mention again that all the experiments in this chapter were based on canonical gestures and they are not including the gestures containing concatenated canonical gestures



(a)



(b)

Figure 6 15 Comparisons of (a) the recognition rate of the algorithms (b) the computation times of the algorithms in recognising a database of 100 canonical gestures using the first level of abstraction in the proposed algorithm

Summary and Conclusion

In this chapter we presented the theory of Hidden Markov Models In order to fast match a gesture within a large database of pre-defined gestures we introduced two hierarchical algorithms, one of them based on the Hidden Markov Models, and the other based on the multidimensional gaussian HyperClasses and Graph-Matching algorithm of Chapter 4

The algorithms, at the first level, use a multidimensional search space of gaussian distributions in order to recognise the beginning shapes of a gesture An abstracted number of gestures are forwarded to the second level where the HMMs and/or gaussian Graph-Matching algorithm are employed Using each of the techniques the algorithms find the best match between the given gesture and the forwarded gestures

In the experiments we showed that the HMMs have 3 8% better recognition rate than the Graph-Matching given the whole database of the gestures

By abstracting the search space in the first level of the algorithm we observed higher recognition rates for both the algorithms In this case the difference between the recognition rates fell to 0 4% while the HMMs remained superior However, from the processing time point of view the Graph-Matching algorithm showed great superiority over the HMMs of 6 times faster processing

While the algorithms compete closely in recognising the gestures correctly, the processing time of each algorithm is an important parameter

However, since there has been much research on Hidden Markov Models there are many models where concatenated canonical gestures are the targets Also the Graph-Matching algorithm has the restriction of number of nodes in the graphs The number of nodes should be large enough to give distinguishing power to the algorithm for a large number of gestures Therefore, it probably does not result in good recognition rate in the case of very short gestures (the gestures recorded in a very short time in only a few images) As a future work we can change the equation of the likelihood so that a larger weight is given to the probabilities than the number of matched vertices In this case it probably works better in recognising short gestures

Given that the HMM has better recognition rate and less restrictions we will use HMM throughout the rest of the thesis

Chapter 7

OCCLUSION DETECTION AND HAND TRACKING IN BIMANUAL MOVEMENTS

Recognition of hand gestures is more realistic when both hands are tracked and any overlapping is taken into account. In bimanual movements the gestures of both hands together make a single gesture. Therefore, one should look at the movement of the hands and track them correctly in order to recognise the whole gesture. An important problem in this type of movements is occlusion. Movement of one hand in front of the other is the main source of occlusion in bimanual movements. In figure 7.1(a), an example of a bimanual movement is shown. Also, for the bimanual movements where there is no occlusion in the essence of the movement, changing the viewpoint can cause one hand to be hidden behind the other occasionally. In Figure 7.1(b) a movement is shown, from the side (see Figure 7.1(c)), one hand is occluded by the other for some moments. Detecting occlusion and tracking the hands are the main problems to be addressed in this chapter.

First a quick review of the hand extraction algorithm will be presented. Then a dynamic model for modelling each hand individually is introduced. Based on this model we predict the movements of the hands separately. By prediction we can forecast the possible occlusion in the movements. We introduce an algorithm for detecting occlusion in a sequence of images. Having the occlusions under control we present a hand tracking algorithm for correct tracking of both hands in bimanual movements. In this algorithm we aim to reacquire the hands after the occlusions. The behaviour of the hands during occlusion is the basis of the process of tracking. Based on a physiological or perceptual[1] phenomenon the hands in bimanual movements tend to be synchronised effortlessly. This synchronisation is the basis of the intelligent tracking algorithm proposed in this chapter. We employ a dynamic model to model the hand movements during occlusion. The model uses the synchronisation of the hands in order to recognise the hands' behaviour. This behaviour forms the basis of the tracking algorithm.

---

[1] Two explanations have been proposed for this phenomenon. A group of scientists believe that the bimanual synchronisation is a very powerful constraint in motor control. Another group, however, has demonstrated that this synchronisation has a perceptual basis. We present more detail about this synchronisation in the next sections.

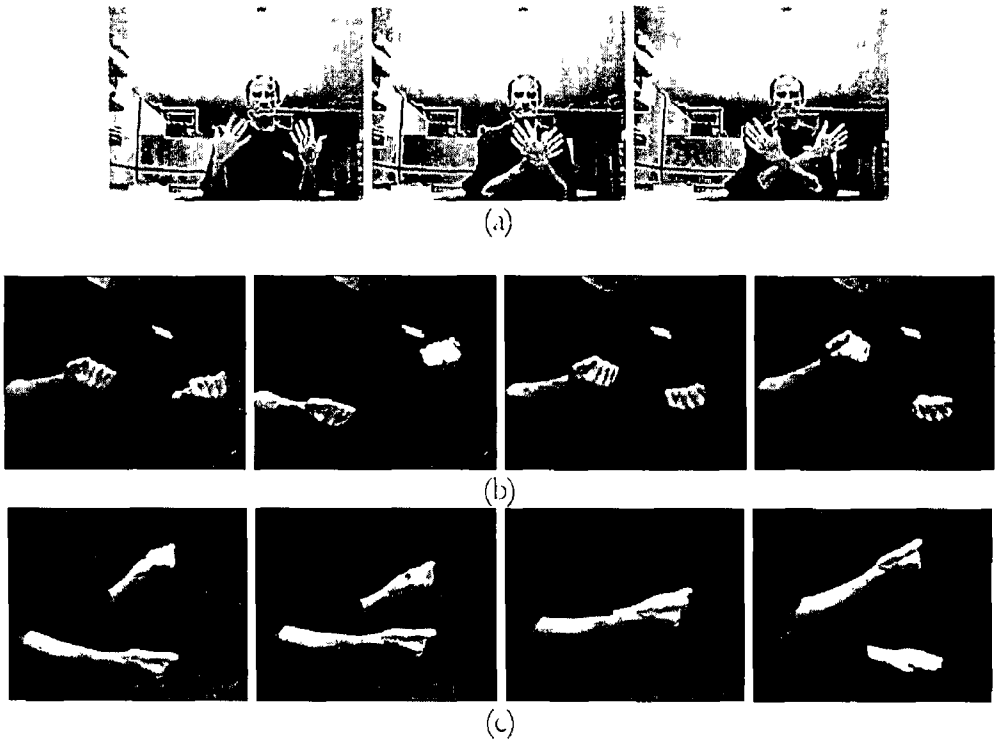


Figure 7 1 (a) A general bimanual movement (b) a bimanual movement from frontal view (c) from side view

7 1 Hand Extraction and Related Work

By using pixel grey-level detection we extract the hands from background In an extracted image only the pixels with a non-zero value could belong to the hands We use the Grassfire algorithm [Pitas 1993] in order to extract the hands Calling from Chapter 3, Grassfire is a *region labelling* or *blob analysis* algorithm It finds all the connected regions and labels them This algorithm scans an image from left to right, top to bottom to find the pixels of connected regions with values belonging to the range of hand colour (in grey scale) For the first pixel found in that range it turns around the pixel to find other pixels

By considering a square around a pixel (see Figure 7 2) the algorithm scans the square

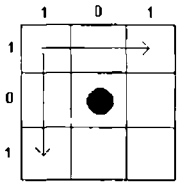


Figure 7 2 The square around a pixel

Therefore by finding all the pixels belonging to a connected region the whole region is extracted A blank matrix is constructed with the same size as the image We call it the shadow matrix For every extracted pixel a numeric label is placed at the same position in the shadow matrix All pixels in the same connected region get the same label When a connected region is totally extracted and labelled in the shadow matrix the algorithm searches for other connected regions Also, for every connected region its area is measured The process repeats until all the connected regions in an image are extracted and labelled At the end of the algorithm the shadow matrix stores the labels corresponding to all the connected regions in the image (see Figure 7 3) Due to noise there might be some other spots appearing in an image with the same grey range as the hands Therefore, we should separate the hands from the noisy spots We look at the size of the objects The objects with very small areas are treated as noisy spots and ignored

Now, we have extracted the hands and labelled them separately But, in the images where the hands are in contact the algorithm extracts only one connected region and is not able to separate the hands due to occlusion Therefore, we cannot recognise the hands correctly in the presence of hand-hand overlapping

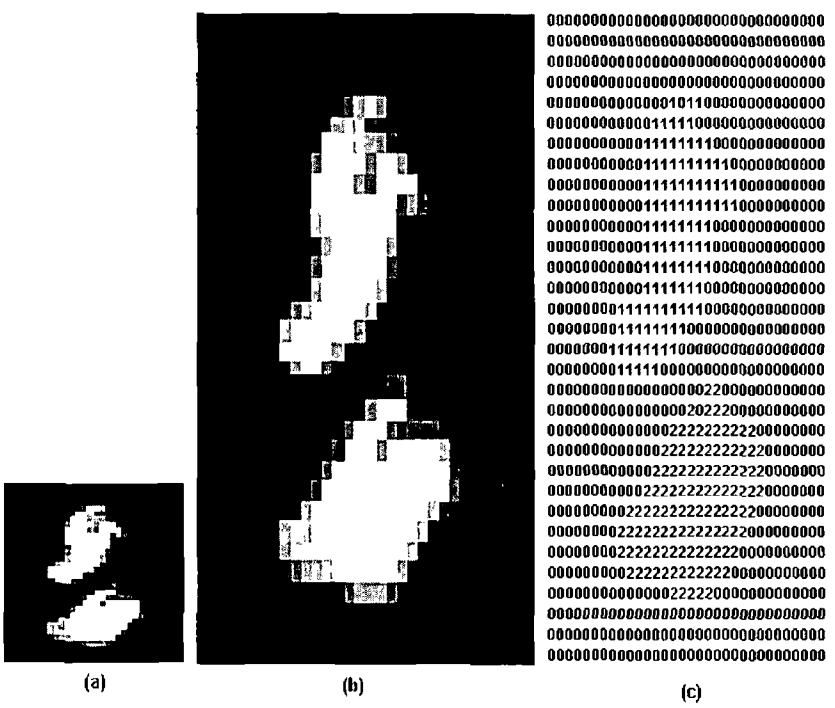


Figure 7 3 (a) An image containing the two hands, (b) the enlarged image (c) the shadow matrix with the labelled hand shapes

There are a number of techniques proposed in the literature for tracking two overlapping objects CONDENSATION is the natural extension of Kalman filter to factored sampling [Isard 1998a] Mammen et al [Mammen 2001] have employed CONDENSATION algorithm to track the hands off-line in video sequences captured at 12 frames-per-second No performance has been reported regarding the processing speed and the efficiency of the algorithm in tracking a set of hand motions

Although, by tracking the edges of two objects the CONDENSATION algorithm tracks the overlapping of objects robustly, it is a very time consuming process (see Section 2 3 2) which cannot be used in a real-time system [Sherrah 2000] On the other hand, we do not really need that much degree of accuracy in tracking the edges of the two hands during occlusion because the separation of the hands during occlusion does not provide us with significant useful information about the occluded hand

The other technique for tracking is Point Distribution Model (PDM) [Cootes 1992], which was explained in Chapter 2 But this technique has also some disadvantages that prevent using it as an efficient technique for tracking The model breaks down for complex objects, and more importantly is that it needs quadratic optimisation to automaticly identify a set of landmark points, which is not an efficient way to work in real-time applications The analytical models such as [McAllister 2002 Davis 1999] also need to do non-linear optimisations

Gong et al [Gong 2002] have used a Bayesian network to track two interacting hands Their proposed algorithm can process 5 frames per second on a Pentium II 330 MHz computer

As it was mentioned in Chapter 5, a fast and efficient technique for tracking is the Kalman filter [Brown 1997, Chui 1999] Dockstader et al [Dockstader 2000] have used the Kalman filter to track human head and body in the presence of occlusion Zieren et al [Zieren 2002] have used Kalman filter to track the two hands but not in the presence of hand-hand occlusion

Based on the advantages of Kalman filter we use this technique and a dynamic model to track the hands in the presence of occlusion The first step is occlusion detection in a sequence of images

7 2 Occlusion Detection in Bimanual Movements

Two types of occlusion are considered here First, the case where one hand occludes the other We call it hand-hand occlusion Second, the case in which something else occludes a hand or the hand hides behind something, e g the body, partially or completely When one hand occludes the other we must detect the exact beginning point of occlusion By this we are able to separate the hand-hand occlusion from the other type of occlusion For this we introduce the following model

As before, a rectangle is constructed around each hand in an image Therefore, by moving a hand its rectangle moves in the same way By tracking these rectangles we detect the start and end points of occlusion To detect the beginning point we look at the movement of the rectangles If at some stage there is any intersection between the rectangles it can be recognised as occlusion However, in some cases there might be an intersection with no occlusion (see Figure 7 4)

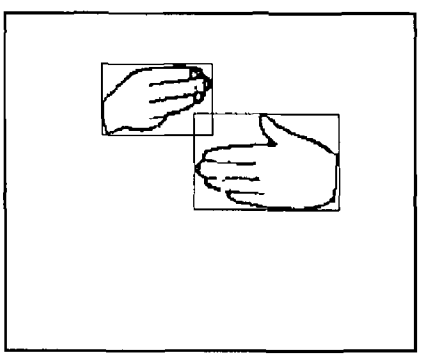


Figure 7 4 An intersection of the rectangles with no occlusion

Also, if we suppose that at time *t* there is no intersection of the rectangles and at time *t+1* occlusion happens, there is only one big blob and one rectangle is constructed around it (see Figure 7 5) It happens because the hand shapes are connected together and the Grassfire algorithm extracts the connected region of the hands as a single object Occlusion is not detectable because this is similar to a hand's movement out of camera frame or hiding behind a part of body

To overcome this problem, we use a model to predict the future movement of each hand

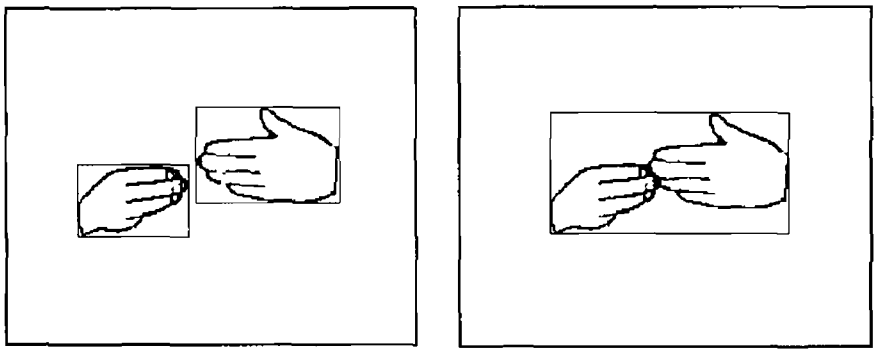


Figure 7 5 Occlusion happens in two consecutive frames

We propose a dynamic model based on Kinematic equations of motion and Kalman filtering to track the movements and predict the future position of the rectangles By this, we can predict possible intersection of the rectangles a few steps in advance This gives us an alarm of any probable occlusion

Every rectangle is modelled by the following equation,

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{w}_k \tag{7 1}$$

where $\mathbf{x}_k$ is the state vector representing the rectangle at time k, $\mathbf{\Phi}$ is the matrix relating the two consequent positions of a rectangle, and $\mathbf{w}_k$ is zero-mean Gaussian white system noise [Brown 1997] The movement of a rectangle can be modelled by the movement of its sides (see Figure 7 6) Therefore, Equation 7 1 is expanded to,

$$\begin{bmatrix} x^i_{1\,k+1} \\ x^i_{2\,k+1} \\ y^i_{1\,k+1} \\ y^i_{2\,k+1} \end{bmatrix} = \mathbf{\Phi} \begin{bmatrix} x^i_{1\,k} \\ x^i_{2\,k} \\ y^i_{1\,k} \\ y^i_{2\,k} \end{bmatrix} + \mathbf{w}^i_k, \quad i = 1,2 \tag{7 2}$$

where $x^i_{1\,k}$, $x^i_{2\,k}$, $y^i_{1\,k}$ and $y^i_{2\,k}$ are the sides of the rectangle i at time k

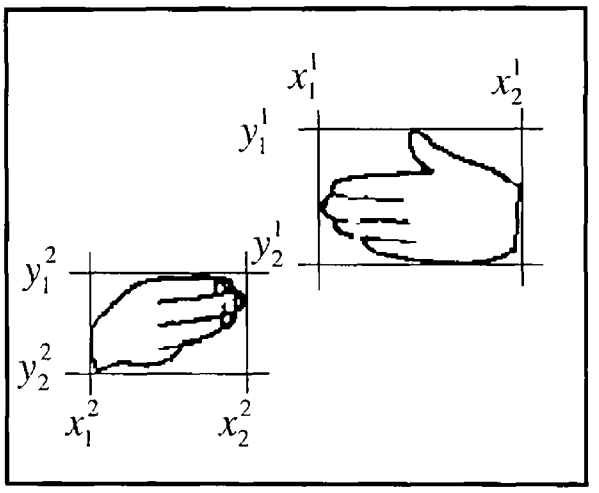


Figure 7 6 Every rectangle is modelled by its sides In this figure $[x_1^1 \quad x_2^1 \quad y_1^1 \quad y_2^1]^T$ is the rectangle of the first hand and $[x_1^2 \quad x_2^2 \quad y_1^2 \quad y_2^2]^T$ is the rectangle of the second hand

For every parameter in this model we have the position, velocity and acceleration Therefore, using the dynamic model of Chapter 5 our model is expanded to the Equation 7 3 for i=1, 2

$$\begin{bmatrix} x^i_{1\,k+1} \\ x^i_{1\,k+1} \\ x^i_{1\,k+1} \\ x^i_{2\,k+1} \\ x^i_{2\,k+1} \\ x^i_{2\,k+1} \\ y^i_{1\,k+1} \\ y^i_{1\,k+1} \\ y^i_{1\,k+1} \\ y^i_{2\,k+1} \\ y^i_{2\,k+1} \\ y^i_{2\,k+1} \end{bmatrix} = \begin{bmatrix} 1 & h & \frac{h^2}{2} & & & & & & & & & \\ 0 & 1 & h & & & & & & & & & \\ 0 & 0 & 1 & & & & & & & & & \\ & & & 1 & h & \frac{h^2}{2} & & & & & & \\ & & & 0 & 1 & h & & & & & & \\ & & & 0 & 0 & 1 & & & & & & \\ & & & & & & 1 & h & \frac{h^2}{2} & & & \\ & & & & & & 0 & 1 & h & & & \\ & & & & & & 0 & 0 & 1 & & & \\ & & & & & & & & & 1 & h & \frac{h^2}{2} \\ & & & & & & & & & 0 & 1 & h \\ & & & & & & & & & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^i_{1\,k} \\ x^i_{1\,k} \\ x^i_{1\,k} \\ x^i_{2\,k} \\ x^i_{2\,k} \\ x^i_{2\,k} \\ y^i_{1\,k} \\ y^i_{1\,k} \\ y^i_{1\,k} \\ y^i_{2\,k} \\ y^i_{2\,k} \\ y^i_{2\,k} \end{bmatrix} + \mathbf{w}^i_k \qquad (7\,3)$$

where x_1^i, x_2^i, y_1^i, y_2^i are assumed to have continuous first and second order derivatives denoted by one-dot and double-dot variables, and $h>0$ is the sampling time [Chui 1999] The position, velocity and the acceleration of every side of a rectangle are related based on the following equation,

$$\begin{cases} x^i_{j\,k+1} = x^i_{j\,k} + h x^i_{j\,k} + \frac{1}{2} h^2 x^i_{j\,k} \\ x^i_{j\,k+1} = x^i_{j\,k} + h x^i_{j\,k} \end{cases} \qquad i = 1,2 \quad j = 1,2 \tag{7 4}$$

where x is the position, x the velocity and x the acceleration of a side As in Chapter 5, only the position of a rectangle is observable Therefore, we define the matrix $\mathbf{H}$ as following,

$$\mathbf{H} = [1 \quad 0 \quad 0] \tag{7 5}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{H}$ gives the noiseless connection between the measured vector $\mathbf{z}^i_k$ and the state vector $\mathbf{x}^i_k$ in,

$$\mathbf{z}^i_k = \mathbf{H}\mathbf{x}^i_k + \mathbf{v}^i_k, \qquad i = 1,2 \tag{7 6}$$

where

$$\mathbf{x}^i_k = \begin{bmatrix} x^i_{1\,k} \\ x^i_{1\,k} \\ x^i_{1\,k} \\ x^i_{2\,k} \\ x^i_{2\,k} \\ x^i_{2\,k} \\ y^i_{1\,k} \\ y^i_{1\,k} \\ y^i_{1\,k} \\ y^i_{2\,k} \\ y^i_{2\,k} \\ y^i_{2\,k} \end{bmatrix}$$

and $\mathbf{v}_k$ is the zero-mean Gaussian white measurement noise Then the Kalman filtering model takes on the following stochastic description [Chui 1999] for i=1, 2,

$$\begin{cases} \mathbf{x}^i_{k+1} = \mathbf{\Phi}\mathbf{x}^i_k + \mathbf{w}^i_k \\ \mathbf{z}^i_k = \mathbf{H}\mathbf{x}^i_k + \mathbf{v}^i_k \end{cases} \tag{7 7}$$

As in Chapter 5, we can decompose the model into 4 submodels, each of which is presented by,

$$\begin{cases} \begin{bmatrix} x_{k+1}(1) \\ x_{k+1}(2) \\ x_{k+1}(3) \end{bmatrix} = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k(1) \\ x_k(2) \\ x_k(3) \end{bmatrix} + \mathbf{w}_k \\ z_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k(1) \\ x_k(2) \\ x_k(3) \end{bmatrix} + \mathbf{v}_k \end{cases} \tag{7 8}$$

where $x_k(1) = x_k$, $x_k(2) = x_k$ and $x_k(3) = x_k$ The Kalman filtering equations for this model are the same as equations stated in Chapter 5 and Appendix C

In this model the prediction of the future is performed by projecting the current state ahead, Equation 7 9

$$\hat{\mathbf{x}}'_{k+1} = \mathbf{\Phi}\mathbf{x}'_k \tag{7 9}$$

This equation predicts the next state of the vector $\mathbf{x}$ one step in advance In other words, it predict the position of the rectangle t one step in advance

We set an *occlusion alarm* if the algorithm predicts an intersection between the rectangles on the next step (see Figure 7 6) Having the occlusion alarm set, as soon as the hand shapes join together we detect the occlusion Therefore, we are able to capture the hand-hand occlusion and differentiate it from the other type of occlusion

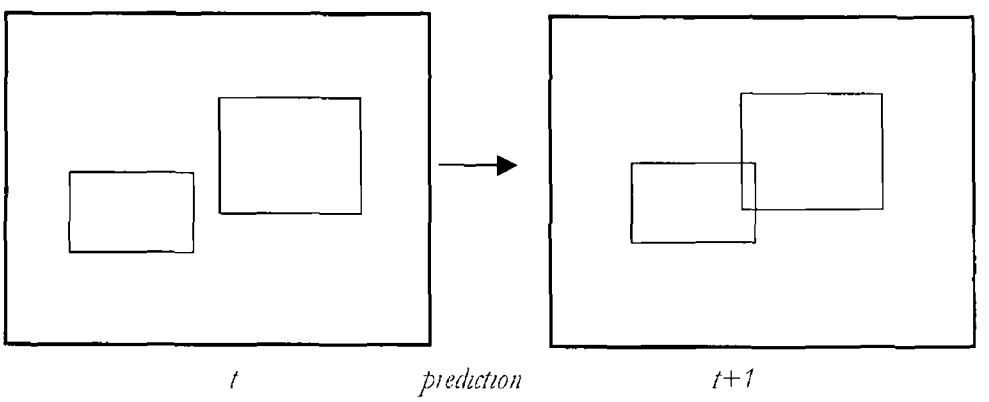


Figure 7 7 Prediction of intersection of the rectangles by Kalman filter and the dynamic model

The occlusion detection algorithm is summarised as following,

1 *By using Grassfire the hands are extracted and the rectangles are constructed*
2 *The dynamic model is applied to each rectangle and the future positions are predicted*
3 *If the predicted rectangles have any intersection the occlusion alarm is set*
4 *In the next captured image if only one hand is detected by Grassfire and the occlusion alarm is already set the hand hand occlusion has happened Otherwise if we see one hand in the image and the occlusion alarm is not set, the other type of occlusion (e g occlusion by a part of body or leaving the scene) is detected*
5 *Image capturing is continued*
6 *In any step that two hands are detected in an image while the hand hand occlusion variable is set the end of occlusion is detected*

By this algorithm we are able to detect the beginning and end of occlusions very accurately

7 3 Hand Tracking in Bimanual Movements

A problem with the hand extraction algorithm (Grassfire) is that the first shape found in an image is labelled as the first hand This causes difficulties in two forms,

1 The hands move so that in two consecutive images the hand shapes are labelled interchangeably (see Figure 7 8) This happens because of the search manner of the Grassfire algorithm

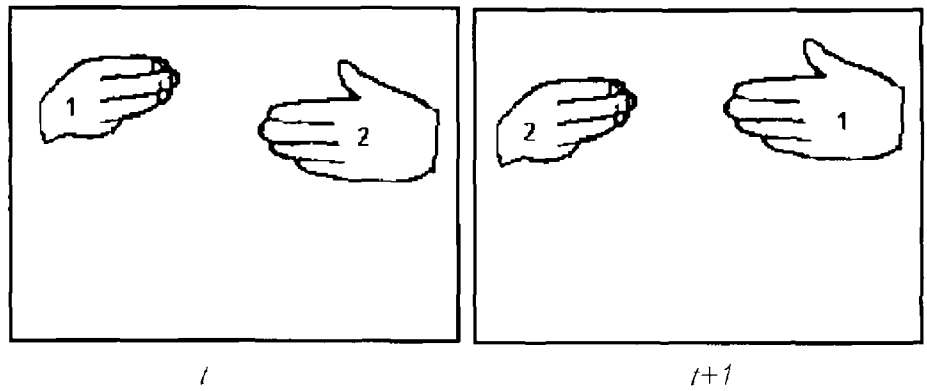


Figure 7 8 The hands in two consecutive images may be labelled interchangeably by the hand extraction algorithm

Chen et al [Chen 2003] track the centroids of colour finger gloves to track the fingers in a surgical operation We use the centroid of the hands to track them in a sequence of images By finding the centroids of the handa and comparing them in two consecutive frames the problem of mislabelling in the consecutive frames can be recovered (see Figure 7 9) The centroids of the hands are the centres of the tracked rectangles in the last time frame and the hand centres in the current observation The movement of the centroids in the consecutive images is smooth that enables us to track the correct position of hands even if the Grassfire algorithm labels them interchangeably

By using this technique we are able to track the hands correctly even when something else occludes them For example, if one of the hands is occluded or totally hidden by the body for some moments and then appears, it can be tracked correctly by keeping records of its last position before occlusion and the position of the other hand This is expected because when a hand moves behind something like the body or moves out of the image frame it most probably appears in an area close to the last position before the occlusion Therefore, if at some points there is only one hand in the image the algorithm keeps tracking the hands properly without any confusion

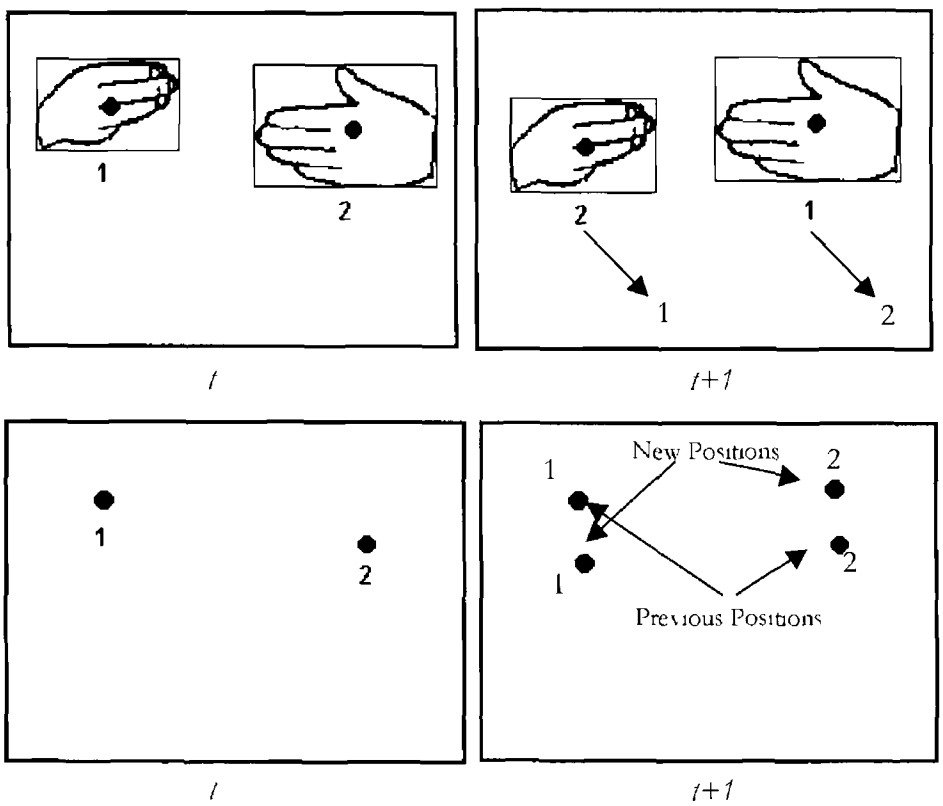


Figure 7 9 By comparing the hand centroids we can track the hands correctly in the consecutive images

2 In a bimanual movement, when one hand, completely or partially, covers the other hand the hand extraction algorithm detects one big blob in the images In this case tracking and resuming the hands accurately at the end of occlusion is crucial

Since we don't know what exactly happens during occlusion, after the end of occlusion, we have to know which hand in the image is the right hand and which hand is the left This is the important and difficult problem of tracking in the presence of occlusion We introduce another algorithm for this problem In order to track the hands we classify the bimanual

movements based on the path of each hand's movement The movements are classifies as follows,

Class 1 The hands move toward each other, one occludes the other for some moments and passes over it Models of *a c d*, and *h* presented in Figure 7 10 (a), (c), (d), and (h)

Class 2 The hands move toward each other, they collide and return in the opposite directions Models of *b, g, k* and *l* shown in Figure 7 10 (b), (g), (k), and (l)

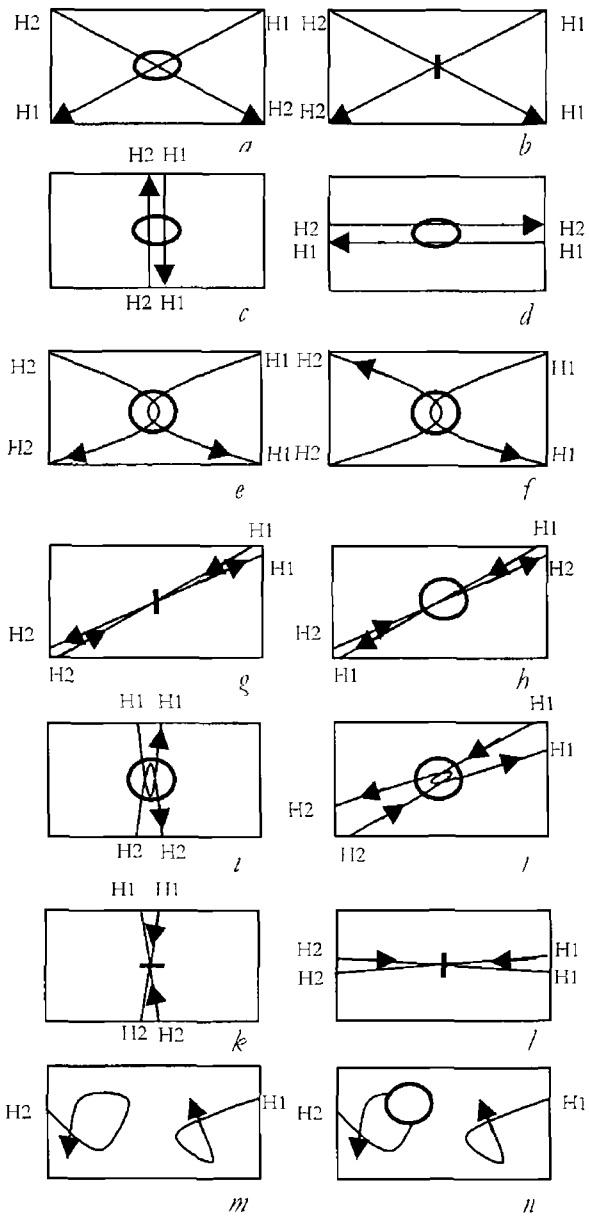


Figure 7 10 The path of the hands in the 14 models of bimanual movements H1 and H2 represent hand number one and hand number two The thick ellipses represent the occlusion areas (*a c d e f h i, j* and *n*) and the solid small rectangles represents collision (*b, g, k,* and *l*)

Class 3 The hands move, at some point one occludes the other with no collision and they return to their previous sides Movements of model *e, f, i* and *j* shown in Figure 7 10 (e), (f), (i), and (j)

Class 4 The hands move with no hand-hand occlusion Occasionally one of the hands may be occluded by something else either partially or completely Movements of model *m* and *n* shown in Figure 7 10 (m) and (n)

In the first class the hands continue their smooth movements without any collision In the second class they collide and change their path In the third class they do not collide but change their path And in the fourth class there is no hand-hand occlusion A tracking system has to be able to recognise these classes and track the hands correctly at the end of occlusion

For example, clapping can be represented by model *g*, tying a knot by model *j*, etc We aim to reacquire the hands at the end of occlusion periods Therefore, we find the class that a movement belongs to in order to understand the behaviour of the hands during a hand-hand occlusion period

We approach the problem from a neuroscience point of view, because in this way we can understand the behaviour of the hands during occlusion periods First, we review a motor control phenomenon called *Bimanual Coordination* and our motivation for using it as the basis of our tracking algorithm Based on this phenomenon we introduce a tracking algorithm to capture bimanual coordination and intelligently track and reacquire the hands in bimanual movements

7 3 1 Bimanual Coordination

Neuroscience studies show that in bimanual movements the hands tend to be synchronised effortlessly [Jackson 2000] This synchronisation appears in both temporal and spatial forms [Diedrichsen 2001] Temporally, when the two hands reach for different goals they start and end their movements simultaneously [Diedrichsen 2001] For example, when people tap with both hands, the taps are highly synchronised Spatially, we are almost not able to draw a circle with one hand while simultaneously drawing a rectangle with the other [Diedrichsen 2001] "Synchronisation of the two hands is in a mirror-like fashion [Mechsner 2002]" This synchronisation appears in two forms symmetrical and parallel (see Figure 7 11)

Researchers have presented different explanations for this phenomenon [Donchin 1998] Some scientists explain that the tendency to move in symmetry is closely related to the symmetrical structure of the body and the nervous system This can be explained by a tendency to co-activate anatomically homologous muscle groups Homologous muscles, as well as bilaterally situated areas in the two brain hemispheres and in the spinal cord can be activated together This is because of their interconnection through neuronal pathways [Mechsner 2002]

However, Mechsner et al [Mechsner 2001] argue that the symmetry in bimanual movements has a perceptual basis They suggest that spontaneous coordination phenomena of this kind are purely perceptual in nature

Figure 7 11 (a) Symmetrical movement of fingers (b) Parallel movement of fingers (Reprinted with permission Nature Publishing Group and F Mechsner ref[Mechsner 2001])

Temporal coordination implies that the hands velocities are synchronised in bimanual movements Also the hands pauses happen simultaneously We exploit hands temporal coordination to track the hands in the presence of occlusion

In order to detect the pauses we monitor the hand velocities A well-known experiment shows that the two hand velocities are highly synchronised in bimanual movements [Kennerley 2002] 'Circle drawing is the task of drawing circles by the two hands simultaneously in a symmetrical fashion [Kennerley 2002]" Figure 7 12 shows the result of this experiment on a healthy person [Kennerley 2002]

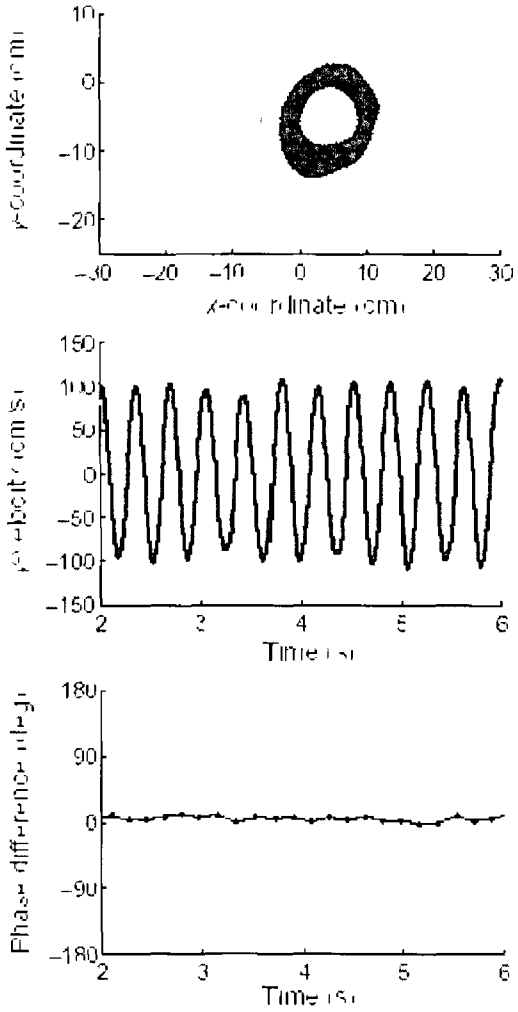


Figure 7 12 Circle drawing experiment Position (top) velocity along the y axis (middle) and relative phase relationship between the two hands (bottom) (Reprinted with permission Nature Neuroscience and R Ivry ref [Kennerley 2002])

In this experiment it is shown that the y-velocities of the two hands are highly synchronised, and no phase difference is observed Velocity synchronisation and concurrent-pauses detection in bimanual movements are the bases of an intelligent tracking algorithm presented in the next section

7 3 2 Tracking Algorithm

We introduce a technique based on the dynamic model of Section 7 2 As in that section a rectangle is constructed around each hand As soon as the occlusion is detected by the occlusion-detection algorithm of Section 7 2 a occlusion-rectangle around the big blob is formed (see Figure 7 13) We call it the *occlusion rectangle*

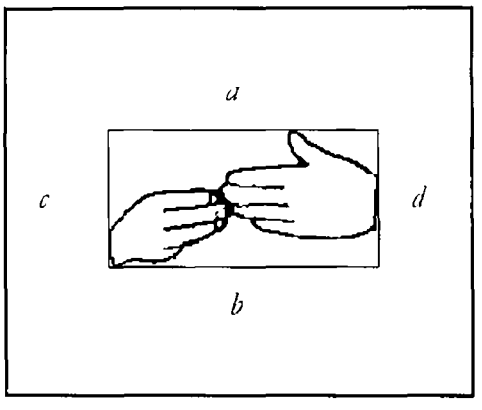


Figure 7 13 An occlusion rectangle is formed around the big blob of hands

We use the dynamic model to model the occlusion-rectangle Therefore, for every side of the rectangle the position, x, velocity, $\dot{x}$, and acceleration, $\ddot{x}$, are involved in the model The horizontal movement of the hands are modelled by the vertical sides, c and d in Figure 7 13, and the vertical movement by the horizontal sides, a and b For simplicity we define the following auxiliary variables,

$v_a = \dot{x}_a$ velocity of side a

$v_b = \dot{x}_b$ velocity of side b

$v_c = \dot{x}_c$ velocity of side c

$v_d = \dot{x}_d$ velocity of side d

Then the following *hand pause model* is defined to model the velocities of the hands in the vertical and horizontal directions,

$$\begin{cases} v_{v,k} = \sqrt{v_{a,k}^2 + v_{b,k}^2} \\ v_{h,k} = \sqrt{v_{c,k}^2 + v_{d,k}^2} \end{cases} \tag{7 10}$$

where the subscript k indicates the discrete time index

In the movements where the hands either collide or pause (Classes 2 and 3) they return to the same sides prior to the occlusion period In these movements the parallel sides of the rectangle in either horizontal or vertical directions pause when the hands pause or collide For example, in the models of c, f and l the hands horizontally pause and return to their previous sides In the models g and j they pause and return in both horizontal and vertical directions The horizontal pauses of the hands are captured by the pauses of the vertical sides

of the occlusion-rectangle and vice versa Due to bimanual coordination the pauses of the parallel sides are simultaneous In other words, when the hands pause either horizontally or vertically the parallel sides associated with the horizontal and vertical movements of hands pause simultaneously For example, in the models i and k the horizontal sides of the occlusion-rectangle pause simultaneously when the hands pause or collide vertically during occlusion In this case the velocities of the horizontal sides of the occlusion-rectangle reach zero This is captured by $v_{v,k}$ in the hand-pause model In fact, a small threshold $\varepsilon > 0$ can provide a safe margin because we are working in discrete time and our images are captured at discrete points in time If $v_{v,k}$ or $v_{h,k}$ falls below the threshold we conclude that the hands have paused vertically or horizontally By detecting the pauses in the horizontal or vertical direction we conclude that the hands have paused or collided and returned to the same sides prior to occlusion in that direction

In the movements where the hands pass each other, no pause or collision is detected but a change in the sign of the velocities is observable The sign change is due to the fact that when the hands pass each other they push the sides in opposite directions (see Figure 7 14) Therefore, the sign of the velocities are changed without passing through zero If no hand pause is detected we conclude that the hands have passed each other

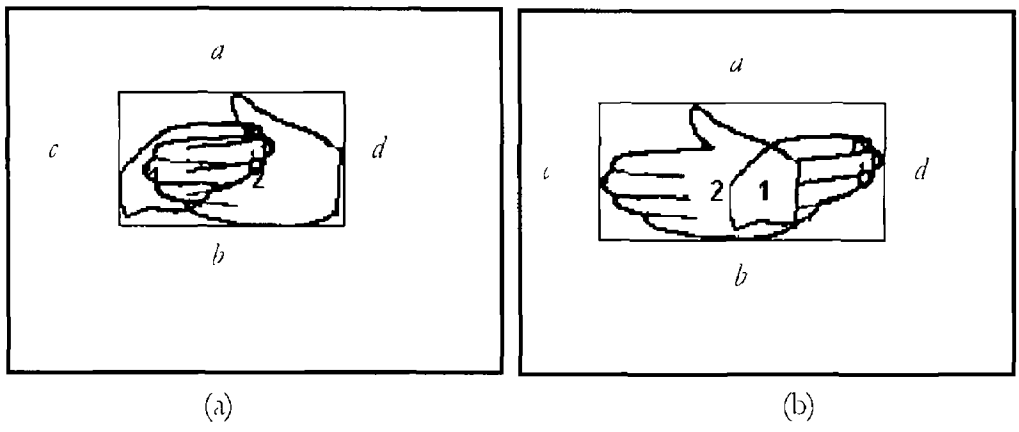


Figure 7 14 The vertical sides of the occlusion-rectangle are pushed back because hands pass each other and push the vertical sides in opposite direction

In a typical movement the hand shapes may change during the occlusion period For example, in a movement where the hands move, the fingers may also move concurrently so that the shape of hand is changed In this case the movement of fingers may prevent the detection of the simultaneous pauses of the hands This is also true when a pair of parallel

sides of the occlusion-rectangle are both connected to one hand since the other hand is in a small shape In this case, for example, one of the vertical sides is connected to the palm while the other is connected to the fingers Therefore, the change in the hand shape may prevent simultaneous pauses of both the vertical sides

To investigate this problem we did an experiment presented in Appendix F The result shows that the fingers and the hand are coordinated in the movement of one hand In other words, the hand and fingers are temporally synchronised Our experiment shows that the velocity of the hand and the velocity of the fingers are highly synchronised with almost no phase difference Therefore, the pauses of the hand and the pauses of the fingers that change the hand shape happen simultaneously This is due to the fact that in motor control the temporal coupling not only between the limbs but also within a limb is a very powerful constraint [Ivry 2003] Therefore, the hand-finger coordination guarantees that the velocities of the parallel sides of the rectangle are synchronised and the pauses happen simultaneously This phenomenon makes the algorithm independent of the changing hand shape

In some of the models where the hands have purely horizontal (models *d* and *l*) or vertical (models *c*, *i*, and *k*) movements, an unwanted pause may be detected in the vertical or horizontal directions For example, when the hands move only horizontally (see Figure 7 10(d)) a vertical pause may be detected because vertically they have not much movement and the speed of the vertical sides may reach zero Also, in the models where a pair of parallel sides of the occlusion-rectangle move in the same direction[2] (e g horizontal sides in models *a*, *b*, and *e*), while no zero velocity (pause) is detected, we may wrongly conclude that the hands have passed each other in that direction (vertical direction in models *a*, *b*, and *e*) These problems can cause the tracking algorithm to run into trouble

In order to solve these problems we classify the velocity synchronisation of the hands mentioned in Section 7 3 1 into two classes, positive and negative In the movements where the two hands move in opposite directions (e g left and right) the velocities are negatively synchronised, while in the movements where they move in the same direction (e g down) the velocities are positively synchronised

[2] Here by direction we mean up down left or right

To distinguish the positive and negative synchronisations we define the following *velocity synchronisation model*, which is the standard deviation of the relative velocities of the parallel sides,

$$\begin{cases} s_v^2 = \dfrac{1}{N-1}\sum_i \left[(v_{a\,i} - v_{b\,i}) - \dfrac{1}{N}\sum_j (v_{a\,j} - v_{b\,j}) \right]^2 \\ s_h^2 = \dfrac{1}{N-1}\sum_i \left[(v_{c\,i} - v_{d\,i}) - \dfrac{1}{N}\sum_j (v_{c\,j} - v_{d\,j}) \right]^2 \end{cases} \qquad (7\ 11)$$

where N is the number of images (frames) during the occlusion period, i and j are the frame indices, $v_{a\,k}$, $v_{b\,k}$, $v_{c\,k}$, and $v_{d\,k}$ are the velocities of sides a, b, c, and d at the k^{th} frame during occlusion

This model results in small standard deviations in the purely horizontal or purely vertical movements as well as the movements where the parallel sides are positively synchronised For example, in a movement of model *c*, the vertical sides of the occlusion-rectangle have almost no movement during the occlusion period Therefore, s_h in the velocity-synchronisation model (System 7 11) will be small In model *e*, the horizontal sides of the occlusion-rectangle are positively synchronised s_v in this case becomes small However, if the velocities of the parallel sides of the occlusion-rectangle are negatively synchronised (e g model *f*) the standard deviations are large Because in this case the velocities of parallel sides are in opposite directions with different signs The thresholds for the small s_h and s_v are determined by experiment

Before we detect the hand pauses we capture any possible positive synchronisation of parallel sides of the occlusion-rectangle during the occlusion period using the velocity-synchronisation model If a positive synchronisation for any pair of parallel sides is observed, the tracking is performed based on the pauses of the other sides of the occlusion-rectangle For example, if a small s_v is observed we base the tracking on the pauses of the other sides, *c* and *d* A small standard deviation in the velocity-synchronisation model means that a pair of parallel sides of the rectangle have been positively synchronised with quite similar velocities during occlusion Therefore, we should look at the pauses of the other sides of the occlusion-rectangle during occlusion

Based on the velocity-synchronisation and hand-pause models the hand tracking algorithm is summarised as following,

1 *If the horizontal sides of the rectangle are positively synchronised (small* s_v *) during the occlusion period*

1 A *If during occlusion there is a k such that* $v_{h,k} < \varepsilon$ *then the hands are horizontally back to their original position*

1 B *Else the hands horizontally passed each other*

2 *Else if the vertical sides of the rectangle are positively synchronised (small* s_h *) during the occlusion period*

2.A *If during occlusion there is a k such that* $v_{v,k} < \varepsilon$ *then the hands are vertically back to their original position*

2 B *Else the hands vertically passed each other*

3 *Else if during occlusion there is a k such that* $v_{h,k} < \varepsilon$ *then the hands are horizontally back to their original position*

4 *Else if during occlusion there is a k such that* $v_{v,k} < \varepsilon$ *then the hands are vertically back to their original position*

5 *Else the hands passed each other*

The above algorithm tracks the hands smartly during occlusion and makes a decision on the position of the hands at the end of occlusion

7 4 Experimental Results

Before we start the experiments some information should be given about the distance of camera and the hands, and the filming area

In our experiments, the camera is placed about 250cm from the subject filming a 70-by-70cm area approximately The resolution of the camera is set to 400-by-200 pixels Therefore, each pixel on the vertical axis corresponds to 3 5mm and on horizontal axis to 1 75mm approximately on the filming area Also based on our experiments, that will be presented later, an image takes 26 6ms to be processed by the algorithm In other words, the time unit between two consecutive images is 26 6 milliseconds

$$u = 26\ 6\ ms$$

Therefore, a velocity of 1 pixel per time-unit is equal to 131 25mm per second or 13 125 cm/s on the vertical axis For example, a velocity of 10 pixels per time-unit is equal to 1 3 meter per second approximately on the vertical axis Also, a velocity of 1 pixel per time-unit on the horizontal axis is equal to 65 626 millimetre per second All the units in the following examples are image pixels and the time unit, u

First, we look at the performance of the dynamic model in predicting the future position of the rectangles The results of employing the dynamic model in tracking and predicting a vertical side of a rectangle is shown in Figure 7 15 In this figure, the solid dots connected by lines are the actual position of the side of a rectangle during a hand movement The small circles show the result of prediction by the dynamic model A closer look at this graph gives a better view of the effectiveness of the algorithm in predicting the next position of a side of the rectangle In Figure 7 16 the part of the graph of Figure 7 15 from the 115th to 162nd time unit is magnified It is clearly visible that the dynamic model is able to predict the future position of the side of the rectangle accurately The graphs of the other sides of the rectangle in this experiment are shown in Figure 7 17

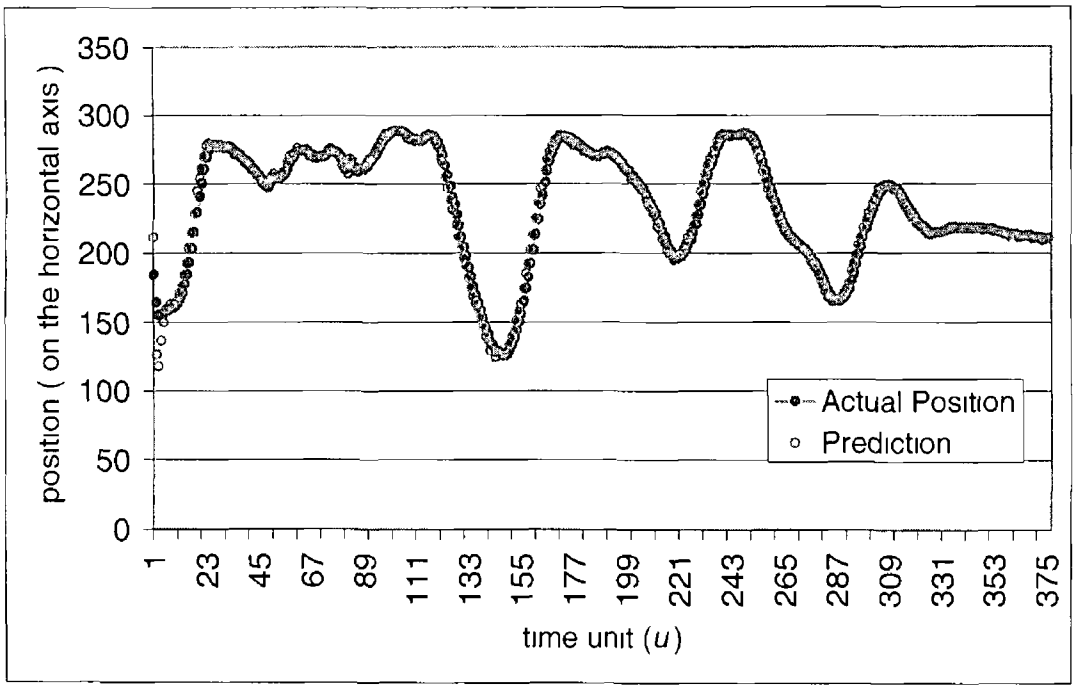


Figure 7 15 The position of a side of a rectangle during a hand movement and the predictions

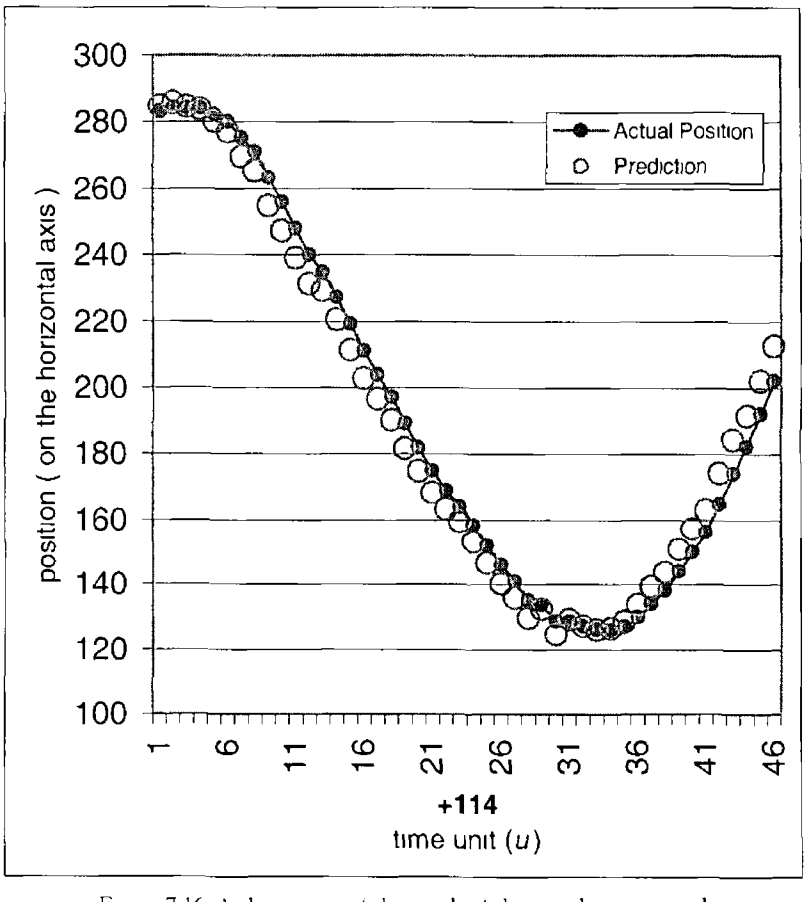
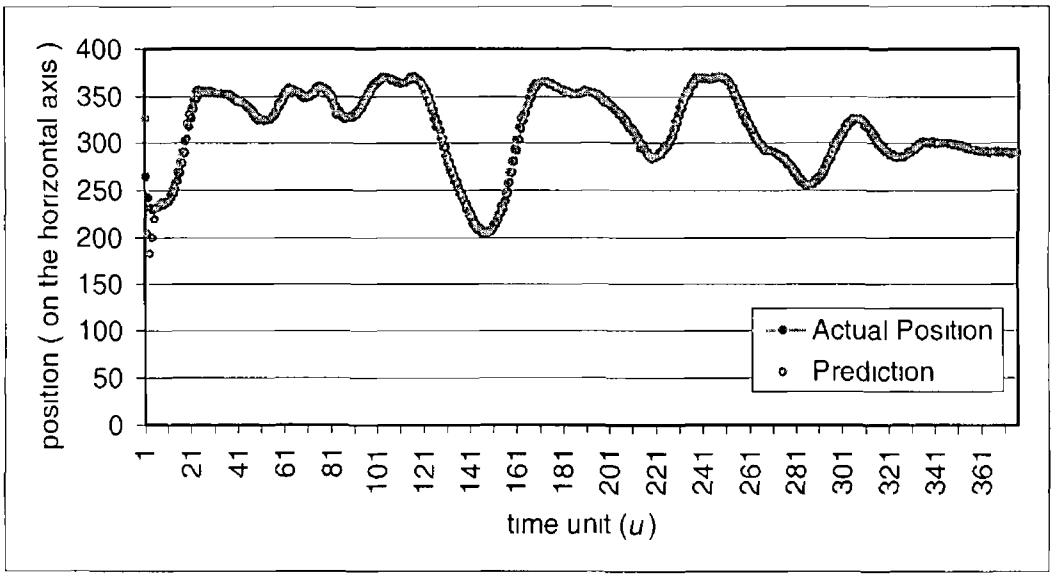
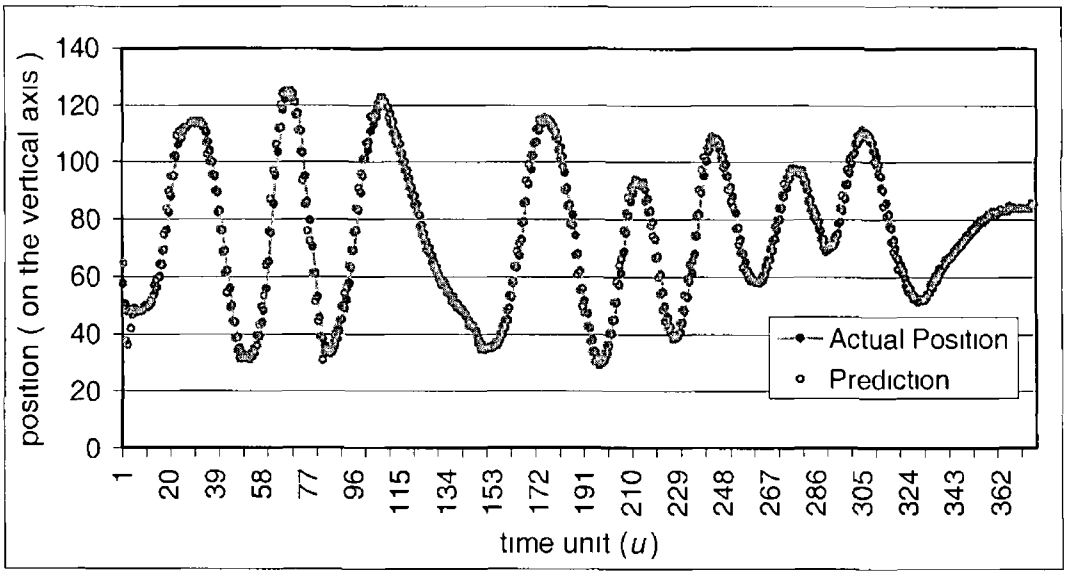


Figure 7 16 A closer view of the graph of the actual position and the prediction by the dynamic model





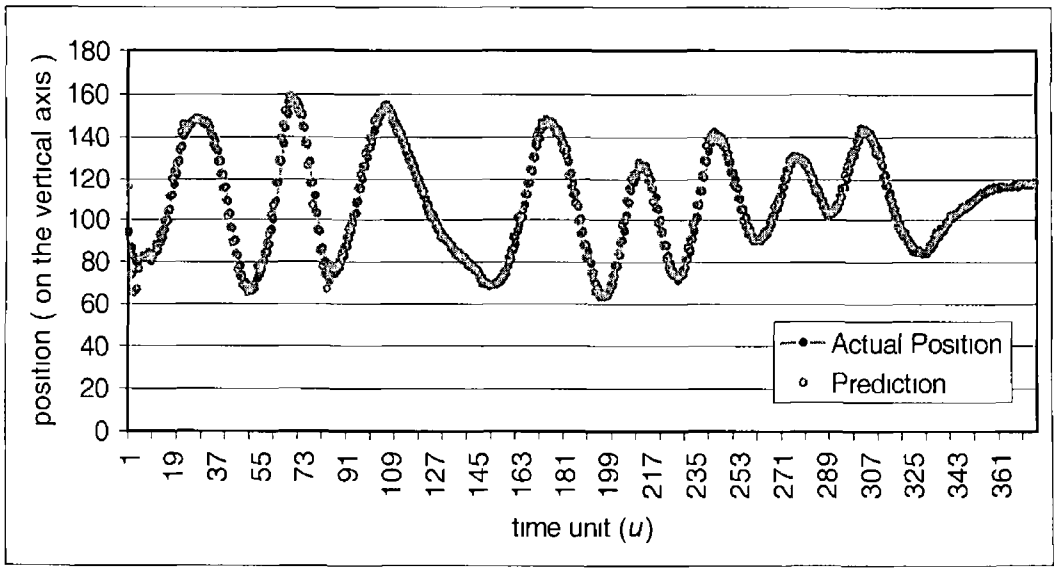


Figure 7 17 The positions and predictions of the rectangle sides

In the second experiment the complete algorithm of occlusion detection and tracking during occlusion is employed to track the hands in a movement in which the hands pass each other (see Figure 7 10 (h)) Some frames of this movement are presented in Figure 7 18 In the fifth to the ninth frame we see that the algorithm has detected occlusion and the occlusion rectangle around the both hands is formed From the tenth frame it is visible that the algorithm has tracked and labelled the hands correctly The hand labels are the small vertical lines on top of the hand rectangles The hand with one line is the right hand and with two lines is the left hand

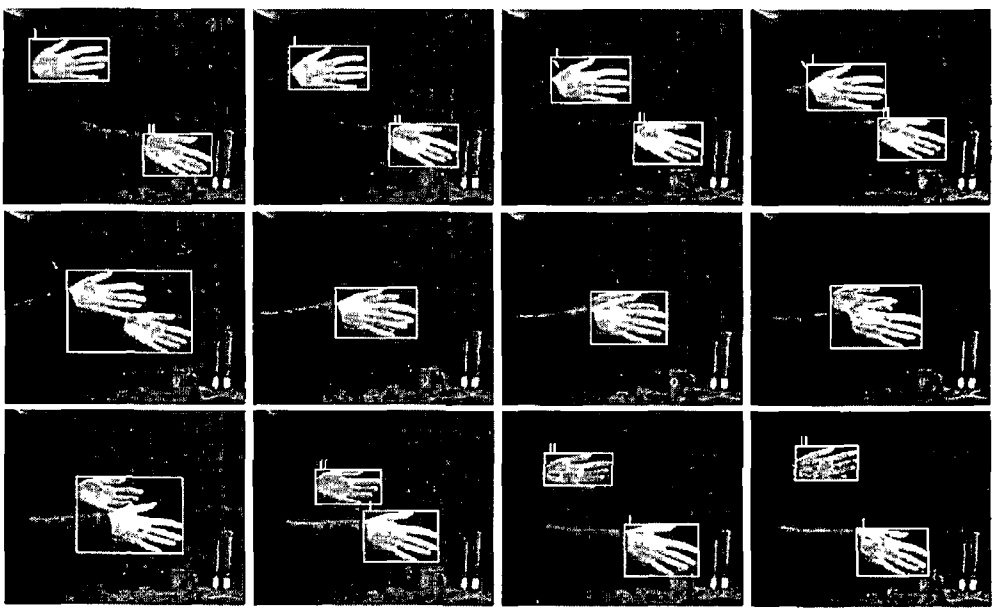
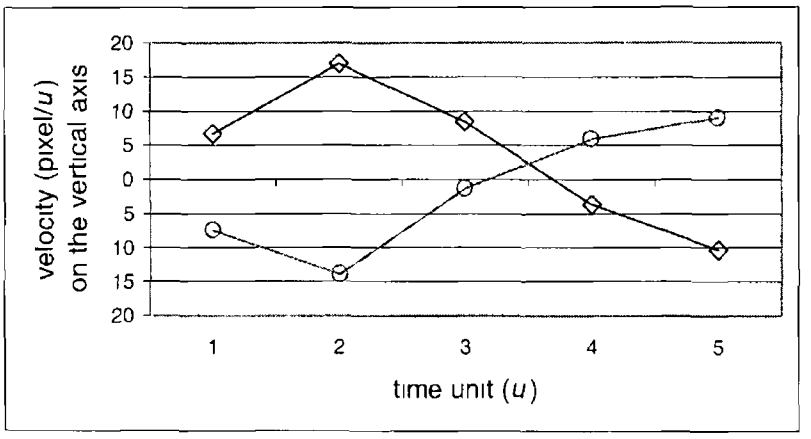


Figure 7 18 A bimanual movement of type *b* The rectangle with one small vertical line on top denotes the right hand and the rectangle with two small vertical lines denote the left hand The hands are tracked correctly at the end of occlusion

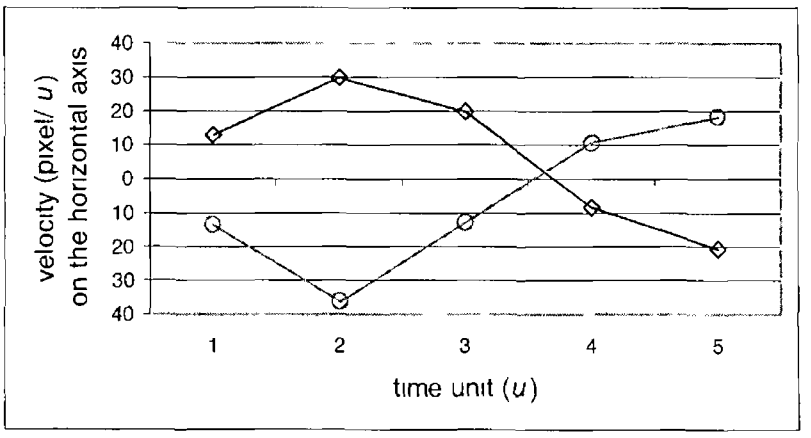
To investigate the process of tracking we look at the graph of the occlusion-rectangle sides The graph of velocities of the horizontal sides of the occlusion-rectangle is shown in Figure 7 19(a) and for the vertical sides in Figure 7 19(b) These velocities belong to the frames in Figure 7 18 with hand-hand occlusion

In this experiment, since the hands pass each other in opposite directions (see Figure 7 18) we observe opposite movements in the rectangle-sides velocities These velocities are negatively synchronised Large standard deviations in this experiment enable us to detect the opposite movements of the hands The values of velocity-synchronisation model are 18 035 for the horizontal sides and 8 828 for the vertical sides In the graphs of Figure 7 19 we can see that at no point the velocity of the parallel sides reach zero together The graphs of the

hand-pause model (System 7 10) for the parallel sides of the occlusion-rectangle are plotted in Figure 7 20

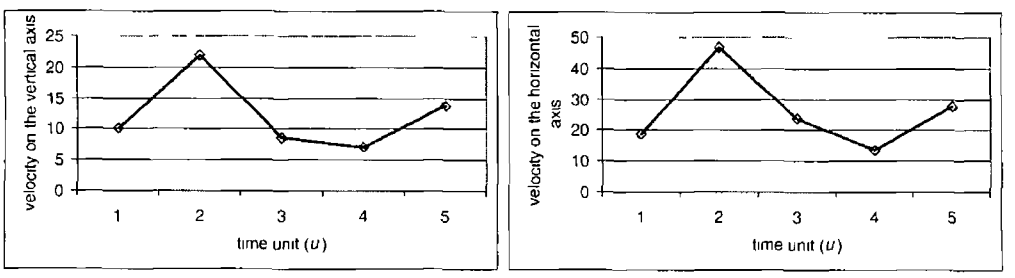


(a)



(b)

Figure 7 19 Velocities of (a) the horizontal sides of the occlusion rectangle (on the vertical axis) (b) the vertical sides of the occlusion rectangle (on the horizontal axis)



(a) (b)

Figure 7 20 The graph of the hand pause model for (a) the horizontal sides of the occlusion rectangle (on the vertical axis), (b) the vertical sides of the occlusion rectangle (on the horizontal axis)

In Figure 7 19 in all the velocities we observe a sign change at some stage Since the sides of the rectangle represent the rightmost, leftmost, top and bottom of the big blob during

occlusion, when one hand passes over the other they push the sides in the opposite directions Therefore, a sign change in the velocities is observed

In the third example we look at the dynamic model in detecting hand pauses Some images of an experiment are shown in Figure 7 21 In this movement (clapping) of type *g* the hands collide and return in opposite directions In the 11th image (the 7th frame during occlusion) the grey box at the left of image shows the detection of hand pause in the horizontal direction Immediately, in the 12th frame (the 8th frame during occlusion) the algorithm detects the vertical pause The grey box on top left of the 12th frame shows the pause detection on the vertical direction

An interesting point is the independence of the algorithm from the hand shapes It is shown in this experiment that during the whole movement the hand shapes are changed but the algorithm keeps tracking and labelling them correctly In this algorithm we do not recognise the shapes and therefore it is independent from hand shape which is a great advantage Processing hand shapes is usually a time consuming process In a real-time hand tracking applications time is so precious The less processing the faster running

In order to investigate the algorithm in this example the velocities of the parallel side of the occlusion-rectangle are plotted in Figure 7 22 In these graphs at the 7th frame during occlusion the velocities of the vertical sides (on the horizontal axis) reach almost zero (see Figure 7 22(a)) Therefore, the algorithm detects the pause in the horizontal direction The velocities of the horizontal sides of the rectangle reach zero at the 8th frame during occlusion (see Figure 7 22(b)) The graphs of the hand-pause model are plotted in Figure 7 23 These graphs show that the hand's horizontal pause was detected at the 7th frame and the vertical pause at the 8th frame

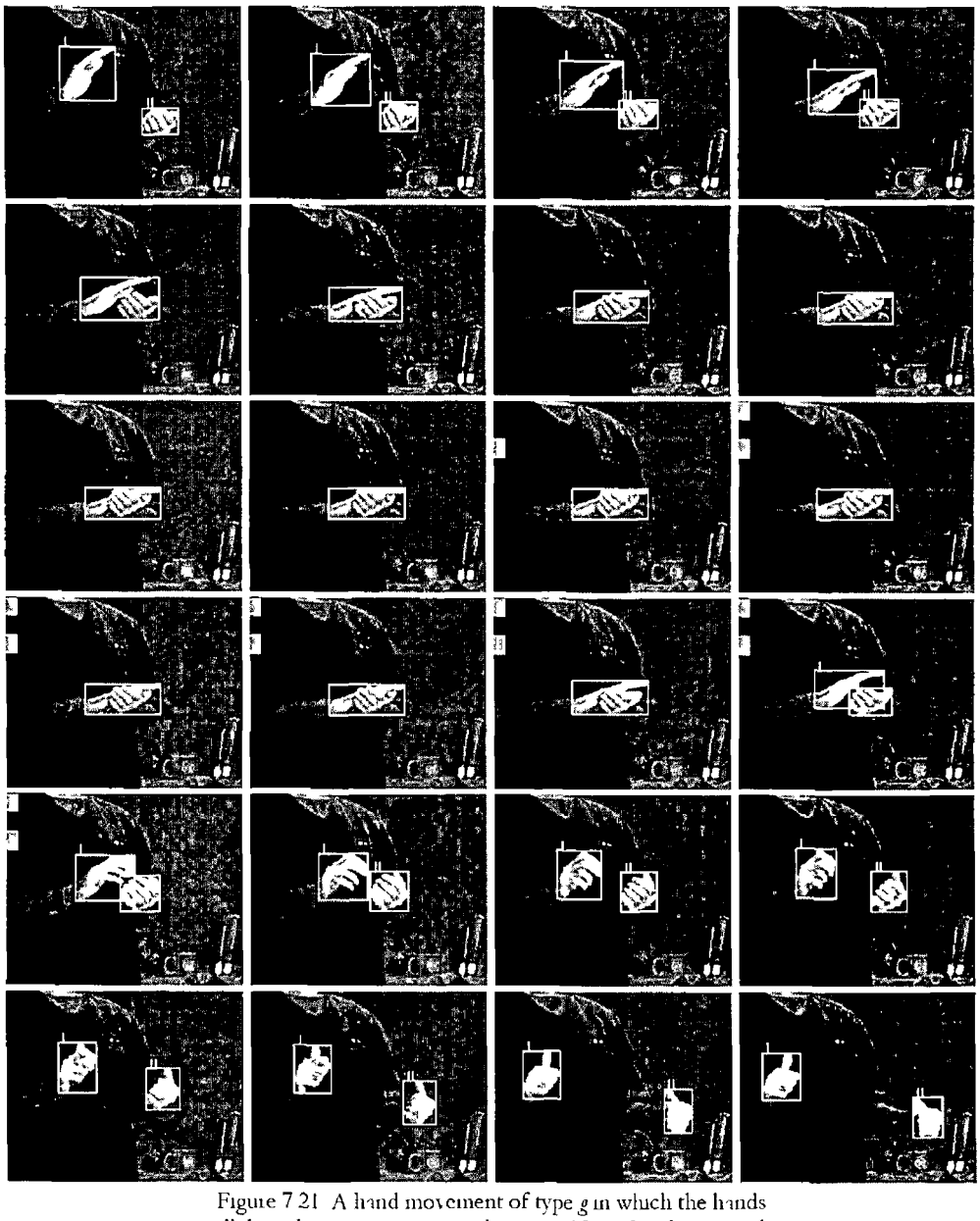
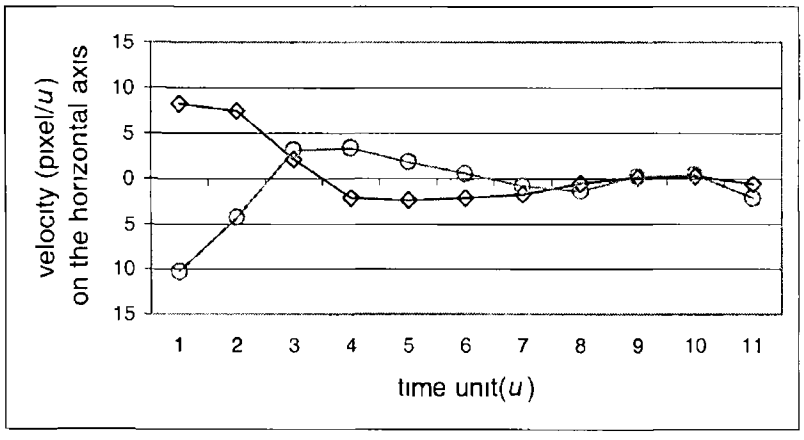
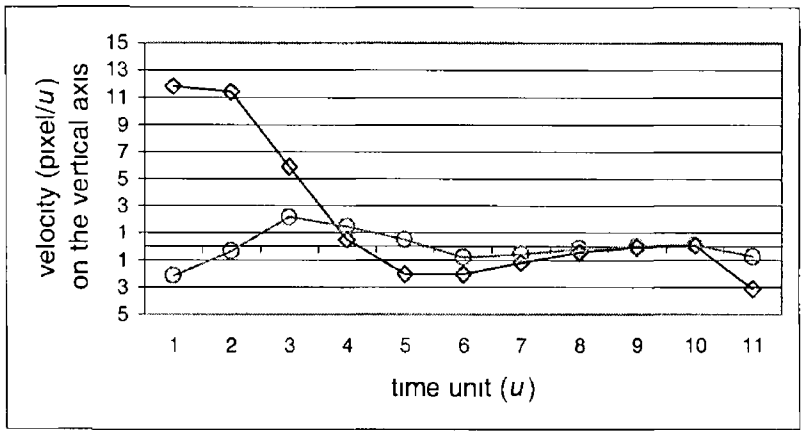


Figure 7 21 A hand movement of type *g* in which the hands collide and return in opposite direction Note the change in the hand shape before and after occlusion

Note that the zero threshold for the horizontal axis is twice the zero threshold for the vertical axis This is due to the fact that the camera is capturing an area of 70×70cm in a 400×200 pixels frame Therefore, in a diagonal movement where the horizontal and vertical velocities of the hand are equal the dynamic model estimate the horizontal velocity as twice as the vertical velocity

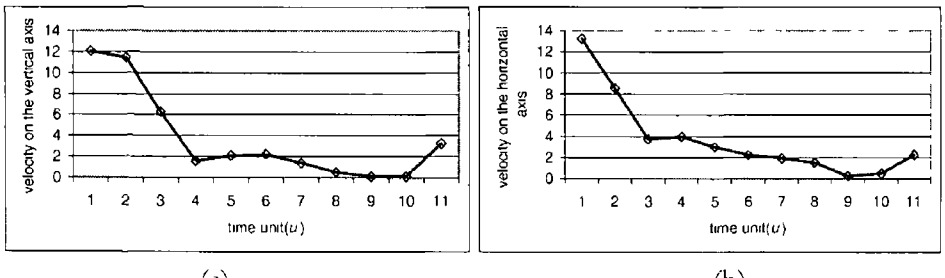


(a)



(b)

Figure 7 22 Graphs of the velocities of parallel sides of the occlusion rectangle (a) vertical sides (b) horizontal sides

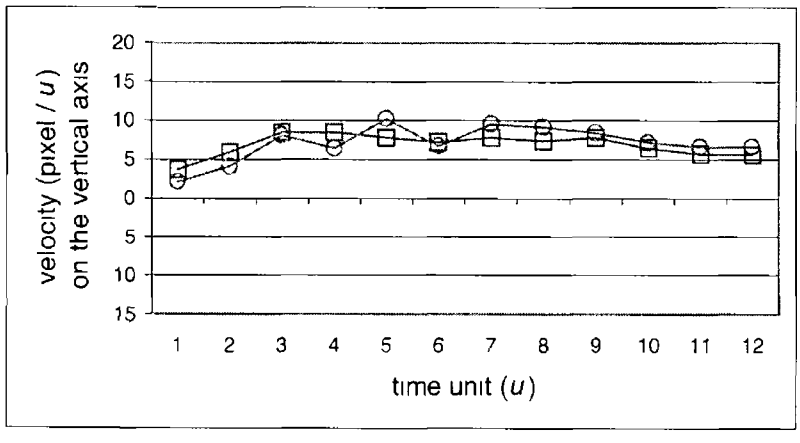


(a) (b)

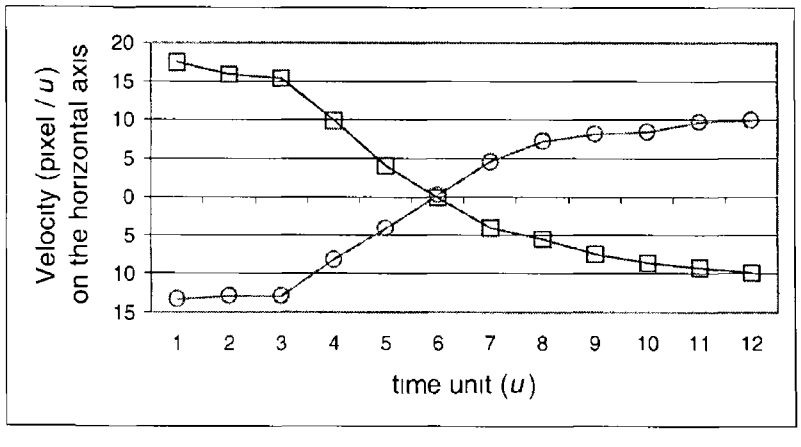
Figure 7 23 Graph of the hand pause model for (a) the horizontal sides of the occlusion rectangle (b) the vertical sides of the occlusion-rectangle

Another movement of type *a* demonstrates synchronisation between the velocities of the horizontal sides of the occlusion-rectangle For this experiment the graphs of the velocities are plotted in Figure 7 24 It is clear that the horizontal sides are positively synchronised with quite similar velocities The velocity-synchronisation model, which is used to catch this

synchronisation, gives the values presented in Table 7 1 The similarity of the horizontal-sides movement is better presented by a high-low lines graph in Figure 7 25



(a)



(b)

Figure 7 24 Graph of the velocities of parallel sides of the occlusion rectangle in a movement of type *a* (a) horizontal sides (b) vertical sides

Table 7 1 Values of velocity synchronisation model for the parallel sides of the occlusion rectangle

Rectangle Sides	Standard Deviation of the Relative Velocities
Horizontal sides	$S_v = 1\ 49$
Vertical sides	$S_h = 20\ 13$

The positively synchronised movement of the horizontal sides is caught by the model and the decision is made based on the pauses of the vertical sides

The negative synchronisation of the vertical sides of the rectangle is observable in Figure 7 24(b) In this figure, the rectangles sides have pretty similar velocities but in opposite directions

From Figure 7 24(b) it seems that at the 6th frame the velocities of the vertical sides reach zero but the algorithm keeps tracking correctly Although, in this example the algorithm tracked the hands correctly, there are some cases occasionally in which the algorithm makes wrong decisions We will measure the performance of the algorithm later

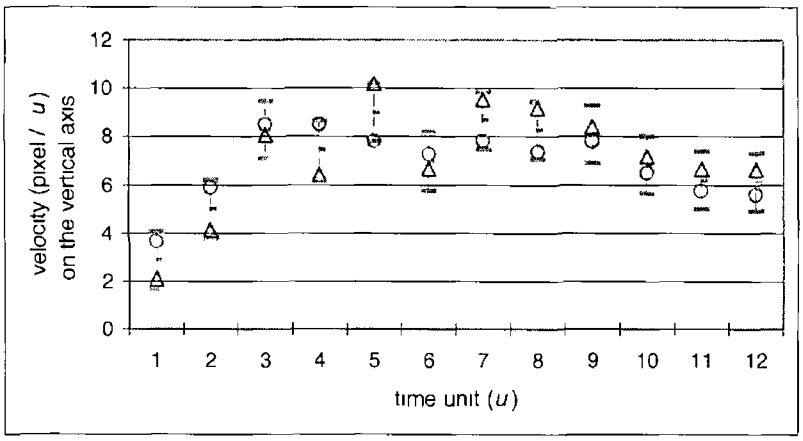
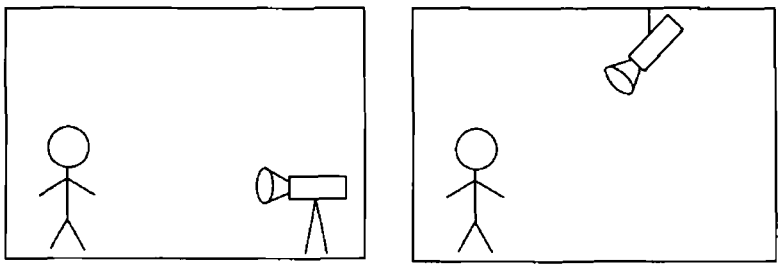


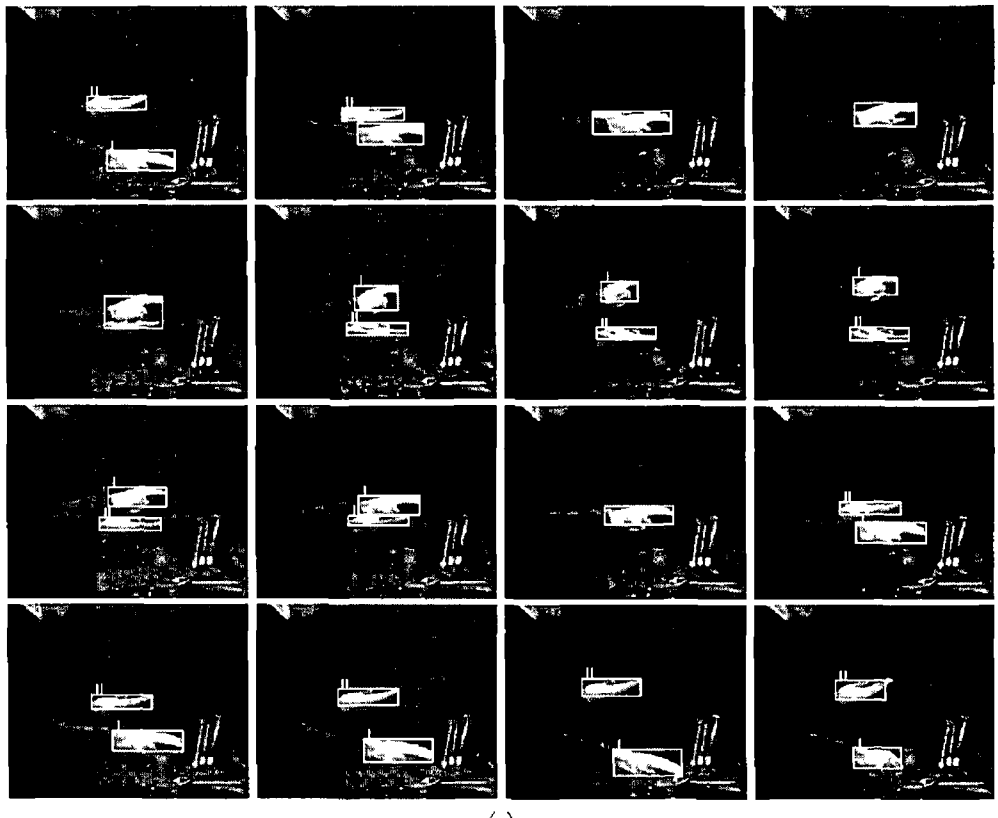
Figure 7 25 The velocities of the parallel sides of the occlusion rectangle are positively synchronised with a similar movement which is caught by the velocity synchronisation model

The fifth experiment is presented to demonstrate the independence of the tracking algorithm from the camera's angle of view and the type of movement In this experiment, a bimanual movement is performed twice In the first time, the camera is placed on the side looking at the hands horizontally (see Figure 7 26(a)) In the second example we changed the position of camera to look at the scene from a top-corner view (see Figure 7 26(b)) The selected movement is so that from the side view the hands pass each other but from the top-corner view they pause and return to their previous sides The results of tracking are presented in Figure 7 27 In 7 27(a) the movement is shown from the side view In 7 27(b) the movement is shown from the top-corner view As can be seen the algorithm tracks the hands correctly in both examples

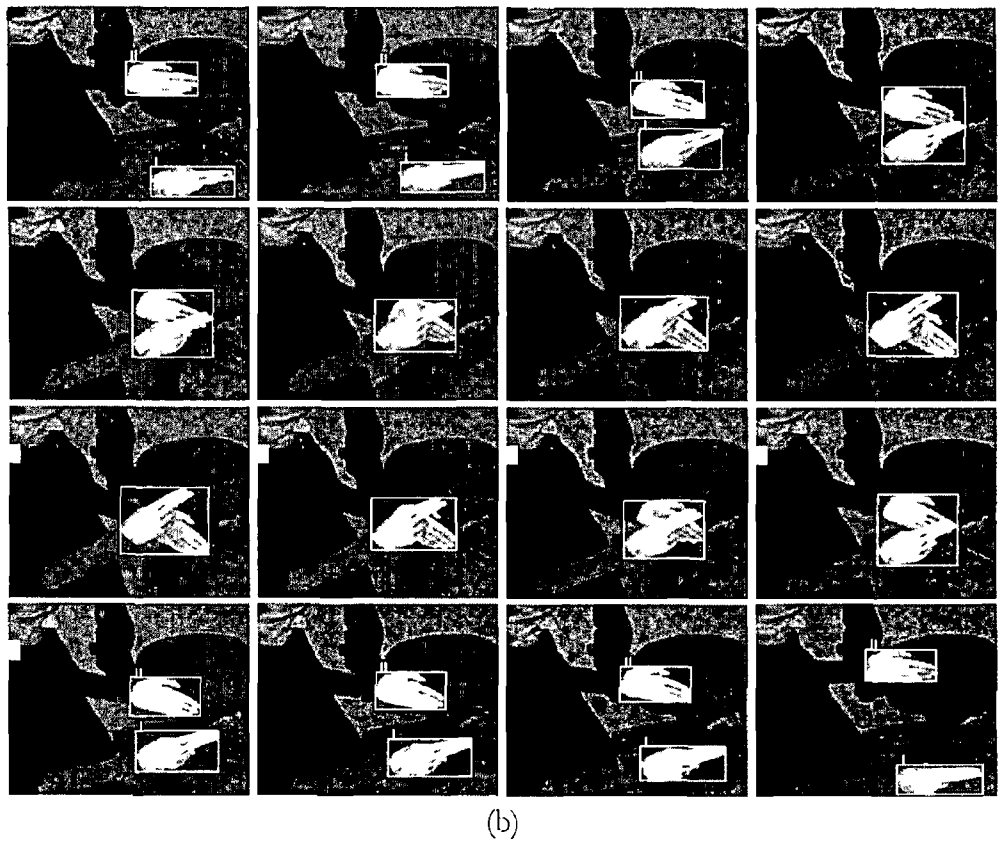


(a) (b)

Figure 7 26 The two angle of views (a) side view, (b) top-corner view



(a)



(b)

Figure 7 27 A bimanual movement seen from two angle of views,
(a) from the side view (b) from the top corner view

In order to evaluate the algorithm 3500 experiments associated with different models were performed with many different hand shapes The results of the experiments are presented in Table 7 2 It is shown that the algorithm is able to track the hands correctly in almost 90% of times

Table 7 2 *Performance of the tracking algorithm using the dynamic model*

# Movements (events)	# Errors	Error Rate (%)	Tracking Rate (%)
3500	351	10 03	89 97

An important factor in the bimanual movements is the speed of movement A movement can be performed slowly, moderately, or fast In our experiments we tried to include all classes of speeds Therefore, in some of the movements where the period of occlusion is short, e g movement of type *h*, the camera speed brings some restriction to the algorithm In this case the number of images taken during occlusion should be large enough so that the algorithm can detect hand collisions, e g the movements of type *h*, or hand pauses, e g the movements of type *e* Particularly, when a movement is performed fast very few images are captured during occlusion The Kalman filtering process is based on the Kinematic equations of motion Therefore, in a fast movement the sides of the occlusion-rectangle have the potential to move further rather than to stop quickly In other words, a large difference between the position of a rectangle side in two consecutive images causes the algorithm to estimate a high velocity for the side Therefore, it will be hard to detect the quick stop of hands in a few consecutive images containing a fast movement

In order to overcome this problem we changed the mechanism of pause detection of the algorithm In this case, the pattern of the velocity changes of the rectangle sides during the occlusion period is determined to match one of the previously trained patterns for hand-pause or hand-pass The new algorithm has demonstrated a good performance The details of the changes are presented in Appendix E

The other important parameter is the processing speed We did many experiments in order to measure the processing speed of the algorithm With a fast camera working in 120 frames per second on a Pentium II, 1 GHz the algorithm is able to process 37 5 frames per second in average It means that every image takes 26 6 milliseconds to be processed by the algorithm on this machine

Although some other groups have done some work on hand tracking in the presence of occlusion, none of them has considered the problem of tracking the hands in bimanual movements so widely The CONDENSATION algorithm [Isard 1998a] has been implemented by Gong et al [Gong 2002] works at a low speed able to process an image in 27 seconds on a Pentium II, 330 MHz Gong et al have also a Bayesian Network-based technique for modelling the semantics of interactive behaviors able to process 5 frames per second on a Pentium II, 330 MHz [Gong 2002] Their result of 13% error is based on the number of images in the database and not the number of events In our experiments of Table 7 2 the tracking results are event-based in which each event (movement) may consist dozens of images If we present our measurement in this way, assuming that on average the number of images before the occlusion period and after the occlusion period are almost equal in a movement, the performance of the algorithm increases to approximately 95% This is due to the fact that before the occlusion period the hands are correctly tracked and the number of fails corresponds to the images after the occlusion periods

All the experiments and results presented here demonstrate that the hands have quite similar velocities in the same or opposite directions in bimanual movements In other words, they are either positively or negatively synchronised This reconfirms the coordination in bimanual movements

Summary and Conclusion

A dynamic model based on the Kinematic equations of motion was presented to track the hands in bimanual movements A procedure for predicting the future movement of the hands in order to detect possible hand-hand occlusions was introduced Therefore, we were able to detect the exact moment that occlusion happens and ends in a bimanual movement

We introduced a novel intelligent algorithm based on the dynamic model to track and reacquire the hands in a movement where hand-hand occlusion exists Based on the bimanual coordination phenomenon we presented a model to capture the coordination in the bimanual movements in order to detect the positive or negative synchronisation of the hands Also, the concurrent hand pauses were detected in a model to track the hands in different types of bimanual movements presented in this chapter

We presented some experimental results in which the proposed tracking algorithm was examined under different types of movements We also demonstrated that the algorithm is

independent from the type of movement, changing hand shapes, and the angle of view In this order, it was shown that given that the hand shapes are changed in the movement the algorithm tracks the hands correctly Also, we changed the camera's angle of view and tested a movement in which from a view direction it belongs to a type of movement while from another view it belongs to another type

We will use this algorithm in the next chapter for tracking and movement segmentation in order to recognise bimanual movements

Chapter 8

RECOGNITION OF BIMANUAL MOVEMENTS

In the last chapter we introduced an intelligent tracking algorithm for tracking the hands in bimanual movements in which the hands are partially or completely occluded for some moments. Now, we move forward to recognise bimanual movements.

By using the tracking algorithm we can separate the hands from each other and look at the movement of each hand individually in order to understand the whole bimanual movement. The meaning of each hand movement must be combined so that the bimanual movement is recognised as a single entity. We introduce a Bayesian network for the recognition of bimanual movements. First, a quick review of Bayesian networks and the belief propagation algorithm is presented. Then the Bayesian network for recognition is described in detail. In this network Hidden Markov Models are employed to recognise the partial movements of the hands. Partially recognised movements are fused at different levels of the network to form the whole bimanual movement. The movement is recognised at the top node of the network as a single movement constituted from the partially recognised movements.

For another set of movements called concatenated periodic bimanual movements we change the belief propagation algorithm to stabilise the belief of the network in these movements. A short-term memory is applied to the network for the stabilisation of the belief.

A Bayesian network containing a single loop is also introduced for the recognition process. We deeply explore some of the problems involved in the loopy networks. The reasons for the convergence of the loopy network for the recognition of bimanual movements are presented. We also explore the parameters involved in the convergence rate of the loopy networks. A new analytical framework is presented to formalise the conditions where the loopy networks converge rapidly. We assess the presented networks on the two sets of test data, and show that all the proposed networks recognise the single bimanual movements very accurately. In the case of concatenated periodic movements the networks demonstrate different recognition rates which are reported at the end of this chapter.

8 1 Bayesian Networks

"A *Bayesian network* is a graphical model that encodes probabilistic relationships among variables of interest" [Heckerman 1996] Over the last decade, Bayesian networks have been used for probabilistic reasoning in a wide range of applications from medical to forensic investigation Particularly, Bayesian networks are employed to combine a set of diagnostic data in order to make a decision There are other similar methods such as Dempster-Shafer [Pearl 1998] or Fuzzy Logic [Zadeh 1992], which are used in the literature for data fusion Although, these methods are in many ways similar to each other, each application needs particular models based on the type of application and the required result We will discuss our application and the reasons about choosing Bayesian networks instead of e g Dempster-Shafer later on in this chapter

The heart of Bayesian networks lies in Bayes' rule,

$$p(H \mid e) = \frac{p(e \mid H)p(H)}{p(e)} \qquad (8\ 1)$$

where H is the hypothesis and e is the evidence This formula states that the belief that H is true upon obtaining evidence e can be computed by multiplying our previous belief $p(H)$ by the likelihood $p(e \mid H)$ that e will be materialised if H is true The term $p(e)$ is the denominator that is used for normalisation

Causal trees are Bayesian networks in which every new piece of evidence e propagates through via message-passing In causal trees each node has at most one parent (see Figure 8 1)

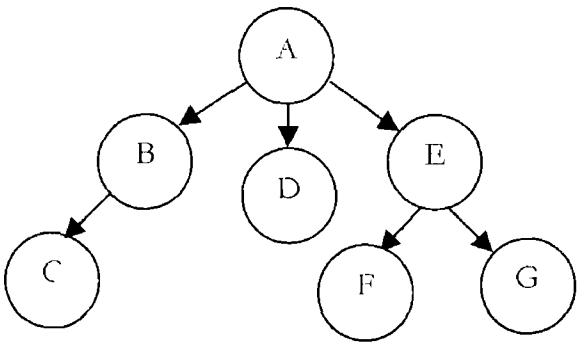


Figure 8 1 A causal tree

Another type of Bayesian network is the *causal polytree* In this type of network each node may have more than one parent (see Figure 8 2) There are other types of Bayesian networks

which can be found in the literature [Pearl 1988], [Neapolitan 1990], [Cowell 1999] A set of networks are the causal polytrees containing loops Regular rules of causal trees and polytrees do not hold for this type of Bayesian network We will discuss these networks further in the next sections

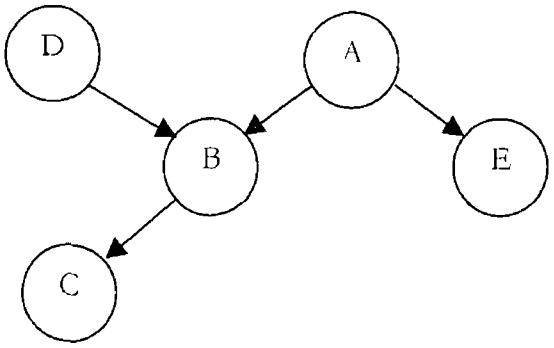


Figure 8 2 A causal polytree

8 1 1 Belief Propagation in Causal Trees

Pearl has proposed a message-passing technique [Pearl 1988] in which the local belief at each node of a causal tree is updated by the message received from the neighbouring nodes These local belief propagation rules are guaranteed to converge to the optimal belief for *singly connected networks* A singly connected network is a network in which no more than one path exists between any pair of nodes Causal trees are singly connected

Suppose that we have a tree depicted in Figure 8 3

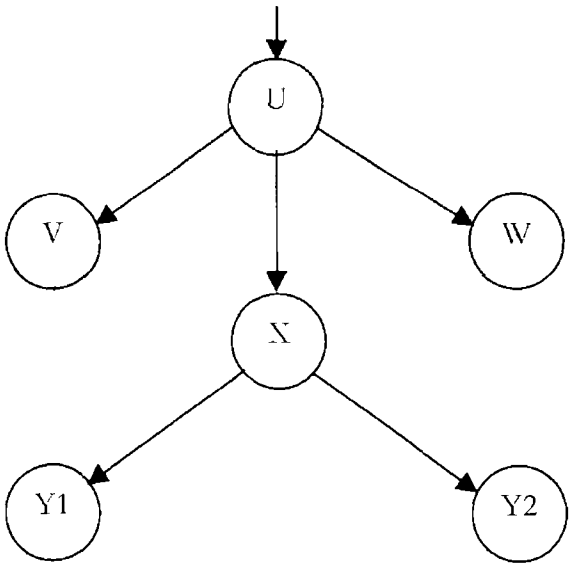


Figure 8 3 A part of a causal tree

A message which is communicated by a child to its parent is expressed as λ The messages that is sent by a parent to its children is expressed as π Assuming a typical node X has m children, $Y_1, Y_2, \ldots, Y_m$ and a parent U the belief propagation rules are summarised as following,

- **Belief updating**

$$\boldsymbol{\lambda}(x) = \prod_j \boldsymbol{\lambda}_{Y_j}(x) \tag{8 2}$$

$$\boldsymbol{\pi}(x) = \boldsymbol{\pi}_x(U)\ \mathbf{M}_{XU} \tag{8 3}$$

$$\mathbf{BEL}(x) = \alpha\boldsymbol{\lambda}(x)\boldsymbol{\pi}(x) \tag{8 4}$$

where $\boldsymbol{\lambda}_{Y_j}(x)$ stands for the message communicated from the j^{th} child of node X, which is the current strength of the diagnostic support,

$\boldsymbol{\pi}_x(U)$ stands for the message communicated from X's parent which is the current strength of the causal support,

α is a normalising constant,

$\mathbf{M}_{XU}$ is the conditional probability matrix, in which the (x, u) entry is given by

$$\mathbf{M}_{XU} = P(x \mid u) = P(X = x \mid U = u)$$

where if the variables at the nodes of the tree are multi-valued with $x_1, x_2, \ldots, x_n$ and $u_1, u_2, \ldots, u_m$

$$\mathbf{M}_{XU} = \begin{bmatrix} p(x_1 \mid u_1) & p(x_2 \mid u_1) & p(x_n \mid u_1) \\ p(x_1 \mid u_2) & p(x_2 \mid u_2) & p(x_n \mid u_2) \\ & & \\ p(x_1 \mid u_m) & p(x_2 \mid u_m) & p(x_n \mid u_m) \end{bmatrix} \tag{8 5}$$

- **Bottom-up propagation**

$$\boldsymbol{\lambda}_x(u) = \mathbf{M}_{XU}\ \boldsymbol{\lambda}(x) \tag{8 6}$$

where $\boldsymbol{\lambda}_{\cdot}(u)$ is the message calculated at node X, using the $\boldsymbol{\lambda}$ messages received, to be sent to its parent U

- **Top-down propagation**

$$\boldsymbol{\pi}_{Y_j}(x) = \alpha\boldsymbol{\pi}(x)\prod_{k\neq j}\boldsymbol{\lambda}_{Y_k}(x) \tag{8 7}$$

where $\boldsymbol{\lambda}_{Y_j}(x)$ are the messages calculated at node X to be sent to each of its children

In Equations 8 2 to 8 7 the parameters have the following probabilistic meanings,

$$\boldsymbol{\lambda}_X(U) = P(\mathbf{e}_{\cdot}^{-} \mid u) \tag{8 8}$$

$$\boldsymbol{\pi}_{\cdot}(x) = P(x \mid \mathbf{e}_{\cdot}^{+}) \tag{8 9}$$

$$\mathbf{BEL}(x) = P(x \mid \mathbf{e}) \tag{8 10}$$

where $\mathbf{e}$ stands for the total evidence available, $\mathbf{e}_{\cdot}^{-}$ is the evidence contained in the tree rooted at X, and $\mathbf{e}_{\cdot}^{+}$ stands for the evidence contained in the rest of the network

8 1 2 Causal Polytrees

In causal polytrees the belief propagation is summarised as following,

- Combine all messages coming into X except for that coming from Y into a vector $\mathbf{v}$ by multiplying all the message vectors element by element
- Multiply $\mathbf{v}$ by the matrix $\mathbf{M}_{XY}$ corresponding to the link from X to Y
- Normalise the product $\mathbf{M}_{XY}\mathbf{v}$ The normalised vector is sent to Y
- The belief vector of node X is obtained by combining all incoming messages to X and normalising

Details of the propagation rules for singly connected networks can be found in [Pearl 1988], [Neapolitan 1990], [Weiss 2000]

Pearl in [Pearl 1988] discusses that the absence of loops in the network permits us to develop a local updating scheme similar to that used for causal trees Local belief propagation rules

proposed by Pearl are guaranteed to converge to optimal beliefs in singly connected networks

8 2 Recognition of Bimanual Movements

Recognition of bimanual movements can be accurately achieved by recognising the gesture of each hand separately

First, in a sequence of images, we have to separate the hands from each other Then for each hand we must recognise the gesture At the end the recognition of the whole bimanual gesture can be achieved by fusing the results of each individually recognised hand gesture

8 2 1 Hand Tracking and Separation

In order to separate the hands we use the tracking algorithm of Chapter 7 By this algorithm we track the hands individually in a sequence of images Therefore, we are able to separate the movement of each hand while no hand occlusion exists However, when we have occlusion the hands are not separately recognisable Thus, we cannot separate the movements of the hands

There can be two solutions for this problem We can ignore the parts of the movement with hand occlusion and just recognise what we see in the whole movement excluding occlusion parts However, in many gestures such as time signs in British Sign Language the important part of the gesture is when the hands are seen connected together (see Figure 8 4)

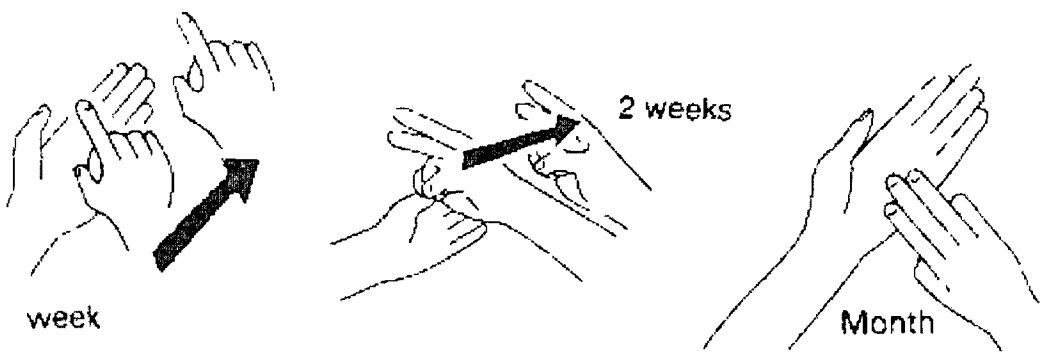


Figure 8 4 Signs for week two weeks and month in British Sign Language

Therefore, we cannot ignore the occlusion parts of a movement The second solution is to take the occlusion parts into account and recognise it separately Then, the recognised

individual movements of hands and the occlusion parts must be fused in order to understand the whole bimanual movement

Each hand is tracked and separately projected into a blank sequence of images (see Figure 8 5) In order to preserve the movement of the hands with respect to the image frame, the direction of movement of each hand is recorded For this we divide the 2-dimensional space of the image frame into 8 equal regions [Wu 2002] as in Figure 8 6

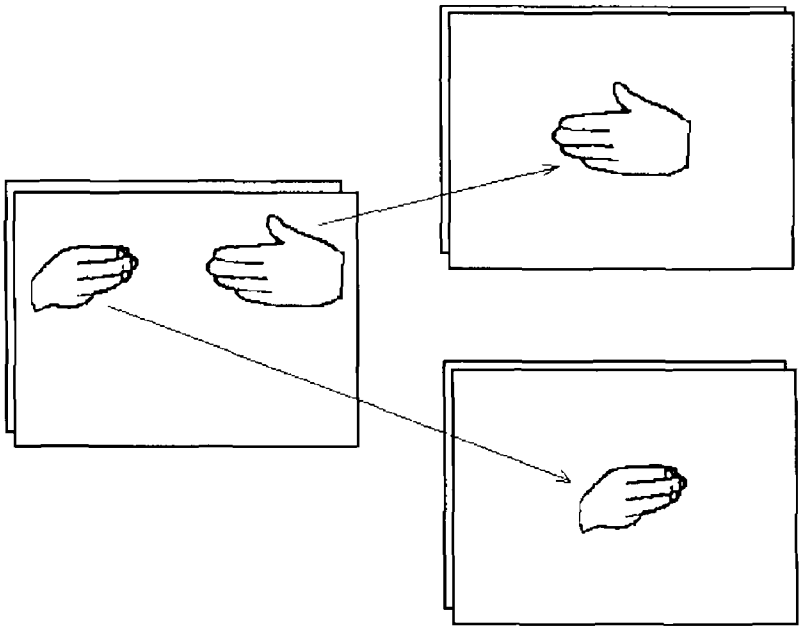


Figure 8 5 The hands movements are separated and projected into the blank sequences of images

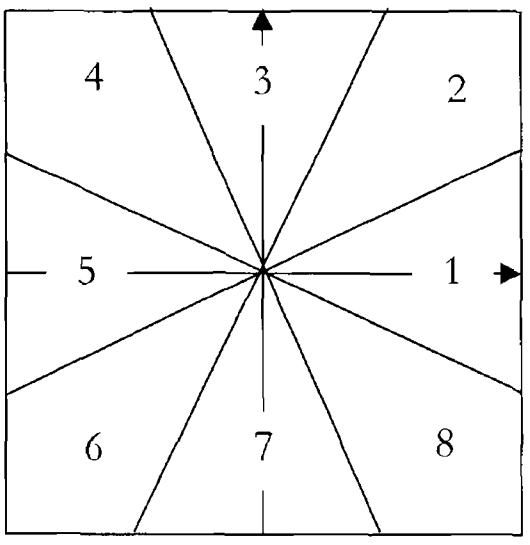


Figure 8 6 The image frame is divided into 8 equal regions

We call this the *regional-map* The index of each region represents the direction of movement in that region Index of zero represents immovability

By tracking the movement of the centre of each hand a vector representing the movement is extracted for every single frame This vector represents the movement from the last image to the present one (see Figure 8 7) The angle of the vector with respect to the horizontal axis determines the region in the regional-map in which the vector maps onto The region index is recorded for the movement at each time *t* Even for a partial sequence including hand occlusion the direction vector for the movement of the big blob is extracted and the region index is recorded

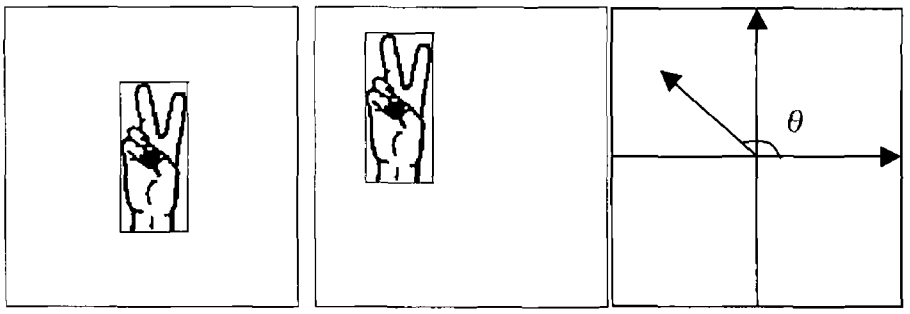


Figure 8 7 The extracted vector for a movement

8 2 2 Movement Segmentation

A bimanual movement is constituted from two groups of parts, the occlusion parts in which one hand is occluded and the other parts, we call them non-occlusion, where the hands are recognisable separately Since a bimanual movement can be a periodic movement like clapping we separate different parts, which we call segments Four segments are obtained as following,

- **A** The beginning segment, from the beginning of a gesture to the first occlusion part
- **B** The occlusion segments, where one hand is occluded
- **C** The middle segments, a part of the gesture between two consecutive occlusion segments
- **D** The ending segment, from the last occlusion segment to the end of gesture

An example of a segmented bimanual movement is illustrated over the time axis in Figure 8 8 Although we have assumed in this figure that the movement starts and ends in non-

occlusion segments, extending the algorithm to the other cases is straight-forward and makes no difference in the essence of the algorithm Also for the gestures in which no occlusion segment is observed the process is the same with only one segment for the whole gesture

A matrix representing the hand segments is created for a gesture We call it the segments-matrix In this matrix, every row is associated with a single frame in the captured image sequence The first column of the matrix represents the segment index, 1 for A, 2 for B, 3 for C and 4 for D

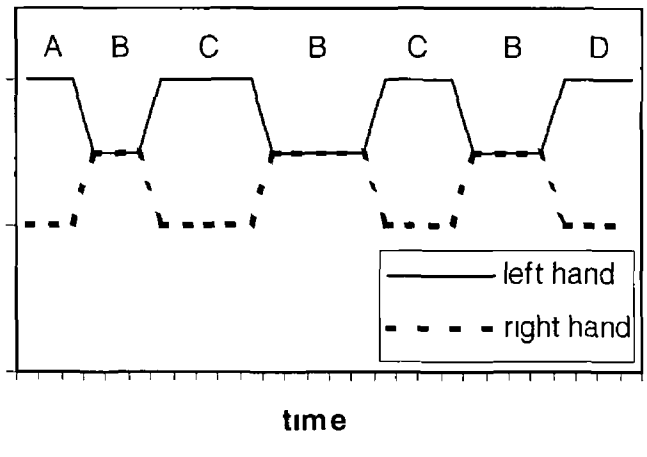


Figure 8 8 Segmentation of a bimanual gesture over a period of time The separate lines at segments A C, and D show the separated hands In segments B the overlaped lines show occlusion

The second column is the region index of the movement of the hand number 1 (normally the right hand) The third column represents the region index of the movement of the hand number 2

For the segments of occlusion the second column is the region index of the movement of the big blob and the third column is set to zero An example of the segments-matrix is presented in the Figure 8 9,

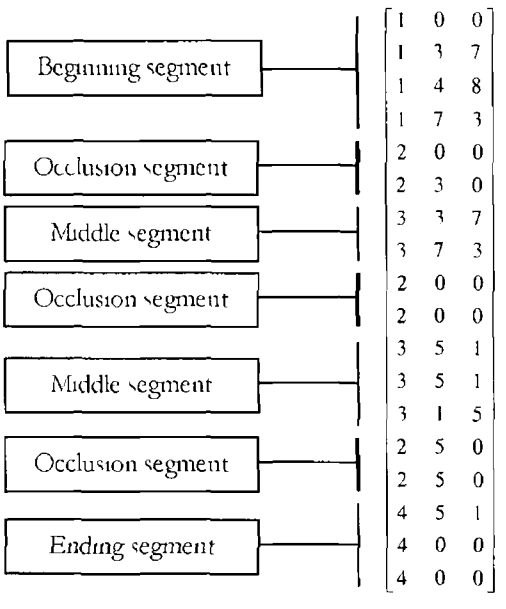
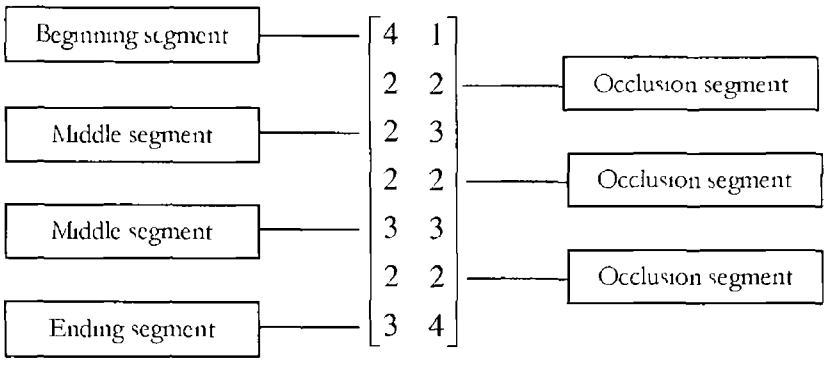


Figure 8 9 The segments matrix is used to record the segments index and the motion vector of each hand in an image The first column is the segment index, the second and third columns are the motion vectors of the first and second hands

For each hand a separate image sequence is recorded Also, a sequence of images is recorded for the occlusion segments Therefore, for a bimanual movement three files of images are recorded, one for each hand separately and one for the occlusion parts In order to synchronise the segments-matrix with the recorded image sequences we create another matrix called synchronisation matrix In this matrix, every row represents a segment The first column of this matrix is the number of images in a segment and the second column is the segment index For example, for the above segments-matrix the synchronisation matrix is extracted as following,



By using the synchronisation matrix we can extract the partial sequence of images of each segment from the recorded files In the above example, the first 4 images of the sequences of the hands belong to the beginning segment of the gesture The first two images of the file of occlusion are the first occlusion segment and so on

8 2 3 Bayesian Fusion of Partial Discrete Hidden Markov Models

In a bimanual movement there can be several occlusion and middle segments For example in Figure 8 8 there are 3 occlusion and 2 middle segments Thus, a data fusion structure must be able to deal with multiple occlusion and middle segments as well as the beginning and the ending segments in order to understand the whole bimanual movement

As it was mentioned earlier, there are different methods for data fusion The well-known methods of data fusion are Bayesian networks, Dempster-Shafer theory, Fuzzy Logic, and Neural Networks [Pearl 1988 Petrou 2001 Zadeh 1992 Theodoridis 1999] Each of these methods can be used to combine the information of the partial hand movements within different segments of a bimanual movement

The movement of a hand within a segment (or the two hands in an occlusion segment) can be treated as a single movement appearing in the sequence of images of the segment These partial movements can be modelled and recognised by Hidden Markov Models as explained in Chapter 6 Therefore, for a bimanual movement we get a set of recognised partial movements of the two hands and the occlusion parts We must combine this information to recognise the bimanual movement[1]

Dempster-Shafer theory is a method for data fusion [Pearl 1988] When we have a synthesis task where the constraints are imposed externally, our concerns centres on issues of possibility and necessity In this case, the Dempster-Shafer theory seems more suitable for anticipated queries On the other hand, the Bayesian networks are more suitable for the tasks of analysis (e g diagnosis) to piece together a model of physical reality [Pearl 1988] Therefore, if we consider the partial movements of hands as the pieces constituting a bimanual movement, Bayesian networks seems more suitable to be employed for our data fusion problem As a future work, however, we can use Dempster-Shafer theory, Fuzzy Logic and Neural Networks for the fusion task and compare them with the Bayesian networks in bimanual movement recognition

An alternative to all the above techniques is the Coupled Hidden Markov Models [Brand 1997] Although this model has been used to model interactive hands [Brand 1997], a major weakness is that this model is unable to deal with occlusion In this model the two hands

[1] There are three types of fusion reported in the literature [Petrou 2001], data level fusion, feature level fusion and decision level fusion Our approach to recognise the partial hand movements by Hidden Markov Models and combine them by each of the data fusion methods is an example of decision level fusion

must be separately recognisable throughout the whole image sequence Therefore, no occlusion (including hand-hand occlusion) can be dealt with using Coupled HMM Since occlusion may happen in most bimanual movements we cannot ignore this part as we mentioned earlier and a solution must be able to deal with it On the other hand, in some of the movements one hand may be occluded by something else (e g the body) or leave the scene in a segment other than occlusion segment A general solution must deal with this type of occlusion too Thus, we must recognise the movement of the hands separately so that we can deal with the segments containing two hands or one hand as well as (hand-hand) occlusion segments

We introduce a Bayesian network in which the whole gesture is divided into the movements of the two hands The movement of each hand is also divided into the four segments (see Figure 8 10) In this figure, the BEG, MID, OCC, and END are the evidence nodes The occluded part of a gesture is a common part for the both hands Therefore, a single shared node is considered According to the number of cases it can accept, each node in this tree represents a multi-valued variable Thus, for a vocabulary containing g bimanual gestures every node is a vector with length g The three top nodes of *Bimanual Gesture*, *Left Hand Gesture*, and *Right Hand Gesture* are non-evidence nodes updated by the messages communicated by the evidence nodes The evidence nodes are fed by the partial Discrete Hidden Markov Models of different segments separately

Due to the fact that the beginning, middle, and ending segments of a gesture have no time overlapping, and assuming that the segments are of equal weight, the causal tree can be abstracted to the tree depicted in Figure 8 11 The NS nodes represent the evidences of the beginning, middle, and ending segments at different times for each hand These evidences are the normalised vectors of likelihoods provided by the partial Discrete Hidden Markov Models at the lowest level of the network These values represent the likelihoods that a given partial gesture is each of the gestures in the vocabulary in the corresponding segment
How do the partial Discrete Hidden Markov Models work and calculate the likelihoods? We discuss this in the next section

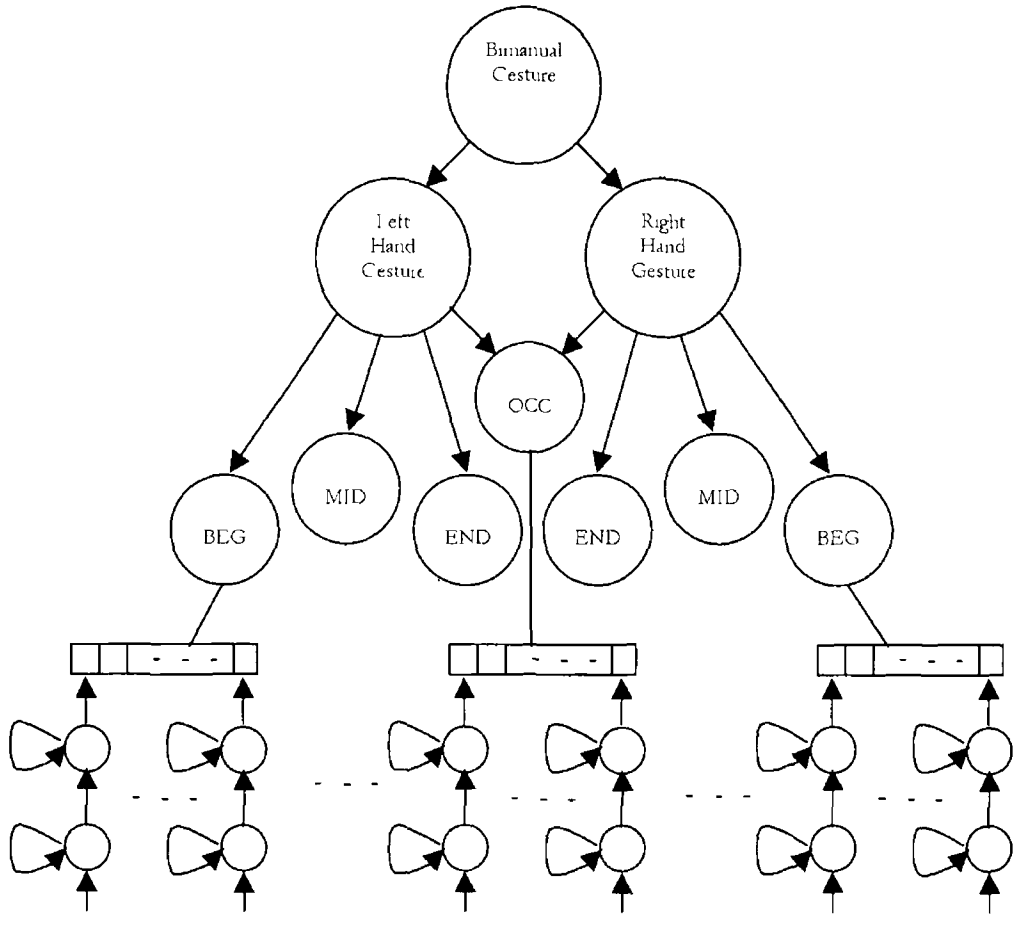


Figure 8 10 A Bayesian network for fusing partial Discrete Hidden Markov Models for the recognition of bimanual movements

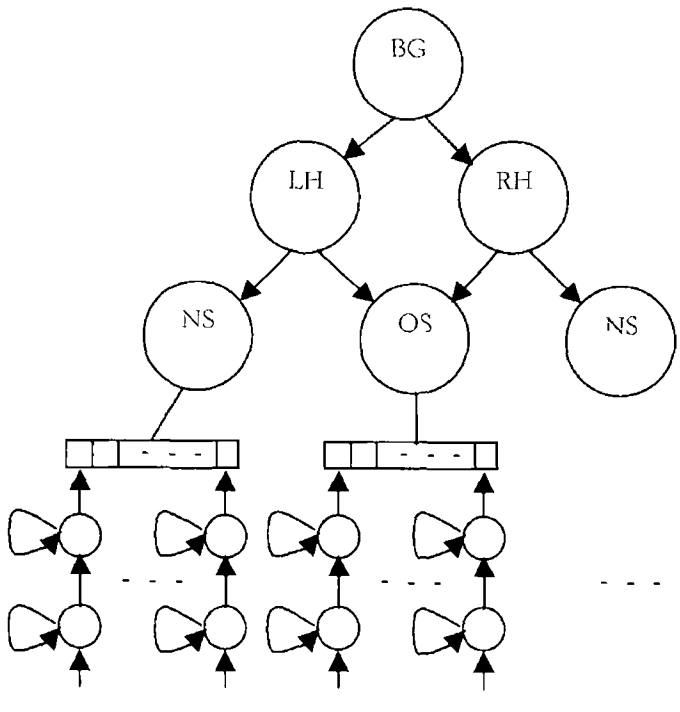


Figure 8 11 The abstracted Bayesian network for the recognition of bimanual movements

8 2 4 Partial Discrete Hidden Markov Models for Partial Gesture Recognition

In order to recognise the whole movement we recognise the partial gestures of each segment separately For this, we construct an eigenspace for each hand based on the results of Chapter 6 A separate eigenspace is created, also, for the occlusion segments These eigenspaces are made by the movements in the training set As in Chapter 6, by projecting all the images of one hand into its own eigenspace a cloud of points is created A set of codewords is extracted for each eigenspace using Vector Quantisation Therefore, by projecting a segment of a gesture into the corresponding eigenspace a sequence of codewords is extracted For each hand in a non-occlusion segment a 2-state left-to-right Discrete Hidden Markov Model (see Figure 8 12) is constructed Due to the fact that a partial movement of a hand in a segment is normally a short movement a 2-state DHMM is suitable to capture the beginning and end of the movement Every segment of a gesture has its individual DHMMs Thus, for every gesture in the vocabulary of bimanual movements seven DHMMs are constructed, two for the beginning segments for the two hands, one for the occlusion segments, two for the middle segments, and two for the ending segments By using the extracted sequence of codewords the DHMM of each hand in different segments is trained The DHMMs of the occlusion segments are trained by the extracted sequence of codewords of the projected images into the corresponding eigenspace

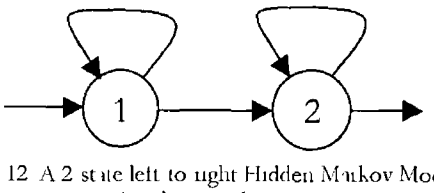


Figure 8 12 A 2 state left to right Hidden Markov Model is used for the partial gestures

For example, for a vocabulary of 10 bimanual movements 70 DHMMs are created and trained In the recognition phase the same procedure is done A given gesture is segmented Images of each segment are projected into the corresponding eigenspace and the sequence of codewords is extracted By employing the trained DHMMs the partial gesture of each hand presented in a segment can be recognised However, we use the DHMMs to calculate the likelihoods that a given partial gesture is each of the corresponding partial gestures in the vocabulary A normalised vector of the likelihoods for a given partial gesture in a segment is passed to one of the evidence nodes in the Bayesian network of Figure 8 11 For example, the second scalar in the NS vector of the left hand is the likelihood that

- In a beginning segment the given partial gesture is the gesture number 2 in the vocabulary, calculated by the partial DHMM of the beginning segment of the left hand of this gesture

- In a middle segment the given partial gesture is the gesture number 2 in the vocabulary, calculated by the partial DHMM of the middle segment of the left hand of this gesture

and so on

The occlusion vector, which is fed by the likelihoods of the DHMMs of the occlusion segments, is a shared message communicated to the LH and RH nodes as evidences for the two hands

The network looks *loopy* (containing a loop) The nodes of BG, LH, OS, and RH form a loop Therefore, the network does not seem to be singly connected and a message may circulate indefinitely However, the node OS is an evidence node Referring to the propagation rules the evidence nodes do not receive messages and they always transmit the same vector Therefore, the NS and OS nodes are not updated by the messages of the LH and RH nodes In fact, the LH and RH nodes do not send messages to the evidence nodes Therefore, although the network looks like a loopy network, the occlusion node of OS cuts the loop and no message can circulate in the loop This enables us to use the belief propagation rules of singly connected networks in this network In the next sections, however, we change the structure of the network to a loopy one and will assess it in recognising the gestures

The procedure of recognising partial gestures and fusing the results by the proposed Bayesian network in order to recognise a bimanual movement is summarised in the following algorithm,

The Algorithm for Bimanual Movement Recognition

1 *A bimanual gesture is segmented by the tracking algorithm*

2 *The beginning segment*
  2 1 For every hand the beginning segment is projected into the eigenspace of the corresponding hand
  2 2 The sequence of codewords is extracted for each hand
  2 3 By employing the DHMMs of the beginning segment of each hand the vector of likelihoods is calculated and normalised

2 4 The vectors of likelihoods are passed into the corresponding NS nodes while the vector of occlusion node is set to a vector of all 1s
2 5 The nodes' beliefs are updated

3 An occlusion segment
3 1 The image sequence of the segment is projected into the eigenspace of the occlusion segments
3 2 A sequence of codewords is extracted
3 3 The vector of likelihoods is calculated and normalised by using the corresponding DHMMs
3 4 The vector is passed into the OS node
3 5 The nodes' beliefs are updated

4 A middle segment
4 1 For every hand the corresponding image sequence is projected into the corresponding eigenspace
4 2 The sequences of codewords are extracted
4 3 The vectors of likelihoods are calculated and normalised by using the corresponding DHMMs
4 4 The vectors of likelihoods are passed to the corresponding NS nodes
4 5 The nodes' belief are updated

5 While there are more occlusion and middle segments the parts 3 and 4 of the algorithm are repeated

6 The ending segment

6 1 For every hand the image sequence is projected into the corresponding eigenspace
6 2 The sequence of codewords are extracted
6 3 The vectors of likelihoods are calculated and normalised by using the DHMMs of the ending segments
6 4 The vectors are passed to the corresponding NS nodes
6 5 The nodes' beliefs are updated

7 The gesture with the highest probability in the local belief of the root node is the best match

8 2 5 Experimental Results

15 bimanual movements were created as if the hands were doing regular daily movements like clapping, signing Wednesday in the British Sign Language, knotting a string, turning over the leaves of a book, and some movements from sign language A few sample frames of each movement are shown in Figure 8 13

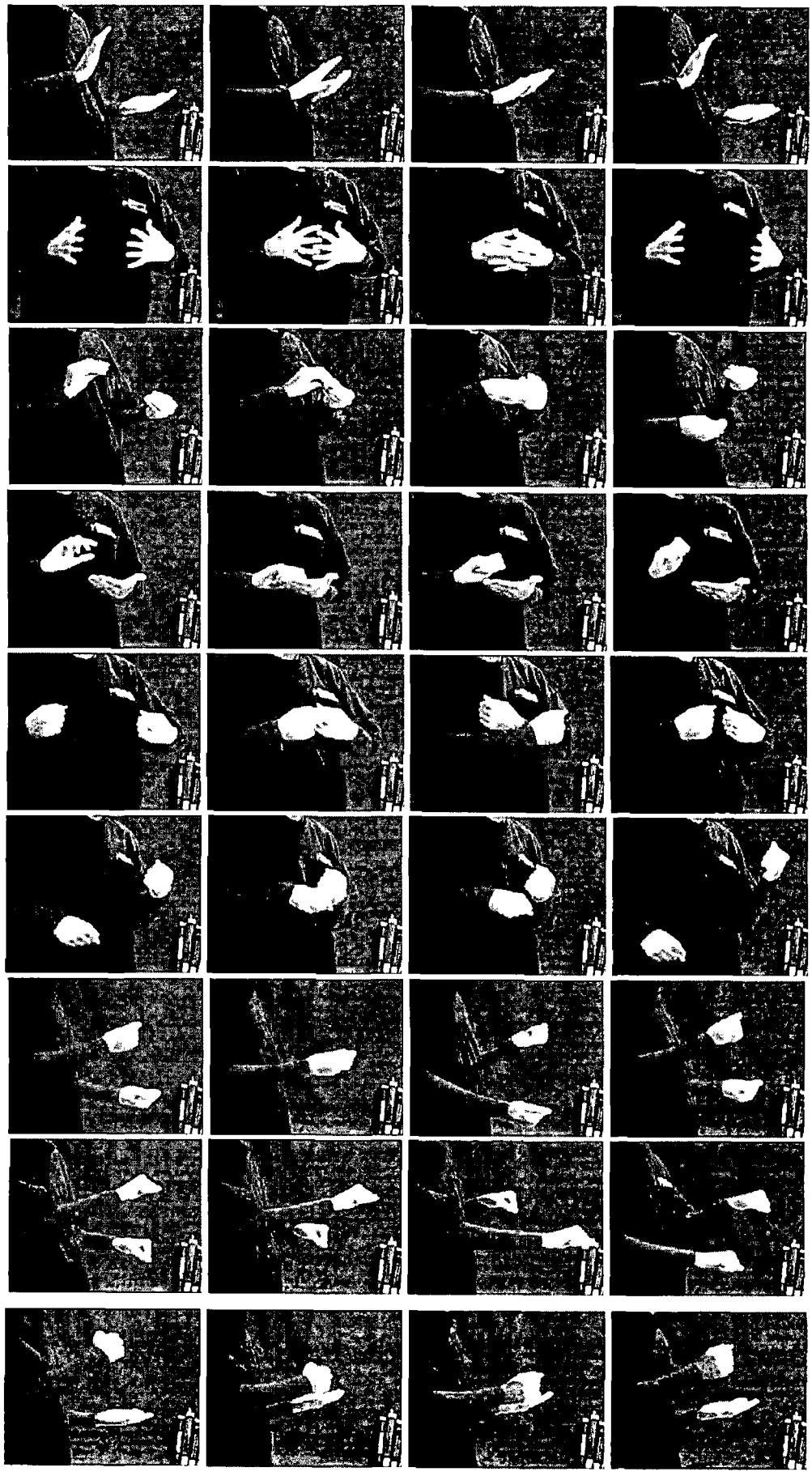


Figure 8 13 Four sample frames of each bimanual movement

Figure 8 13 (contd) Four sample frames of each bimanual movement

For every movement we captured 10 samples Half of the samples (75) were treated as the training set and the rest as the test set By using Principal Component Analysis the eigenspaces were formed By applying Vector Quantisation 128 codewords for each eigenspace were extracted By this number, each codeword represents approximately 100 data points in the training set[2] A two states left-to-right Discrete Hidden Markov Model was

[2] In Chapter 6 it has been shown that based on the variation in data, similarities and processing speed this rate is a proper choice

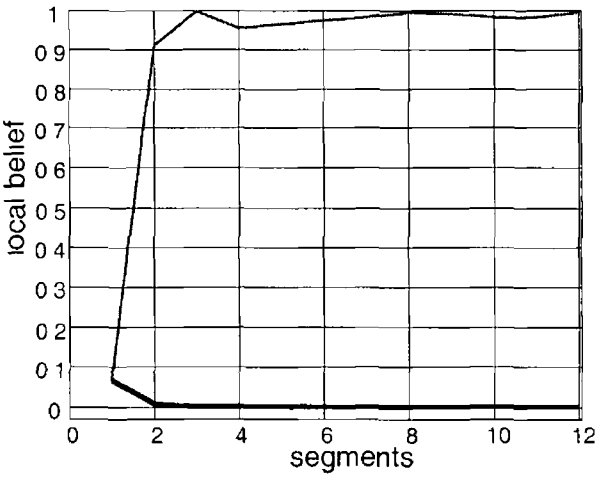
created for the segments of the hand gestures The DHMM of every segment of a gesture was trained by the 5 samples in the training set

In the recognition phase the samples of the test set were used Here we present the results of some experiments on the gestures in the test set The belief changes of the LH, RH, and BG nodes of the network for a clapping gesture are shown in Figure 8 14 In this figure, the thin lines represent the value of the belief vector corresponding to the correct gesture The other lines, which appear as thick solid lines represent the other elements of the belief vector At the beginning the network is initialised Due to the fact that all the gestures in the vocabulary are assumed to have equal prior probabilities the prior probability of the root node is assumed to be a 15-vector with equal values which are 0 0667 Also the initialisation of the network is done by setting all the evidences to 1 The transition matrices were set to,

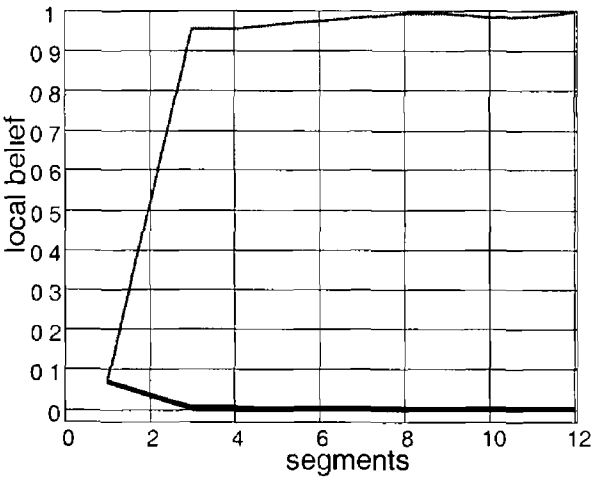
$$m_{i,j} = \begin{cases} 0\,8 & i = j \\ 0\,014286 & i \neq j \end{cases}$$

The belief of the nodes starts from the initial equilibrium This initial point is presented as the initial segment in the graphs of Figure 8 14 By processing the beginning segment of the gesture the beliefs of the nodes are updated and presented as the second segment in the graphs The graphs show that the gesture has been recognised rapidly in the beginning segments and this result has been preserved throughout the rest of gesture

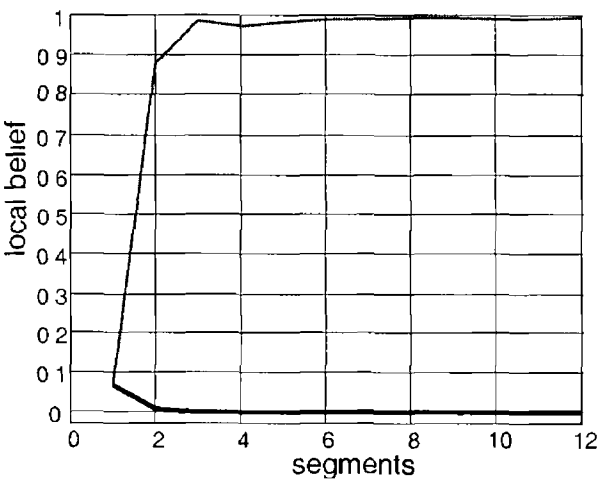
The network was employed to recognise all the movements in the test set By this algorithm 74 out of 75 movements in the test set were recognised correctly The graphs of the belief change of the only gesture that was not recognised correctly are sketched in Figure 8 15 At the beginning the gesture was correctly recognised From one point onward the beliefs are changed so that the gesture was recognised differently Our investigation shows that from this point the DHMMs have resulted in different likelihoods and this result has been preserved throughout the rest of the movement Although, in the mis-recognised part of the movement the beliefs are not as confident as the correctly recognised part, recognition is performed based on the highest probability Therefore, it is concluded that the recognised gesture is the gesture with highest probability in the local belief of the root node



(a)

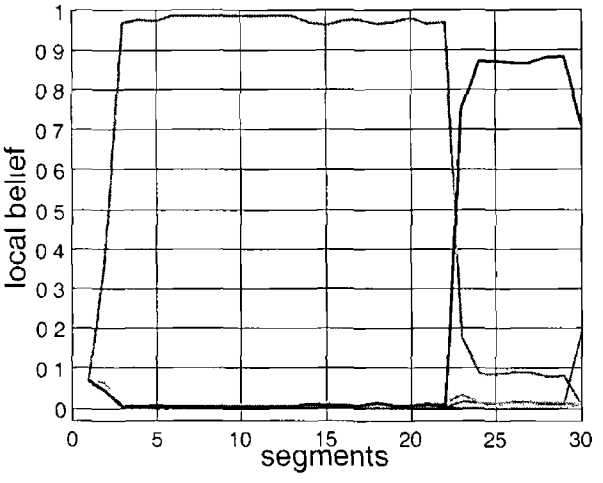


(b)

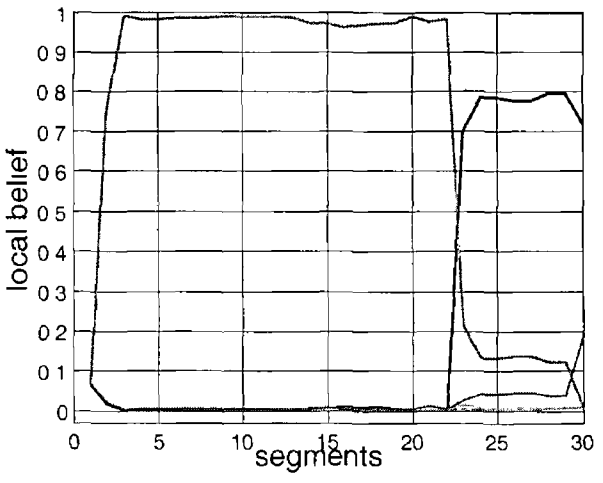


(c)

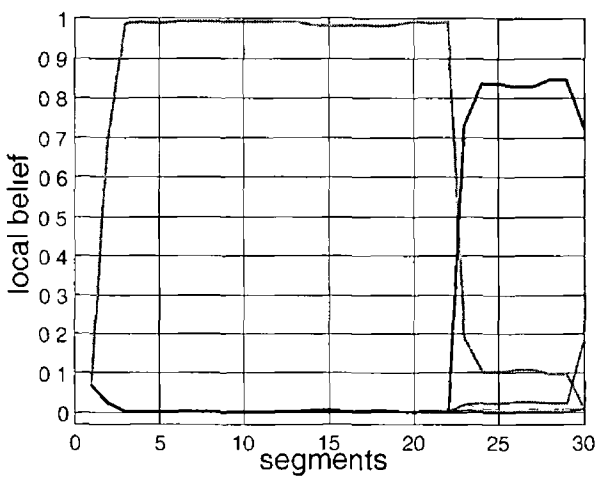
Figure 8 14 The graphs of belief changes in the (a) LH (b) RH, and (c) BG nodes for the clapping gesture



(a)



(b)



(c)

Figure 8 15 The belief changes of the gesture recognised wrongly (a) LH, (b) RH, and (c) BG nodes

8 3 Recognition of Concatenated Periodic Bimanual Movements

Many bimanual movements can be periodic in essence Clapping and drumming are some examples In the environments where the bimanual movements are used as a communication method, e g Virtual Reality, recognising concatenated periodic movements is crucial [1]

We use the Bayesian network of the last section in order to recognise this type of movement The important points in such a process are correct recognition of the movements over the whole repetition periods and exact detection of gesture changes when different movements are concatenated We use the trained models of the last section

8 3 1 Recognition by the Original Bayesian Network

Using the proposed Bayesian network, we did many experiments on concatenated periodic movements The results of one of them are presented here

Three bimanual gestures were performed consecutively, each of which was repeated dozens of times From the 15 movements, first gesture number 3 was repeated 5 times It was followed by gesture number 2 repeated 30 times and followed by gesture number 5 repeated 41 times Therefore, the first gesture is divided into 11 segments, the second gesture into 61 segments, and the last one into 83 segments Given the fact that the first segment in the graph of local beliefs represents the belief of initialisation, the first gesture transition should appear in the 13^{th} segment and the second transition in the 74^{th} segment The local belief of the root node is plotted in Figure 8 16 The gestures are correctly recognised most of the time Also, the gesture transitions are detected properly However, it can be seen, particularly in the graph of the second gesture, that the belief is not very stable and it varies such that at some points it falls below the graph of other gestures This happens when the partial gestures of one or two hands are recognised wrongly

Although the confusions can be treated as temporary spikes, we may come to a conclusion that the gesture has changed at some points In fact the belief in other gestures is higher than the second gesture at those points that supports the transition hypothesis

[1] Many Virtual Reality applications have been introduced in industry where bimanual movements are the main source of communication Some interesting applications are demonstrated by a Canadian company called Vivid Group at www.vividgroup.com In these applications users do some basic periodic bimanual gestures in order to control a spacecraft shoot the enemy troops etc

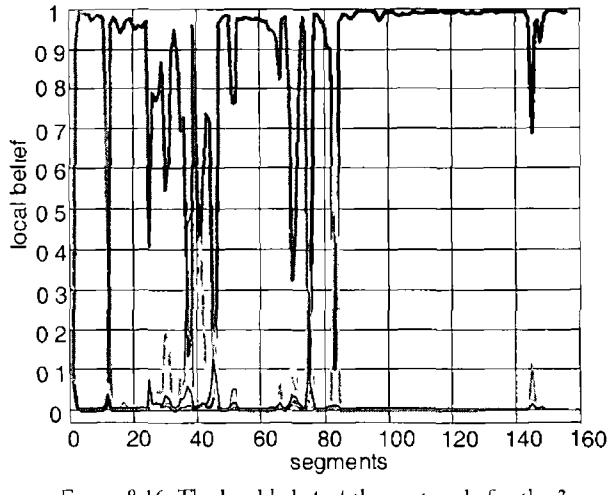


Figure 8 16 The local belief of the root node for the 3 concatenated movements

8 3 2 A Modified Belief Propagation

In order to avoid these confusing spikes we make a slight change in the belief propagation algorithm If the belief of the root node is somehow memorised so that the temporary confusing evidence cannot change the belief easily the confusing spikes are eliminated

For this, we add a memory to the root node of the network This is done by replacing the prior probability of the root node with the current local belief of the node In other words, the current belief of the root node is treated as the prior probability of the node in the next step When a hypothesis (that one of the gestures in the vocabulary is the correct gesture) is strengthened multiple times by the messages received from the DHMMs, many strong pieces of evidence are needed to change this belief

Although, by applying this modification the local beliefs are no longer representing the correct posterior marginals, the result would be useful for recognition Figure 8 17 shows the result of this modification on the movement mentioned in the last subsection Obviously, the confusing spikes are eliminated and the gestures are recognised correctly However, replacing the prior probability of the root node with the node belief can cause numerical underflows[3] while a gesture is repeated several times This will result in extreme delays in detecting gesture transitions (see Figure 8 17)

[3] By numerical underflow we mean both the IEEE definition of numerical underflow and extremely small numbers

In this figure, the first gesture transition point is detected after the 20th segment while the actual transition point is in the 13th segment The second transition is detected a little after the 120th segment but the actual transition point is in the 74th segment

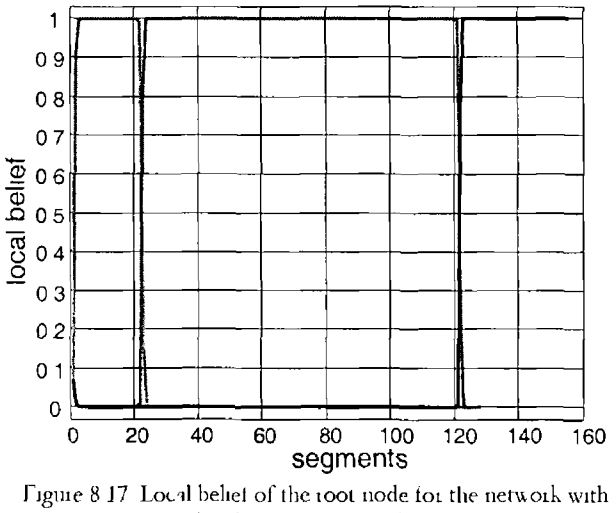


Figure 8 17 Local belief of the root node for the network with memory for the 3 concatenated movements

To avoid the numerical underflows and confusing spikes we restrict the memory By this restriction the prior probabilities of the root node cannot fall below a certain limit The results of the network with short-term memory with the limit of 10^{-3} are presented in Figure 8 18

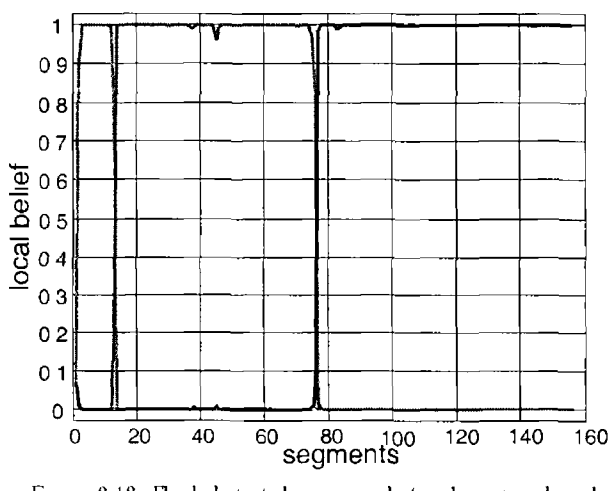


Figure 8 18 The belief of the root node for the network with short term memory for the 3 concatenated movements

In this figure the confusing spikes are avoided while delays in detecting the transition points have reduced to a few units (segments) The first and second transitions were detected one segment and two segments respectively after the actual transition points

8 4 A Loopy Network for Recognition

In Section 8 2 4 we mentioned that the network can contain a loop if the node of OS is not an evidence node Pearl in [Pearl 1988] states that

> "When loops are present, the network is no longer singly connected and local propagation schemes will invariably run into trouble If we ignore the existence of loops and permit the nodes to continue communicating with each other as if the network were singly connected, messages may circulate indefinitely around the loops and the process may not converge to a stable equilibrium "

Murphy et al [Murphy 1999] have empirically shown that Pearl's belief propagation algorithm works as an approximate inference scheme in a wide range of medical applications containing non-singly connected networks or the networks with loop In this section we investigate a loopy network to see whether it converges to approximate probabilities in our problem, and under what circumstances a loopy network converges rapidly so that we can expect little errors in the probabilities Also, we test the loopy network to see if it has any advantage over the singly connected network from a recognition rate point of view

8 4 1 Belief Propagation in the Loopy Network

We change the structure of the original Bayesian network of Figure 8 11 so that the node OS is replaced by a sub-tree of two nodes rooted at a non-evidence node making the network loopy (see Figure 8 19)

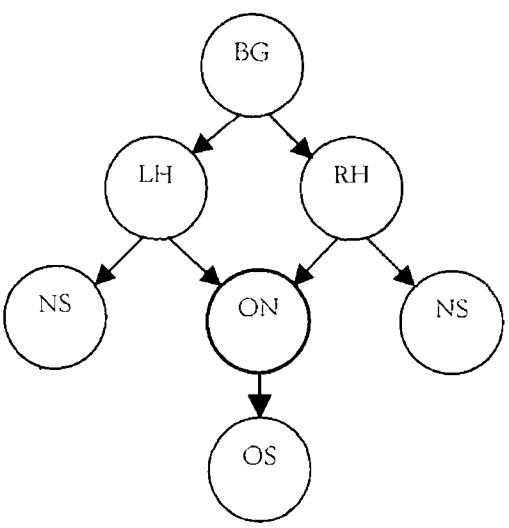


Figure 8 19 The Bayesian network containing a loop

In this network, the OS is an evidence node while ON is not The local belief of ON is updated by the messages received from OS, LH, and RH Messages can circulate in this loopy network indefinitely While the messages are circulating the local belief of every node is updated regularly

We employed this network to recognise the bimanual gestures in the test set It was observed that the algorithm converges to approximate posterior marginals on the correct side of the decision line Figure 8 20 shows the result of the same clapping gesture as Figure 8 14 using the loopy network

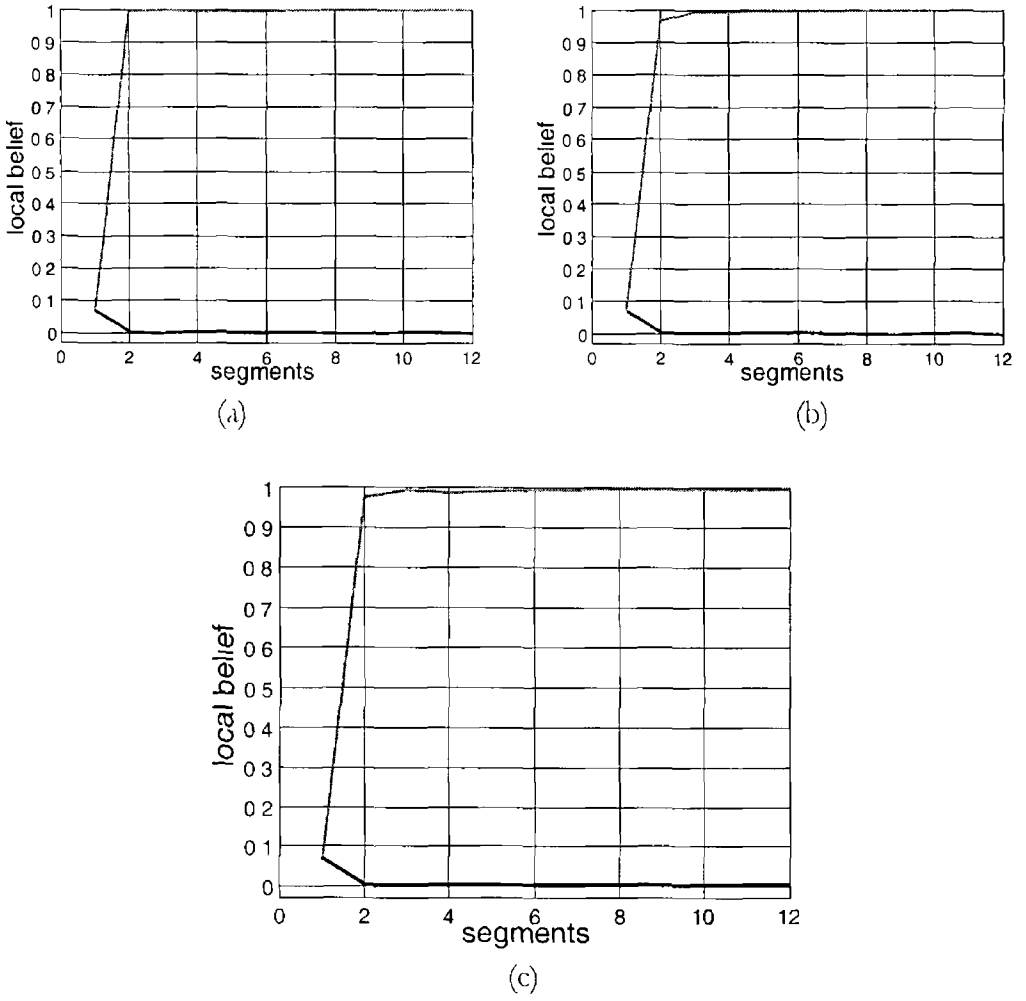


Figure 8 20 The local belief of the (a) LH (b) RH and (c) BG node of the loopy network for the clapping gesture

In order to investigate the belief convergence of the algorithm we keep tracking the belief changes of the nodes in the network As an example the belief change of the root node for the above clapping gesture is plotted in Figure 8 21 In Figure 8 21(a) the belief convergence of the root node as a function of iteration for the beginning segment of the gesture is shown

In this experiment the convergence limit has been set to 10^{-14} which means that the network stop propagating messages when the total change of the belief vector of root node is less than the limit We have chosen this value in this example to better show the convergence of the algorithm Normally, a limit of 10^{-3} suffices Figures 8 21 (b) to (d) are the convergence of the root node belief for an occlusion, a middle and the ending segments respectively As it is shown in these graphs, the algorithm converges rapidly in all the segments of the gesture

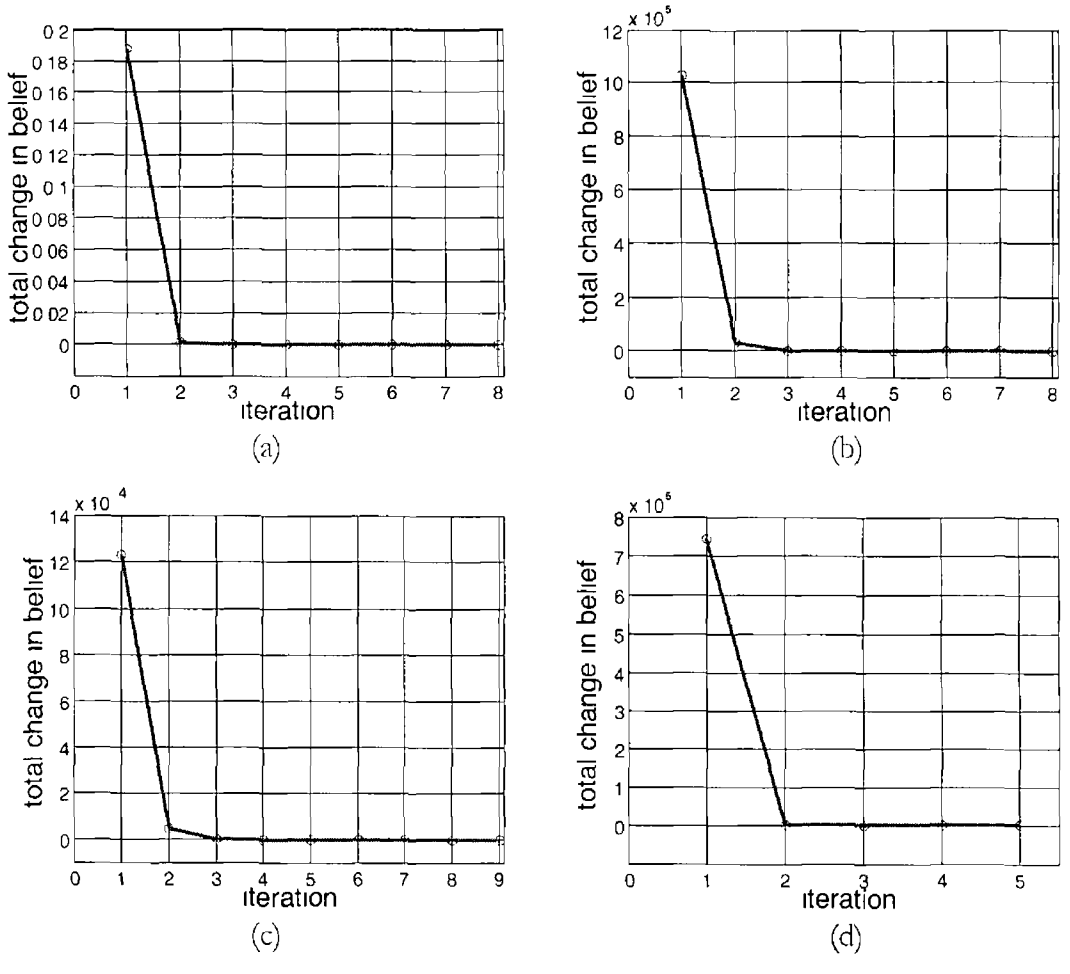


Figure 8 21 Belief convergence of the root node in the loopy network at the (a) beginning (b) occlusion (c) middle and (d) ending segments for the clapping gesture

In order to show the convergence rate of the algorithm the *log* plots of the belief changes are plotted in Figure 8 22

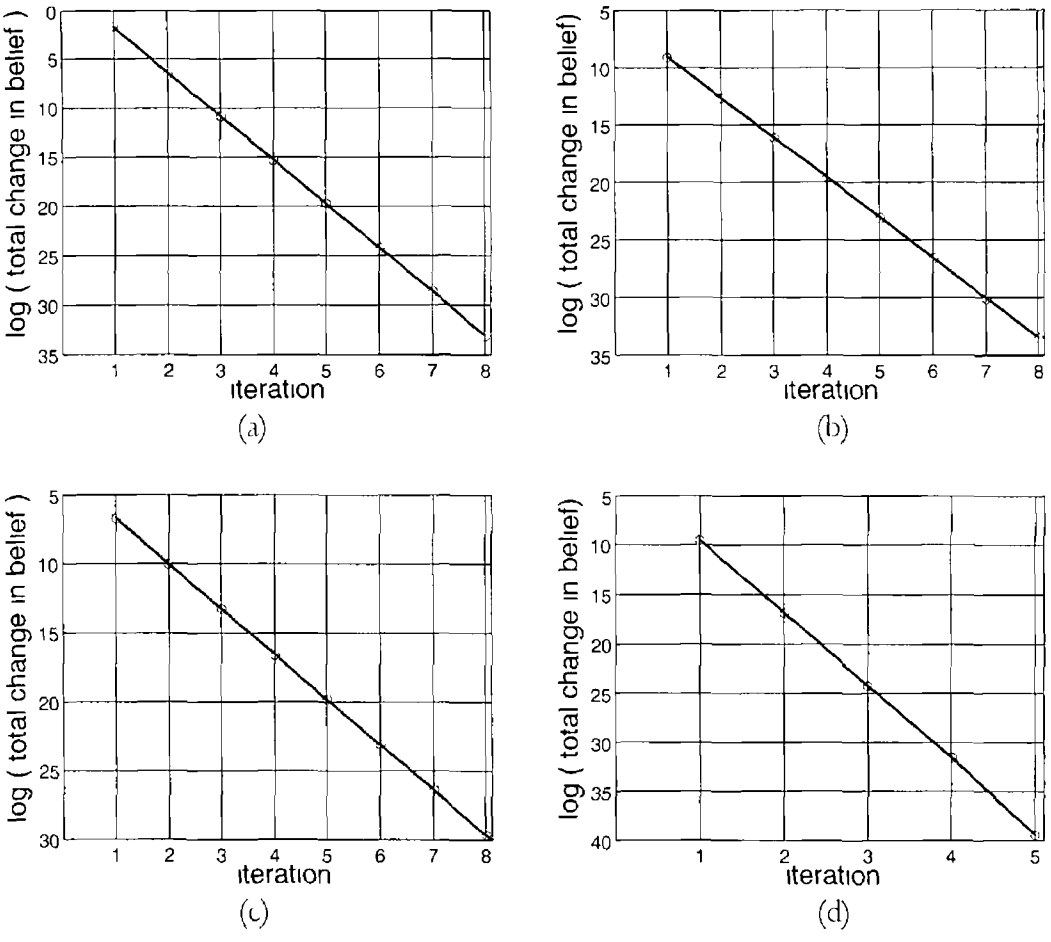


Figure 8 22 Log of belief changes for (a) beginning (b) occlusion (c) middle and (d) ending segments for the clipping gesture

The *log*-plots show that the error rate decreases geometrically as the iteration increases The linear relation and the number of iterations in the *log*-plots demonstrate the rapid convergence of the algorithm as is depicted in Figure 8 21 This rapid convergence, then, can predict that although the calculated beliefs are not correct posterior marginals the error will be small

8 4 2 Why Does the Loopy Propagation Converge in our Network

Pearl [Pearl 1988] has stated that in order for a message passing scheme to be successful *double counting* must be avoided In a singly connected network double counting is avoided by the conventional belief propagation algorithm But in the loopy networks double counting cannot be avoided as the messages are circulating around the network Then why should the loopy propagation ever converge? Weiss in [Weiss 1997] states that

"... although the evidence is double counted, all evidences are double counted in equal amounts."

The double counting in our loopy network can be formalised by an unwrapping technique [Weiss 1999], [Weiss 2000]. In order to unwrap the network first we should convert the Bayesian network to a pairwise Markov net. For any node that has multiple parents a *compound node* is created into which the common parents are clustered. Then the sub-tree rooted at the original node is replaced by a sub-tree rooted at the compound node with the original node as its child. The pairwise Markov net for the Bayesian network of Figure 8.19 is shown in Figure 8.23. In order to keep the probability distributions identical to the original network the pairwise potentials of the Markov net are the conditional probabilities of children given parents, except for the potentials between the compound node and its parents. These potentials are the identity matrix, which elements are set to one if the node has a consistent estimate of the parent node and zero otherwise [Weiss 2000]. The unwrapping of the Markov net can be done as following.

For the node BG in our loopy network at iteration time t we construct an unwrapped tree by setting BG to be the root node and repeating the following routine t times [Weiss 2000],

- Find all leafs of the tree (nodes without any children)
- For each leaf, find k nodes in the loopy graph that neighbour the node corresponding to this leaf
- Add k-1 nodes as children to each leaf, corresponding to all neighbours except the parent node

The transition matrices are identical to those in the loopy network. For our loopy Markov net the unwrapped tree is shown in Figure 8.24 for three iterations.

The unwrapped network is constructed so that the messages received at node BG after t iterations are identical to those that would be received at the loopy network. The unwrapped network is singly connected. Therefore, it is guaranteed that the belief propagation algorithm gives correct beliefs at time t. But every iteration of the loopy propagation gives the correct belief for a different problem. Then, why should this scheme ever converge? The answer is that the unwrapped network at time $t+1$ is the unwrapped network at time t plus an additional finite number of nodes at the boundary. Therefore, the loopy propagation will converge when adding boundary nodes does not change the posterior probability of the BG node in the centre of the unwrapped network.

An important point is how fast the loopy propagation converges. In the subsequent section we formalise the circumstances that the algorithm converges rapidly in the loopy network for bimanual gesture recognition.

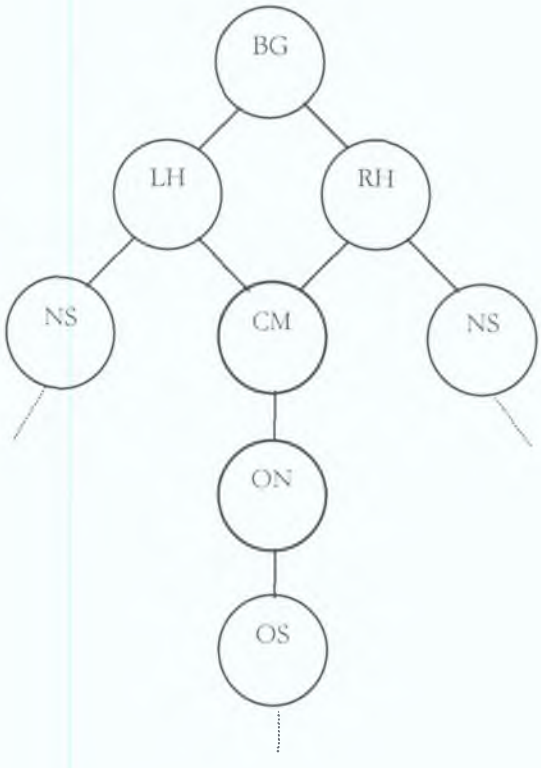


Figure 8.23. The pairwise Markov net of the loopy Bayesian network

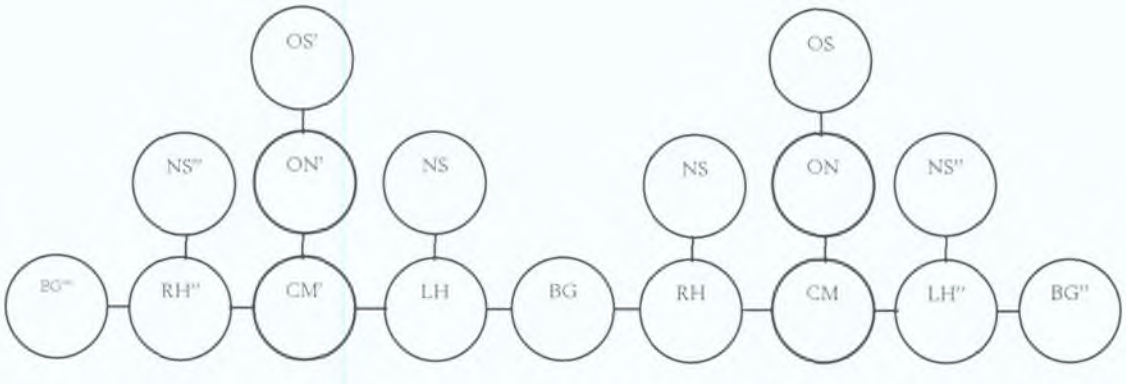


Figure 8.24. The unwrapped network of the loopy Markov net

8.4.3 Convergence Speed in the Loopy Network

In order to find the general principles for the convergence speed of the algorithm we consider a loopy network with N unobserved (hidden) nodes $U_1, U_2, \ldots, U_N$ and N observed (evidence) nodes $O_1, O_2, \ldots, O_N$ each of which associated with a hidden node depicted in Figure 8.25.

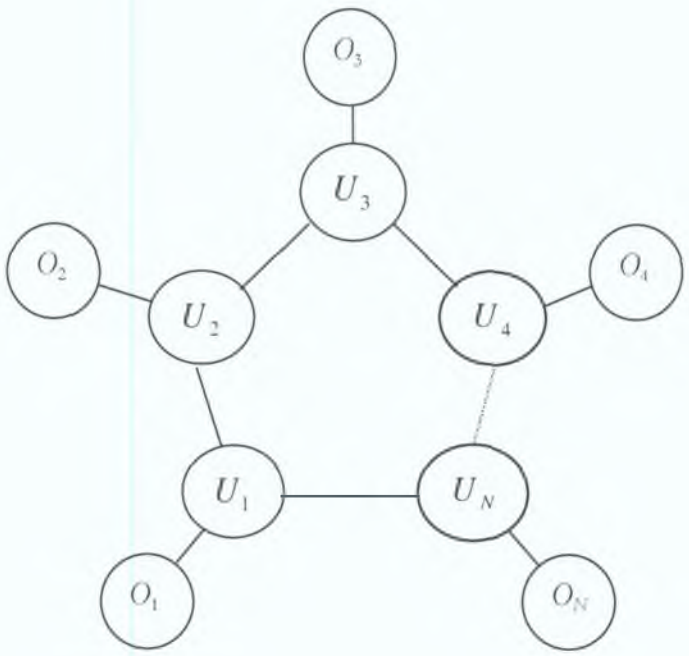


Figure 8.25. A single loop network with N unobserved nodes and N observed (evidence) nodes each of which associated with an unobserved node

Here, we use the same notation as [Weiss 2000] for the mathematical expressions. Based on the basic rules of message passing in the singly connected networks, the message node U_N sends to U_1 is given by,

$$\vec{v}_{U_N U_1} \leftarrow \alpha M_{U_N U_1} (\vec{v}_{O_N U_N} \otimes \vec{v}_{U_{N-1} U_N}) \tag{8.11}$$

where $\vec{z} = \vec{x} \otimes \vec{y} \leftrightarrow \vec{z}(i) = \vec{x}(i)\vec{y}(i)$, $\vec{v}_{O_N U_N}$ is the message the observable node O_N sends to U_N, $\vec{v}_{U_{N-1} U_N}$ is the message node U_{N-1} sends to U_N, $M_{U_N U_1}$ is the transition matrix corresponding to the link from U_N to U_1, and α is the normalising factor.

Similarly the message that node U_{N-1} sends to U_N is

$$\vec{v}_{U_{N-1}U_N} \leftarrow \alpha M_{U_{N-1}U_N}(\vec{v}_{O_{N-1}U_{N-1}} \otimes \vec{v}_{U_{N-2}U_{N-1}}) \tag{8.12}$$

Continuing around the loop, we return to U_1 by,

$$\vec{v}_{U_1U_2} \leftarrow \alpha M_{U_1U_2}(\vec{v}_{O_1U_1} \otimes \vec{v}_{U_NU_1}) \tag{8.13}$$

Thus, the message U_N sends to U_1 depends on the messages U_N sends to U_1 at N steps before,

$$\vec{v}_{U_1U_2}^{(t+N)} \leftarrow \alpha C_{N1}\vec{v}_{U_1U_2}^{(t)} \tag{8.14}$$

where the matrix C_{N1} is defined as following,

$$C_{N1} = M_{U_NU_{N-1}}D_N M_{U_{N-1}U_{N-2}}D_{N-1}\cdots M_{U_1U_2}D_1 \tag{8.15}$$

and the matrix D_i is defines as a diagonal matrix whose elements are the constant message sent from observed node O_i to U_i. Using the matrix C_{N1}, Weiss [Weiss 2000] has proven that in a single loop network,

1. $\vec{v}_{U_NU_1}$ converges to the principal eigenvector of C_{N1}.
2. $\vec{v}_{U_2U_1}$ converges to the principal eigenvector of $D_1^{-1}C_{N1}^T D_1$.
3. The convergence rate of the messages is governed by the ratio of the largest eigenvalue of C_{N1} to the second-largest eigenvalue

...

He states that when the ratio between the second-largest eigenvalue and the largest one is small, loopy belief propagation converges rapidly and furthermore the approximation error in calculating the correct posterior marginals is small i.e.,

$$\frac{\lambda_{\text{sec}}}{\lambda_{\max}} << 1 \tag{8.16}$$

Now we formalise the circumstances for which the above unequality holds.

Obviously the second-largest to largest eigenvalue (SLLE) ratio is related to the evidences and the transition matrices. We assume that each node in the network represents a p-valued

variable. Therefore, each message is a vector with length p and $M_{U_iU_j}$ is a pxp matrix. We define r_i to be the sum of the elements on the i^{th} row of C_{N1} excluding the diagonal element,

$$r_i = \sum_{\substack{j=1 \\ j \neq i}}^{p} |c_{ij}| \tag{8.17}$$

The i^{th} Gerschgorin disk is defined as the set of points on the imaginary-real plane whose distances to c_{ii} are at most r_i. In other words, G_i is the set of all complex numbers z such that,

$$G_i = \left\{ z \in Z : |z - c_{ii}| \leq r_i = \sum_{\substack{j=1 \\ j \neq i}}^{p} |c_{ij}| \right\} \tag{8.18}$$

Based on the Gerschgorin theorem [Hager 1988] every eigenvalue λ of C_{N1},

$$\lambda \in \bigcup_{i=1}^{p} G_i \tag{8.19}$$

Also, if m Gerschgorin disks form a connected region of R disconnected from the other disks, then there are exactly m eigenvalues in the region. In order to find the relationship between the eigenvalues of C_{N1} and the evidences, we first assume that $M_{U_iU_j}$ is a stochastic matrix close to the identity matrix. By closeness we mean that the diagonal elements of matrix M are much larger than the off-diagonal elements. In this case, by multiplying M by the evidence matrix D, the diagonal elements of the product have approximately the same values as the evidence D, while the off-diagonal terms are kept small, given D is normalised. Specifically,

$$M : \begin{cases} m_{ij} \approx 1 - \varepsilon & i = j \\ m_{ij} \approx \dfrac{\varepsilon}{p-1} & i \neq j \end{cases} \tag{8.20}$$

where ε is a small value, then

$$Q = M \cdot D$$

$$Q: \begin{cases} q_{ij} \approx d_{ij} & i = j \\ q_{ij} << 1 & i \neq j \end{cases} \tag{8.21}$$

The Equations 8.20 do not necessarily mean that all the diagonal or off-diagonal elements of M should be equal.

The eigenvalues of Q are inside the Gerschgorin disks with d_{jj} as the centre, and the sum of off-diagonal elements of row j as the radius. Due to the fact that the off-diagonal elements are very small, the radius of the Gerschgorin disks will be small too. Therefore, the eigenvalues are mainly positioned by the disk centres. Furthermore, the disk centres are strongly under the influence of evidence matrix D. We can conclude that the eigenvalues of Q are strongly related to the diagonal matrix D whose elements are the evidence vector E,

$$\Lambda \propto E \tag{8.22}$$

where Λ is the set of eigenvalues.

$$\frac{\lambda_{\text{sec}}}{\lambda_{\max}} \propto \frac{e_{\text{sec}}}{e_{\max}} \tag{8.23}$$

where λ_{sec} and e_{sec} stand for the second largest eigenvalue and the second largest evidence respectively. Therefore, if the ratio of the second-largest value to the largest value of evidence E is small the SLLE ratio will be small too.

In matrix C_{N1} if all the transition matrices along the loop from U_1 to U_N are close to the identity matrix, the ratio of the eigenvalues of C_{N1} is related to the evidences E_1 to E_N,

$$\frac{\lambda_{\text{sec}}}{\lambda_{\max}} \propto \frac{e_{\text{sec}}^{loop}}{e_{\max}^{loop}} \tag{8.24}$$

where e_{sec}^{loop} and $e_{\max}^{loop}$ are the second largest and the largest elements in the product of all the evidence vectors along the loop.

Therefore, if the final value of the term on the right side of Relation 8 24 is very small the belief propagation algorithm converges rapidly In other words, if the product of all the evidences along the loop support one of the hypotheses strongly, the loopy belief propagation converges fast

However, if the diagonal elements of the transition matrices around the loop are not much larger than the off-diagonal elements, then Relation 8 24 no longer holds In fact the ratio of the eigenvalues of C_{N1} is dominated by the ratio of the eigenvalues of the transition matrices Essentially, the ratio of the second-largest to the largest eigenvalue of a stochastic matrix with equal elements tends to zero Therefore, the closer the transition matrices are to such a matrix, the faster the loopy propagation algorithm converges In other words, the higher the uncertainty, in the transition matrices the faster the algorithm converges to some uncertain results

8 4 4 Simulation Results

In Section 8 4 1 we showed that the loopy propagation converges rapidly for recognition of bimanual gestures In order to investigate the convergence speed of a loopy network we have done some simulations Some results are presented in the following

A network same as the one for bimanual recognition is shown in Figure 8 26

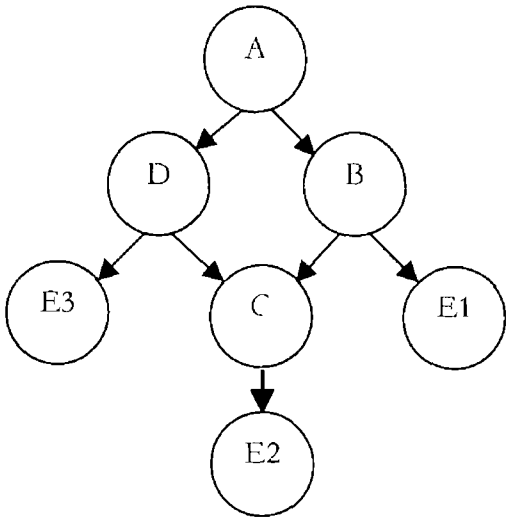


Figure 8 26 A loopy network for simulations to measure the convergence rate

The node A has no evidence node connected to it and equal prior probabilities have been assigned Nodes B, C, and D receive evidences All the nodes represent 15-valued variables We present 3 sets of evidences and transition matrices with convergence limit equal to 10^{14}

1 Small evidence ratio

$$M \begin{cases} m_{ij} \approx 0\,95 & i = j \\ m_{ij} \approx 0\,00357 & i \neq j \end{cases}$$

$$\frac{e_{\text{sec}}^{loop}}{e_{\text{max}}^{loop}} = 0\,0209$$

The belief propagation algorithm converges in 9 iterations In other words, the belief change of the root node is less than 10^{14} after 9 iterations Given the small evidence ratio the rapid convergence of the network was expected

The graphs of product of the evidences (the evidences-product vector) and the final belief at the root node are plotted in Figure 8 27 The convergence rate as a function of iteration is plotted in Figure 8 28

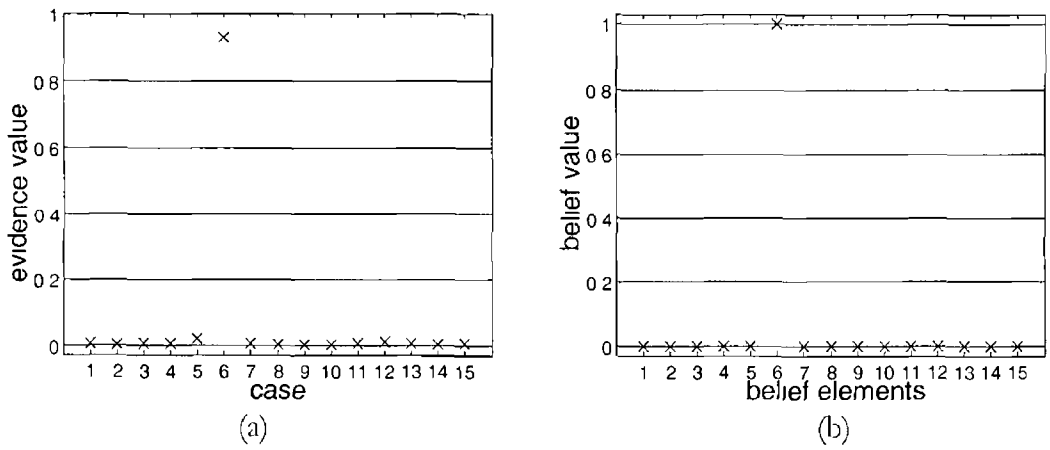
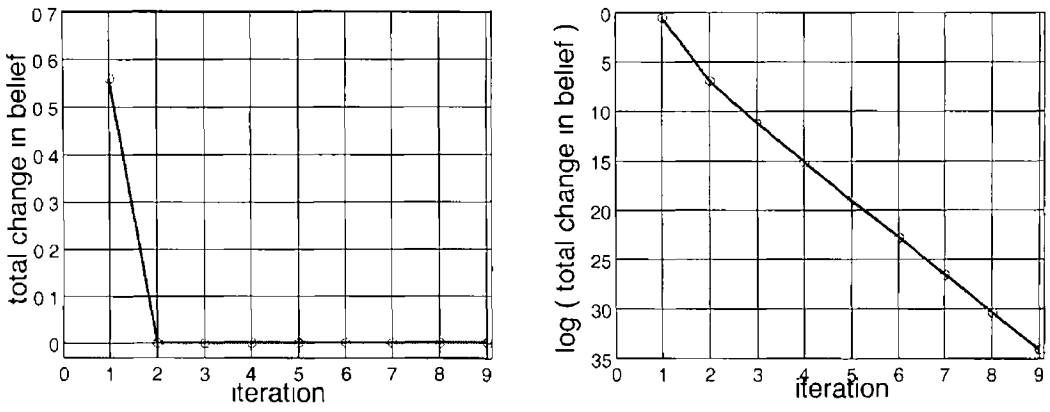


Figure 8 27 (a) The normalised product of all the evidences along the loop and (b) the final belief at the root node when the belief propagation algorithm converge



(a) (b)

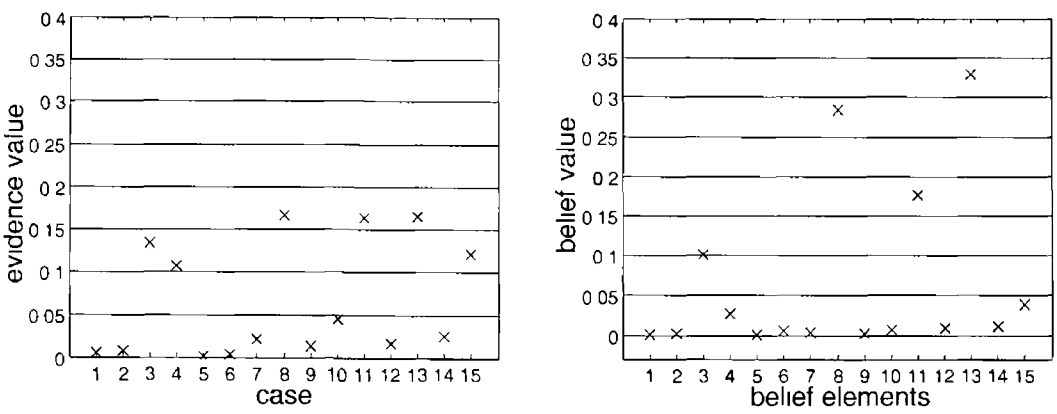
Figure 8 28 The belief propagation algorithm converges in 9 iterations in this example (a) The graph of total change in the local belief of the root node while the algorithm converges, (b) the *log*-plot of the total change in the local belief of the root node

2 Large evidence ratio

$$M \begin{cases} m_{ij} \approx 0\,95 & i = j \\ m_{ij} \approx 0\,00357 & i \neq j \end{cases}$$

$$\frac{e^{loop}_{sce}}{e^{loop}_{mix}} = 0\,9940$$

Figure 8 29 shows the graphs of the product of all the evidences along the loop (normalised) and the final belief at the root node In this example, the process converges very slowly in 95 iterations This was expected due to the fact that in this example the SLLE ratio is large The graphs of the convergence rate as a function of iteration are shown in Figure 8 30



(a) (b)

Figure 8 29 The graph of (a) the product of all the evidences along the loop (b) final belief at the root node when the belief propagation algorithm converges In this example the loopy belief propagation has increased the differences between the elements with high and low probabilities

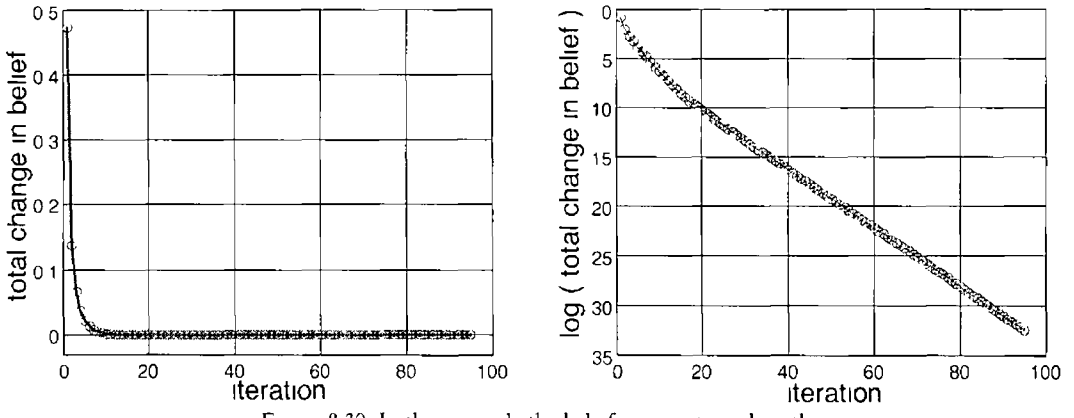


Figure 8 30 In this example the belief propagation algorithm converges very slowly due to the large SLLE ratio (a) The total change in the local belief of the root node, and (b) the *log*-plot of the total change

In the third set of data the transition matrices are not close to the identity matrix as opposed to the Examples 1 and 2 We use the second set of evidences with which the belief propagation algorithm converged slowly in 95 iterations in Example 2 However, with the new transition matrix the process converges very rapidly

3 Transition matrix not close to identity matrix

$$M \quad \begin{cases} m_{ij} \approx 0\,1 & i = j \\ m_{ij} \approx 0\,064286 & i \neq j \end{cases}$$

$$\frac{e_{\text{sec}}^{loop}}{e_{\text{max}}^{loop}} = 0\,9940$$

The belief propagation algorithm converges in 3 iterations The graph of the local belief of the root node at convergence is shown in Figure 8 31 The graph of the convergence rate as a function of iteration is plotted in Figure 8 32 As can be seen, the speed of convergence is fast but the values of the local belief at the end are not very confident

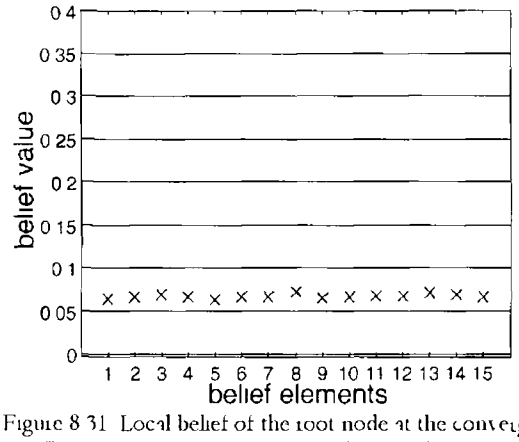


Figure 8 31 Local belief of the root node at the convergence point The probabilities are almost equal for all the elements of the belief vector In this example the differences between the elements with high and low probabilities have been reduced

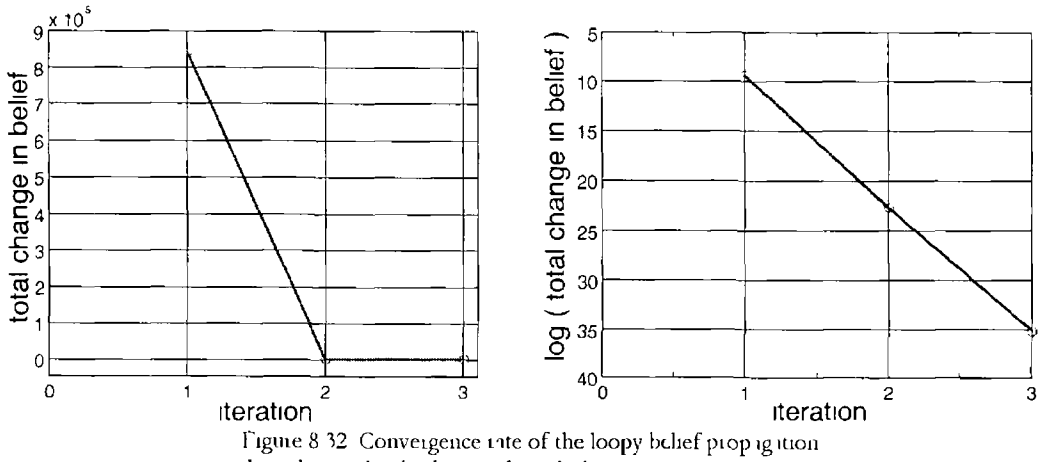


Figure 8 32 Convergence rate of the loopy belief propagation algorithm in the third example with the transition matrices not close to the identity matrix

8 5 Recognition Results of the Proposed Networks

In this section we employ the three proposed Bayesian networks to recognise the single bimanual movements and the concatenated periodic bimanual movements

8 5 1 Recognition of Single Bimanual Movements

The original Bayesian network, the network with short-term memory, and the loopy network were tested with the 75 bimanual movements of the test set All of them recognised the same 74 out of 75 gestures correctly (see Table 8 1) The loopy network converged rapidly in all the experiments The results show that the loopy network can recognise single bimanual movements as well as a singly connected network In other words, despite the conventional belief that the loopy network may run into trouble, it can recognise the single bimanual movements well

In [Brand 1997] the Coupled HMM has resulted in 94 2% recognition rate with a vocabulary of 3 T'ai Chi gestures (with no occlusion) and a small test set including one third of the examples in the training set As shown in Table 8 1, the proposed algorithm is superior to the Coupled HMMs with higher recognition rate and a larger number of gestures in the vocabulary As it was mentioned earlier the Coupled HMMs cannot deal with occlusion which is a considerable weakness

Table 8 1 Recognition results for the single bimanual movements

Bayesian Network	# Gestures	# Correctly Recognised gestures	Recognition Rate based on # gestures	# Segments	# Correctly Recognised Segments	Recognition Rate based on # Segments
Original Network	75	74	98 6%	1035	1022	98 74%
Network with Short-term Memory	75	74	98 6%	1035	1030	99 5%
Loopy Network	75	74	98 6%	1035	1021	98 64

8 5 2 Recognition of Concatenated Periodic Movements

Many concatenated periodic bimanual movements were also tested by the three proposed networks The results of recognition for the example of Section 8 3 for the three networks are presented in Figure 8 33

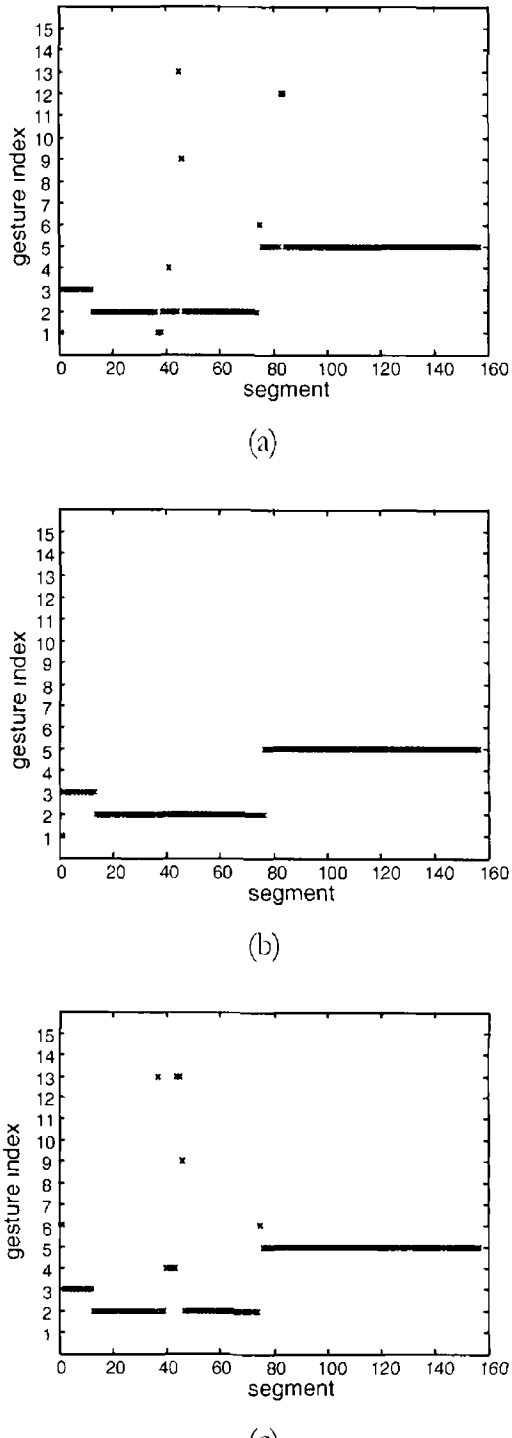


Figure 8 33 Recognition result of (a) the original network, (b) the network with short term memory, (c) the loopy network

Obviously, the network with short-term memory recognises the movements better than the others The loopy network recognises almost the same as the original network

In order to measure the recognition rate of each network in the concatenated periodic bimanual movements we tested the networks on all the concatenated periodic bimanual gestures we have in our test set The results are presented in Table 8 2

Table 8 2 Recognition rate of the networks for concatenated periodic movements with total number of segments equal to 398

Bayesian Network	Number of correctly recognised segments	Recognition rate
Original Network	336	84 4 %
Network with Short-term Memory	348	87 4 %
Loopy Network	332	83 4 %

All the networks recognise the concatenated periodic movements very well While the original and the loopy networks have quite the same recognition rate the network with short-term memory has resulted in a few percent better recognition rate

The main sources of evidences are the Discrete Hidden Markov Models, which produce the same set of evidences for all types of the networks we proposed here The better recognition rate of the network with short-term memory is due to its robustness in detecting the correct gesture transitions As we said earlier, strong evidences are needed to change the belief of the root node in the network with short-term memory Therefore, the occasional misrecognised partial gestures by the DHMMs cannot easily change the belief of the network A hypothesis that the gesture is changed should be repeated at least twice so that the network believes it In other words, the network with short-term memory tends to keep a hypothesis unchanged rather than changing it quickly Therefore, it works more robust than the other networks in the cases where the gestures have more repetitions than transitions

We should not forget that although the DHMMs are the source of evidences to the Bayesian networks, the bimanual gestures are tracked and segmented by the tracking algorithm of Chapter 7 Therefore, the presented recognition rates summarise the recognition rate of the tracking algorithm, the partial Discrete Hidden Markov Models, and the presented Bayesian

networks together In other words, the error rates presented here include the error rates of the tracking algorithm, DHMMs, and the Bayesian networks

Summary and Conclusion

A novel technique for recognition of bimanual movements was introduced A bimanual gesture is tracked and segmented by the intelligent tracking algorithm of the last chapter Then partial Discrete Hidden Markov Models are employed to recognise the partial movements of the hands in every segment A Bayesian network was introduced in order to fuse the likelihoods of the DHMMs of the hands to recognise the whole bimanual movement Our experimental results showed that the proposed network recognises the Bimanual movements very accurately

In order to recognise a set of concatenated periodic bimanual movements we changed the conventional belief propagation algorithm Since we need to stabilise the belief of the root node during the periodic movements we replaced the prior probability of the root node with the current local belief Based on this idea we demonstrated that the network's local belief at the root node is stabilised while the correct gesture transition points were almost preserved

Furthermore, a loopy Bayesian network was introduced and the loopy belief propagation was employed to recognise the segmented gestures It was shown that the loopy belief propagation algorithm converges to approximate posterior marginals on the correct side of decision line We formalised the circumstances where the loopy propagation algorithm converges rapidly We showed that there is a relationship between the evidences provided to the network and the convergence rate of the network

We employed the three proposed networks to recognise the two sets of test data The recognition rate of the three proposed Bayesian networks were estimated in recognising the single bimanual movements All the proposed networks resulted in very accurate results in recognising the single bimanual movements

The second set of results demonstrated that the three networks recognise the concatenated periodic bimanual movements well But the network with short-term memory resulted in a better recognition rate than the other networks due to its robustness against temporary confusing information

In comparison with Coupled HMM, the proposed Bayesian networks can deal with hand-hand occlusion as well as the other type of occlusion Even when something else occludes one of the hands or the hand leaves the scene the proposed Bayesian networks can deal with it because the movement of the two hands are separately recognised (as opposed to Coupled HMM) and the results are combined Therefore, even if one of the hands is not visible in a segment (other than hand-hand occlusion segment) the movement of the other hand is recognised and passed into the Bayesian network using the corresponding partial DHMMs Also for hand-hand occlusion we considered an individual recognition component Thus, the proposed Bayesian networks have great advantages over the Coupled HMM with respect to these problems

As a future work, we must consider other techniques for recognising bimanual movements and compare them with the proposed Bayesian networks For example, Fuzzy Logic and Neural Networks are two well-known inference schemes We may use these techniques as well as different structures of Hidden Markov Models to deal with occlusion and recognise bimanual movements

Chapter 9

SUMMARY, CONCLUSION, AND FUTURE WORK

Human Computer Interaction has engaged a variety of research topics in computer science and engineering. Computer Vision as a substantial issue in Machine Learning is considerably involved in Human Computer Interaction. Hand and body movement understanding has been given great attention by researchers around the world.

In this dissertation we aimed to understand bimanual movements, a problem that has not yet been addressed in the literature using computer vision, machine learning, artificial intelligence and cognitive techniques.

Recognition of bimanual movements, as a large set of movements people do in their daily life and the basis of some of the sign languages around the world, requires a wide range of techniques including single-hand shape recognition, dynamic gesture recognition, hand tracking, and recognition of synchronously performed hand movements.

9.1 Summary and Conclusion

We started by reviewing the methods and algorithms associated with static shape recognition for the recognition of non-rigid objects, partially occluded shape recognition, motion tracking, stereo imaging for occluded moving object tracking and spatio-temporal recognition of hand and body gestures. Then, we took a look at the basic attributes of a visual system. We briefly explained illumination, image formation, Charge Coupled Devices (CCD) sensors, sampling and digitisation to represent an image in a digital format.

As the preliminary part of the project we investigated a statistical method called Principal Component Analysis. This method was exploited to reduce the dimensionality of the data, which are the hand images. Using the dimensionality-abstracted data we investigated some techniques in statistical pattern recognition to identify a hand shape appearing in an image taken by a CCD camera.

A hand-computer interaction algorithm was introduced for controlling a mouse pointer in a Graphical User Interface We used the Principal Component Analysis and nearest neighbour methods to recognise the static hand shapes A state machine was introduced as a graphical entity in which the edges are defined to be events like pushing or releasing a button and moving the pointer

Once, we had identified a hand shape we were able to take a step further to analyse a dynamic hand gesture

We explored deeply the projections of the hand gestures into the feature space, or eigenspace, constructed by Principal Component Analysis The trajectory of a gesture in the eigenspace was used as the identifier of a gesture

An unsupervised clustering technique called Vector Quantisation was described in detail, which was used in many parts of the dissertation We introduced a new spatio-temporal pattern matching technique for the recognition of dynamic hand gestures Based on this model, the gestures in a vocabulary are modelled by multidimensional gaussian distributions forming a graph A new unknown gesture is also modelled by a graph Then a Graph-Matching algorithm finds the best match between the gestures in the vocabulary and the input gesture We saw that the proposed algorithm can recognise the dynamic hand gestures very well

For the recognition of bimanual gestures we had to track the hand motions We proposed a dynamic model for motion tracking This model, which was based on the Kinematic equations of motion, is a stochastic model which is used in a Kalman filtering process to track the position, velocity, and acceleration of a hand in a sequence of images In the experiments it was shown that the proposed model is able to track the hand motion correctly Particularly, the estimated velocity and the acceleration of the hand in both horizontal and vertical directions were shown to perfectly match the movement of the hand in different types of movements It was also shown that the model is able to detect hand pauses in order to detect the beginning of a gesture

Before we enter the bimanual tracking problem we explored a statistical technique called Hidden Markov Models (HMM) which has been widely used in speech and gesture recognition

We proposed a hierarchical algorithm based on the beginning hand shape of a gesture detected by the dynamic model and the Hidden Markov Models Another version of the algorithm with the HMMs replaced by the gaussian Graph-Matching was also proposed

The two algorithms were employed to recognise a large database of gestures One hundred canonical hand gestures were created Ten examples of each gesture and a total of one thousand examples were recorded for the training and the test set 500 recorded videos were used to train the algorithm and the rest were treated as the test set

In the first experiment we tested the Hidden Markov Models by bypassing the first stage of the algorithm that recognises the beginning hand shape of the gesture It was observed that the algorithm was able to recognise 89 4 % of the gestures correctly The second version of the algorithm with the gaussian Graph-Matching was employed to recognise the same test set We observed that the algorithm recognised 85 6% of the gestures

In the second experiment we employed the complete hierarchical algorithms to recognise the gestures in the test set Given that the first stage of the algorithm forwarded two groups each containing four different gestures starting with the same hand shape, the two versions of the algorithm competed closely in recognising the gestures The algorithm using HMMs recognised 95 4% of the gestures and the algorithm using gaussian Graph-Matching resulted in 95% recognition rate In this experiment the algorithm with gaussian Graph-Matching showed great superiority in speed However, despite the fast processing speed of the gaussian Graph-Matching algorithm it had some restrictions on the number of nodes in the graphs Therefore, we decided to use the Hidden Markov Models in the rest of the dissertation

Once we had a good single-hand gesture recognition technique in hand we took one step further to hand tracking in bimanual movements

In bimanual movements hands tend to be synchronised effortlessly We explored the phenomenon of bimanual coordination from a cognitive and neuroscience point of view Because of this phenomenon, temporally, when the two hands reach for different goals they start and end their movement simultaneously Spatially, we are almost unable to draw a circle with one hand and a rectangle with the other at the same time

We exploited the temporal coordination to detect positively synchronised hand movements and concurrent hand pauses in order to distinguish the hands' collisions and pauses from the

hands' passes during an occlusion period A new model was introduced based on the proposed dynamic model to model each hand individually and also the blob of the hands during occlusion

Using the individual hand models we introduced a procedure to predict hand occlusions in order to detect the exact starting point of occlusion Having a predictable occlusion period we used the model of occlusion to detect hand collision and pauses by monitoring the points at which hands velocities went to zero Based on this model we are able to track the hands correctly when the occlusion period is finished In other words, at the end of the occlusion period the algorithm is able to recognise which hand is the left hand and which one is the right hand

We presented some experimental results to demonstrate the effectiveness and robustness of the algorithm in different types of movements We also presented an example in which the independence of the tracking algorithm from the camera's angle of view and the type of movement were demonstrated Using the presented algorithm we tested the eight types of movements The algorithm was able to correctly track the hands in almost 90% of the movements

We also proposed a gaussian model in order to recognise the velocity changes of the hands during occlusion, (Appendix E) Based on this model the patterns of velocity changes during occlusion were classified and recognised by patterns of gaussian distributions In the experiments we demonstrated the patterns of velocity changes in the two classes of movements the hand passes and the hand collisions or pauses Our experiments demonstrate a good performance for the modified tracking algorithm Since the proposed algorithm was independent from the background and the actual hands velocities, we tested the algorithm in active vision applications It was demonstrated that the algorithm tracks the hands properly in these applications

In the next step, we segmented a bimanual movement into four segments using the presented tracking algorithm Each segment is associated with the movement of hands at different stages of a bimanual movement

A new Bayesian network for fusing the partial Discrete Hidden Markov Models was introduced for the recognition of bimanual movements In this network a bimanual movement is divided into the movements of the left hand and the right hand The movement

of each hand is divided into the occluded and non-occluded segments The evidence nodes of the network are fed by the partial Discrete Hidden Markov Models The HMMs are classified into seven classes, each of which is associated with a segment in the segmented bimanual movements

Using the conventional belief propagation rules we tested the algorithm on 15 different bimanual movements like clapping, knotting, and some gestures from the sign language

Given a test set including 75 examples of bimanual movements the Bayesian network demonstrated a performance of 74 out of 75 correct recognition

We discussed the application of the bimanual movements in different areas such as Virtual Reality In these applications the bimanual movements are usually used in a periodic manner while a number of them are concatenated in order to do the tasks

We employed our Bayesian network to recognise the concatenated periodic movements

In order to get a stable belief at the root node of the Bayesian network during the periodic movements we changed the belief propagation algorithm by replacing the prior probability by the belief of the root node of the previous step The network resulted in a very stable condition However, due to the numerical underflows the network's response to the gesture changes was delayed severely

Therefore, we constrained the prior probability of the root node not to fall below a certain level of belief Using this new algorithm the network's performance improved dramatically in recognising the concatenated periodic bimanual movements We called this network the Bayesian network with short-term memory

The proposed rules were able not only to stabilise the belief of the network but also to detect the movement changes quite accurately

We also tested a third version of the Bayesian network called the loopy network Despite the conventional belief that the loopy Bayesian network (the networks including loops) may not converge to a stable equilibrium we changed the structure of the original network so that it included a loop It was demonstrated that the algorithm converges to approximate posterior marginals on the correct side of the decision line

We also presented the reasons for the convergence of the algorithm by an unwrapping method In this method a pairwise Markov net models the Bayesian network An unwrapping technique unwraps the Markov net in different stages corresponding to different turns that the messages circulate in the loopy network Since the unwrapped network is singly connected its convergence to the correct posterior marginals is guaranteed

An important parameter in the loopy networks is the convergence rate of the belief propagation algorithm We proposed a new analysis in which we showed the convergence rate of the algorithm is related to the evidences provided to the network We formalised the conditions where the loopy propagation converges rapidly under different circumstances A set of simulation results was presented regarding the analysis of the loopy network convergence rate

We employed the three proposed Bayesian networks, the original network, the network with the short-term memory, and the loopy network to recognise the bimanual gestures in the test set All the networks showed the high performance of 74/75 recognition rate While the network with short-term memory was proven to result in more stable beliefs the loopy network represented overly confident results in recognising the gestures

We also, employed the three networks to recognise a set of concatenated periodic bimanual movements The original network and the loopy network resulted in quite the same recognition rate of 84 4% and 83 4% recognition rates respectively The network with short-term memory, however, demonstrated a superior recognition rate of 87 4% Since the beliefs in the network with short-term memory are stabilised it is more robust than the other networks in recognising the periodic bimanual gestures

The results show that the techniques and the algorithms we presented for tracking the hands and recognising the bimanual movements work robustly and effectively in the real applications We believe that this project as the first project ever in tracking and recognising bimanual movements is a big step toward a complete movement recognition system We tried to introduce general solutions with the least restrictions in every step of the project

The tracking algorithm as an example works independent of the hand shapes or the position of the camera Therefore, in applications where the camera can be positioned at different places (e g surveillance applications) the proposed algorithm can track the hand motions

accurately Since the algorithm does not work based on the hand shapes or the type of movement it correctly tracks the hands from any angle of view

Also the proposed Bayesian networks for data fusion and recognition of bimanual movements work independent of the type of movements Therefore, if the partial Hidden Markov Models are trained so as they are able to recognise the partial gestures appearing in a segmented movement from any angle of view the Bayesian networks can fuse the partially recognised gestures to recognise the whole movement

In the next section we propose some further possible work in order to improve the proposed system to cover more general problems

9 2 Future Work

We must compare the proposed Bayesian networks for recognition of bimanual movements with other techniques Fuzzy Logic, Neural Networks, and Dempster-Shafer theory are the alternatives to the proposed Bayesian networks that must be considered and compared

The techniques for recognition of hand shapes from different angle of views can be employed in order to recognise a partial gesture even if the system is not trained to do so This idea can provide the movement recognition systems with a recognition power to understand the movements of people from any angle, which can be very useful in the Virtual Reality and surveillance systems

The recognition of movements from any angle of view requires the recognition of occasional partially hidden gestures For example, in a movement where one or both hands are hidden behind the body for some moments, the recognition of the whole movement cannot be complete without recognising the hidden part

A solution to this problem is to recognise the hidden part based on the previously seen sequence of the movements A technique called *Probabilistic Suffix Automata (PSA)* has been proposed in the literature [Ron 1996] with application in different areas including natural language processing

This model is a variant of order L Markov chains in which the order (or the memory) is variable When PSA generates a sequence, the probability distribution on the next generated element is completely defined given the previously generated sequence Therefore, in a sequence of hand shapes the hidden part can be predicted based on the previously observed sequence of hand shapes This estimation can be improved by a smoothing filter given the observed sequence of the hand shapes after the hidden part

In the bimanual movements where the canonical gestures are closely concatenated to imply a meaning, e g the British Sign Language, the recognition of a partial gesture in a segmented movement may entail the recognition of many canonical gestures appearing in the segment In this case each of the canonical gestures must be recognised separately

An approach to this problem can be the Hierarchical Hidden Markov Models (HHMMs) [Fine 1998] In these models, unlike the conventional Hidden Markov Models, every state is a HHMM as well Therefore, the states output sequences rather than a single symbol These

sequences are produced by activating the submodels each of which with a different length Given that a segment in a segmented bimanual movement is modelled by a HHMM the canonical gestures of different length can be recognised by the sub-HHMMs

The bimanual movement recognition can be a part of larger research in order to understand all human movements Still the recognition of human movements particularly in the presence of occlusion in low-resolution images is an open research area

An enormous number of applications in Virtual Reality (VR) are waiting for the new models and algorithms of hand and body gesture recognition to realise the wish for human-computer natural interfacing Freedom of movement in Virtual Reality environments is a wish that when it comes true, thousands of VR systems around the world have been waiting to utilise it

Recognition of speech, hand gestures, facial expressions and body movements should be combined so that a person in a short time-period is totally understood by a machine Given a trained model of hand movements and body expressions a robotic system can imitate the human behaviour

The proposed bimanual recognition system can be used to train the models of human behaviour for hand movements An example is the recognition of the hands' movements in different moods of a person An angry person normally moves his/her hands faster with more stress on the meaningful parts of the movements Instead, a tired person moves the hands slowly with less stress on the hand pauses in different gestures

In the same way we can understand the mood of a person, a machine can too When the machines learn how to behave as humans they will be moved from desks and isolated rooms to the world outside

By tracking people and recognising their movements they can find their position in the daily life of communities in order to help people improve the quality of life

APPENDIX A

JAI CV-M40 Monochrome Camera Data Sheet[1]

Specifications for CV-M40

Specifications	CV-M40
Scanning system	Progressive 525 lines 60 frames sec
Frame rate Normal Vertical binning Partial 1/2 Partial 1/4 Partial 1/8 Partial 1/16	60 Hz 648 (h) x 492 (v) pixel 120 Hz 648 (h) x 242 (v) pixels 180 Hz 648 (h) x 240 (v) pixels 150 Hz 648 (h) x 120 (v) pixels 200 Hz 648 (h) x 60 (v) pixels 233 Hz 648 (h) x 30 (v) pixels
Line frequency	31.469 kHz
Pixel frequency	24.54 MHz
CCD sensor	Monochrome 1/2" Hyper HAD IT progressive scan CCD
Sensing area	6.4 (h) x 4.8 (v) mm
Picture elements	659 (h) x 494 (v)
Effective pixels	648 (h) x 484 (v)
Cell size	9.9 x 9.9 µm
Resolution (horizontally)	480 TV lines
Resolution (vertical)	480 TV lines
Sensitivity (on axis)	0.23 Lux Max 2000 video
S/N ratio	>48 dB (AGC off, Gamma 1)
Gamma	0.45 1.0
Gain	Auto or manual (0 to 12 dB)
Video output	Composite VS signal 1.0 p-p 75 Ohm or video without sync 0.7 Vp-p 75 Ohm
Readout system Normal Vertical binning	1 progressive frame (1/60 sec) 525 lines 1 progressive frame (1/120 sec) 262 lines
Partial scan With RS-232C	Full 1/2 1/4 1/8 1/16 From 3 to 480 lines
Synchronization	Int. X'tal or Ext. HD/VD or random trigger
HD/VD sync input/output	4V 75 Ohm or TTL
Trigger readout modes	Normal/Edge Pulse width/Frame delay readout
Trigger input	32 pulse stream 4V 75 Ohm or TTL
Shutter	Off 1/250 1/500 1/1000 1/2000 1/4000 1 None 1 (2 sec) sec
WEN output	TTL
Pixel clock output	TTL
Serial control	RS-232C
Controls and functions Gamma Gain Scanning format Readout mode Trigger Readout modes Shutter Manual scan Gain Setup White clip File Memory Memory	 0.45 1.0 Fixed/Manual/Auto Full 1/2 1/4 1/8 1/16 Normal/Vertical binning Normal/Edge Pulse width/Frame delay readout Off to 1/2 sec sec in 8 steps Potentiometer on rear plate Relative 0 255 Relative 0 255 Relative 0 255 Load to and from file Restore and store user setup Restore factory setup
Operating temperature	5 C to 45 C
Humidity	20 80 % non condensing
Power	12V DC ±10% 0.5 Amp
Lens mount	C mount
Dimension	40 x 50 x 80 mm (Hx WxD)
Weight	240 g

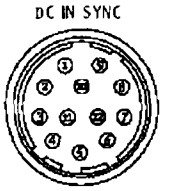
Ordering Information

CV-M40 1/2" Double Speed Progressive Scan Monochrome camera (CIA compliance)
12P Hirose Plug
12S Hirose Plug
Cable for RS-232C Interface

Must be ordered separately

Connection Description

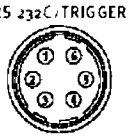
DC IN/SYNC



Hirose HR10A-10P-12P Male

Pin 1	Ground
2	+12V DC
3	Ground
4	Video output
5	Ground
6	HD in/output/ Trigger input
7	VD in/output/ WEN output
8	Ground
9	Pixel clock output **
10	Ground
11	+12V DC
12	Ground

RS-232C/TRIGGER



Hirose HR10A-7P-6S Male

Pin 1	TXD
2	RXD
3	Ground
4	Ground
5	Trigger input
6	WEN output

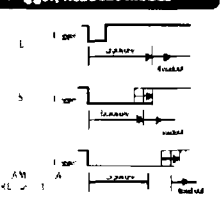
** Pixel clock output is a square wave setting

Partial Scan Example

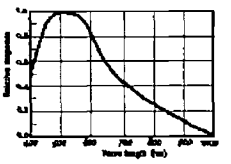
Normal

Partial

Trigger/Readout Modes

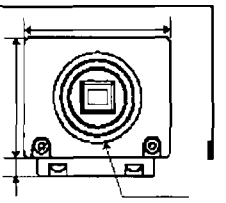


Spectral Sensitivity

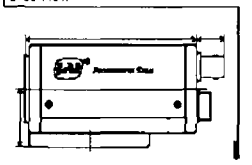


Dimensions

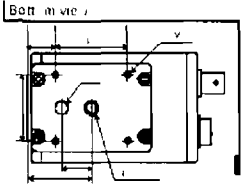
Front view



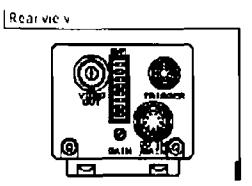
Side view



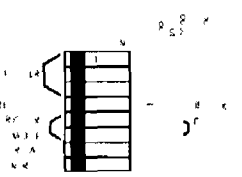
Bottom view



Rear view



Switch Setting



JAI A/S Denmark
Phone 45 44 / 8888
Fax 45 4491 8880
www.jai.com

JAI Corporation Japan
Phone 81 4 933 4 0
Fax 81 45 931 6146
www.jai-corp.co.jp

JAI UK Ltd. England
Phone 44 1442 8 9 669
Fax 44 1442 8 9 281
www.jai.com

JAI America Inc. USA
Phone 1 949 472 5900
Fax 1 949 472 5 08
www.jai.com

JAI OY Finland
Phone 3 8 9 8 5 2 2
Fax 358 9 8 0 3 3 5

THE MECHADEMIC COMPANY

Visit our web site on www.jai.com

[1] Source of data sheet: www.jai.com

APPENDIX B

A MATHEMATICAL DESCRIPTION OF VECTOR QUANTISATION

Let T be a training set of M vectors given by $T = \{x_1, x_2, \quad, x_M\}$ where each vector is k-dimensional,

$$x_m = \begin{bmatrix} x_{m1} \\ x_{m2} \\ \\ x_{mk} \end{bmatrix}, \; m = 1, 2, \quad, M$$

Let N be the number of codevectors and $C = \{c_1, c_2, \quad, c_N\}$ represents the codebook Each codevector is k-dimensional,

$$c_n = \begin{bmatrix} c_{n1} \\ c_{n2} \\ \\ c_{nk} \end{bmatrix}, \; n = 1, 2, \quad, N$$

Let s_n be the encoding region associated with the codevector c_i and $P = \{s_1, s_2, \quad, s_N\}$ denotes the partition of the space

Then the Vector Quantisation algorithm is as following,

1 Given T Fix $\varepsilon > 0$ to be a small number

2 Let $N=1$ and

$$c_1^* = \frac{1}{M} \sum_{m=1}^{M} x_m \qquad \text{(B 1)}$$

Calculate

$$D_{avg}^{*} = \frac{1}{M\,k}\sum_{m=1}^{M}\left\|x_m - c_1^{*}\right\|^2 \tag{B 2}$$

3 Splitting For $i=1, 2 \quad N$, set

$$\begin{aligned} c_i^{(0)} &= (1+\varepsilon)c_i^{*}, \\ c_{N+i}^{(0)} &= (1-\varepsilon)c_i^{*} \end{aligned} \tag{B 3}$$

Set $N=2N$

4 Iteration Let $D_{avg}^{(0)} = D_{avg}^{*}$ Set the iteration index $i=0$

I For $m=1\ 2 \quad , M$, find the minimum value of

$$\left\|x_m - c_n^{(i)}\right\|^2 \tag{B 4}$$

over all $n=1\ 2 \quad , N$ Let n^{*} be the index which achieves the minimum Set

$$Q(x_m) = c_{n^{*}}^{(i)} \tag{B 5}$$

II For $n=1, 2, \quad , N$, update the codevector

$$c_n^{(i+1)} = \frac{\sum_{Q(x_m)\in c_n^{(i)}} x_m}{\sum_{Q(x_m)\in c_n^{(i)}} 1} \tag{B 6}$$

III Set $i = i + 1$

IV Calculate

$$D_{avg}^{(i)} = \frac{1}{M\,k}\sum_{m=1}^{M}\left\|x_m - Q(x_m)\right\|^2 \tag{B 7}$$

V If $\frac{D_{avg}^{(t-1)} - D_{avg}^{(t)}}{D_{avg}^{(t-1)}} > \varepsilon$, go back to Step (I)

VI Set $D_{avg}^{*} = D_{avg}^{(t)}$ For $n=1, 2, \ldots N$, set $c_n^{*} = c_n^{(t)}$ as the final codevectors

5 Repeat Steps 3 and 4 until the desired number of codevectors is obtained

APPENDIX C

KALMAN FILTERING PROCESS AND THE TRACKING MODEL OF CHAPTER 5

C 1 Kalman Filter

To explain this filter we assume the process to be estimated can be modelled in the form [Brown 1997],

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k \qquad \text{(C 1)}$$

At discrete points in time the measurement of the process occurs with the following linear relationship,

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \qquad \text{(C 2)}$$

where

$\mathbf{x}_k$ the state vector of process at time t_k

$\mathbf{\Phi}_k$ a matrix relating x_k to x_{k+1}

$\mathbf{w}_k$ a white noise sequence with known covariance structure

$\mathbf{z}_k$ measurement vector at time t_k

$\mathbf{H}_k$ matrix giving the noiseless connection between the measurement and the state vector at time t_k

$\mathbf{v}_k$ measurement error – assumed to be a white noise sequence with known covariance structure

The covariance matrices for the $\mathbf{w}_k$ and $\mathbf{v}_k$ are given by,

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} \mathbf{Q}_k & i = k \\ 0 & i \neq k \end{cases}$$

$$E[\mathbf{v}_k \mathbf{v}_i^T] = \begin{cases} \mathbf{R}_k & i = k \\ 0 & i \neq k \end{cases}$$

$$E[\mathbf{w}_k \mathbf{v}_i^T] = 0 \quad \textit{for all } k \textit{ and } i$$

The third equation above shows that the measurement error $\mathbf{v}_k$ and the system error $\mathbf{w}_k$ are not correlated. The state-vector $\hat{\mathbf{x}}_k^-$ represents an initial estimate of the process at time t_k. This estimate is based on all our knowledge about the process prior to t_k. Therefore, the estimation error is given by,

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-$$

with the covariance matrix,

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \qquad \text{(C 3)}$$

To improve the prior estimate $\hat{\mathbf{x}}_k^-$ a linear equation is chosen with a mixture of noisy measurement and the prior estimate

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \qquad \text{(C 4)}$$

where

$\hat{\mathbf{x}}_k$ the updated estimate

$\mathbf{K}_k$ a factor to be determined

$\mathbf{K}_k$ should be determined so as the update estimate is optimal. The error covariance matrix associated with the updated estimate is,

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \qquad \text{(C 5)}$$

By substituting C 2 into Equation C 4 and then into C 5 the error covariance matrix is obtained as,

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \qquad \text{(C 6)}$$

The individual terms along the major diagonal of $\mathbf{P}_k$ represent the estimation error variance for the elements of the state-vector being estimated. Therefore, $\mathbf{K}_k$ should be determined

so as these terms are minimised The optimisation problem can be done in different ways addressed in the literature [Brown 1997]

The optimal $\mathbf{K}_k$ is calculated as,

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{C 7}$$

which is known as the *Kalman gain* By substituting the optimal $\mathbf{K}_k$ in Equation 5 6 the error covariance matrix for updated estimate will be,

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^- \tag{C 8}$$

The updated estimate $\hat{\mathbf{x}}_k$ is projected ahead by the transition matrix,

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_k \tag{C 9}$$

By calculating the error covariance matrix for $\hat{\mathbf{x}}_{k+1}^-$ and substituting in Equation C 4,

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{Q}_k \tag{C 10}$$

Equations C 5, C 7, C 8, C 9 and C 10 form the Kalman filtering algorithm

C 2 Dynamic Model's Kalman Filter Equations

Given the following stochastic description of the tracking model,

$$\begin{cases} \begin{bmatrix} x_{k+1}(1) \\ x_{k+1}(2) \\ x_{k+1}(3) \end{bmatrix} = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k(1) \\ x_k(2) \\ x_k(3) \end{bmatrix} + \mathbf{w}_k \\ z_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k(1) \\ x_k(2) \\ x_k(3) \end{bmatrix} + v_k \end{cases} \tag{C 11}$$

If the initial condition $E(x_0)$ and $Variance(x_0)$ are given, the Kalman filtering algorithm for this model is obtained as follows [Chui 1999]

With $\mathbf{P}_0 = Variance(x_0)$, the Kalman gain,

$$\mathbf{K}_k = \frac{1}{P_k^-(1,1) + R_k(1,1)} \begin{bmatrix} P_k^-(1,1) \\ P_k^-(1,2) \\ P_k^-(1,3) \end{bmatrix} \tag{C 12}$$

The updated estimate with measurement z_k,

$$\begin{bmatrix} \hat{x}_k(1) \\ \hat{x}_k(2) \\ \hat{x}_k(3) \end{bmatrix} = \begin{bmatrix} 1-K_k(1) & h(1-K_k(1)) & \frac{h^2}{2}(1-K_k(1)) \\ -K_k(2) & 1-hK_k(2) & h-\frac{h^2K_k(2)}{2} \\ -K_k(3) & -hK_k(3) & 1-\frac{h^2K_k(3)}{2} \end{bmatrix} \begin{bmatrix} \hat{x}_k^-(1) \\ \hat{x}_k^-(2) \\ \hat{x}_k^-(3) \end{bmatrix} + \begin{bmatrix} K_k(1) \\ K_k(2) \\ K_k(3) \end{bmatrix} z_k \tag{C 13}$$

with $\hat{x}_0 = E[x_0]$

The error covariance for updated estimate is given by,

$$\mathbf{P}_k = P_k^- - \frac{1}{P_k^-(1,1) + R_k(1,1)} \begin{bmatrix} P_k^{-\,2}(1,1) & P_k^-(1,1)P_k^-(1,2) & P_k^-(1,1)P_k^-(1,3) \\ P_k^-(1,1)P_k^-(1,2) & P_k^{-\,2}(1,2) & P_k^-(1,2)P_k^-(1,3) \\ P_k^-(1,1)P_k^-(1,3) & P_k^-(1,2)P_k^-(1,3) & P_k^{-\,2}(1,3) \end{bmatrix} \tag{C 14}$$

and the prior error covariance,

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}\mathbf{P}_k\mathbf{\Phi}^T + \mathbf{Q}_k$$

with

$$\mathbf{\Phi} = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix}$$

is calculated as following,

$$P_{k+1}^{-}(1,1) = P_k(1,1) + 2hP_k(1,2) + h^2P_k(1,3) + h^2P_k(2,2) + h^3P_k(2,3) + \frac{h^4}{4}P_k(3,3) + Q_k(1,1)$$

$$P_{k+1}^{-}(1,2) = P_k(1,2) + hP_k(1,3) + hP_k(2,2) + \frac{3h^2}{2}P_k(2,3) + \frac{h^3}{3}P_k(3,3) + Q_k(1,2)$$

$$P_{k+1}^{-}(2,1) = P_{k+1}^{-}(2,1)$$

$$P_{k+1}^{-}(2,2) = P_k(2,2) + 2hP_k(2,3) + h^2P_k(3,3) + Q_k(2,2)$$

$$P_{k+1}^{-}(1,3) = P_k(1,3) + hP_k(2,3) + \frac{h^2}{2}P_k(3,3) + Q_k(1,3)$$

$$P_{k+1}^{-}(3,1) = P_{k+1}^{-}(1,3)$$

$$P_{k+1}^{-}(2,3) = P_k(2,3) + hP_k(3,3) + Q_k(2,3)$$

$$P_{k+1}^{-}(3,2) = P_{k+1}^{-}(2,3)$$

$$P_{k+1}^{-}(3,3) = P_k(3,3) + Q_k(3,3)$$

The prediction step is calculated by,

$$\begin{bmatrix} \hat{x}_{k+1}^{-}(1) \\ \hat{x}_{k+1}^{-}(2) \\ \hat{x}_{k+1}^{-}(3) \end{bmatrix} = \begin{bmatrix} 1 & h & \frac{h^2}{2} \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_k(1) \\ \hat{x}_k(2) \\ \hat{x}_k(3) \end{bmatrix} \qquad \text{(C 15)}$$

In these equations $\bar{x}_k(1)$ denotes the position, $\bar{x}_k(2)$ denotes the velocity and $\bar{x}_k(3)$ is the acceleration of the hand central point at time t_k

APPENDIX D

HIDDEN MARKOV MODELS EVALUATION AND LEARNING PROBLEMS

D 1 Forward Backward Algorithm

In this algorithm the probability of the partial observation sequence $y_1, y_2, \ldots, y_t$ is calculated by,

$$\alpha_t(i) = p(y_1, y_2, \ldots, y_t, s_t = i) \tag{D 1}$$

where α_t is a recursively calculated auxiliary variable Then with boundary conditions,

$$\alpha_1(j) = \pi_j b_j(y_1) \qquad 1 \le j \le N \tag{D 2}$$

the following recursive relationship holds,

$$\alpha_{t+1}(j) = b_j(y_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij}, \quad 1 \le j \le N \quad 1 \le t \le T-1 \tag{D 3}$$

Using this recursion we can calculate $\alpha_T(i)$, $1 \le i \le N$ The required probability is given by,

$$p(Y) = \sum_{i=1}^{N} \alpha_T(i) \tag{D 4}$$

This method has a complexity proportional to $N^2 T$ In a similar way the backward variable $\beta_t(i)$ is calculated as the probability of partial observation sequence $y_{t+1}, y_{t+2}, \ldots, y_T$ given the current state i,

$$\beta_t(i) = p(y_{t+1}, y_{t+2}, \ldots, y_T \mid s_t = i) \tag{D 5}$$

Again a recursive relationship holds to calculate $\beta_t(i)$ efficiently,

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(y_{t+1}), \quad 1 \le i \le N, \quad 1 \le t \le T-1 \tag{D 6}$$

where

$$\beta_T(i) = 1, \quad 1 \le j \le N$$

We can see that [Warakagoda 1996],

$$\alpha_t(i)\beta_t(i) = p(Y, s_t = i) \quad 1 \le i \le N, \quad 1 \le t \le T \tag{D 7}$$

Therefore, by using both forward and backward variables,

$$P(Y) = \sum_{i=1}^{N} p(Y, s_t = i) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \tag{D 8}$$

Full description and expansion of the equations of the Forward/Backward algorithm has been presented in [Warakagoda 1996]

D 2 Learning Problem

The learning problem is to adjust the HMM parameters so that the given observations in the training set are represented with maximum probability by the model There are different methods for learning We describe a method based on Maximum Likelihood (ML)

In Maximum Likelihood the probability of a given sequence of observations Y belonging to a given class c, given the HMM for this class, is to be maximised This probability is the total likelihood of observations,

$$L_{total} = p(Y \mid \lambda_c) \tag{D 9}$$

where λ_c denotes the HMM of class c

Since we consider only one class at a time we drop the subscript c There is no known way to analytically solve for the model $\lambda = (A, B, \pi)$ which maximise the L_{total} But an iterative method can be used so that the parameters are maximised locally

Baum Welch Algorithm

By defining a new variable as the probability of being in state i at time t and in state j at $t+1$, we have,

$$\xi_t(i,j) = p(s_t = i,\ s_{t+1} = j \mid y, \lambda) \tag{D 10}$$

$$p(s_t = i, s_{t+1} = j \mid y, \lambda) = \frac{p(s_t = i, s_{t+1} = j, y \mid \lambda)}{p(Y \mid \lambda)}$$

Therefore,

$$\xi_t(i,j) = \frac{p(s_t = i,\ s_{t+1} = j, y \mid \lambda)}{p(Y \mid \lambda)} \tag{D 11}$$

By substituting the Equations D 3 and D 6 into D 11 we get,

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} \beta_{t+1}(i) b_j(y_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(y_{t+1})} \tag{D 12}$$

Also we define a second variable as the probability of being in state i at time t given the observation sequence Y

$$\gamma_t(i) = p(s_t = i \mid Y, \lambda) \tag{D 13}$$

By substituting the forward and backward variables,

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{D 14}$$

and the following relationship holds,

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j), \quad 1 \le i \le N, \quad 1 \le t \le M \tag{D 15}$$

Assuming an initial model $\lambda = (A, B, \pi)$ the forward variable αs and the backward variable βs are calculated by Equations D 3 and D 6 respectively ξs and γs are calculated using D 12 and D 15 Then the parameters of the model are updated by the re-estimation formulas,

$$\pi_i = \gamma_1(i), \qquad 1 \le i \le N \tag{D 8}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)}, \quad 1 \le i \le N, \quad 1 \le j \le N \tag{D 16}$$

$$\bar{b}_j(k) = \frac{\sum_{\substack{t=1 \\ o_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \quad 1 \le j \le N, \quad 1 \le k \le M \tag{D 17}$$

The $\bar{x}$ in the above equations stand for re-estimation of a variable x

In practice, for long sequences both α_t and β_t become small as the recursion progresses Therefore, usually they are re-normalised to sum to one at each step of the recursions This makes the computation of the relevant expectations much more numerically well-behaved, and has the nice side-effect that the sum of the log normalisations in the forward pass is the log likelihood of the observation sequence [Ghahramani 2001]

APPENDIX E

LATEST DEVELOPMENTS IN THE HAND TRACKING ALGORITHM

During occlusion the number of images should be large enough so that the velocities converge to zero in the cases of hand collisions and pauses. The Kalman filtering process proposed in Chapter 7 is based on the Kinematic equations of motion. Therefore, in a fast movement the sides of the occlusion-rectangle have the potential to move further rather than to stop quickly. The algorithm should have enough time and images so that the rectangle sides' velocities reach zero in the cases that a collision or pause is detected.

If the speed of movement increases the estimated speeds of the rectangle sides may not exactly reach zero. This problem becomes more difficult if the camera is working in a low speed (low frame rate). Therefore, the algorithm may not detect the collision and pauses accurately and may run into trouble. Also in some applications where the visual system moves (e.g. active vision) the velocities may not exactly reach zero. Therefore, we need a technique to make the algorithm independent from the actual velocities.

To deal with these problems we investigate the speed changes of the occlusion-rectangle sides.

When a pause occurs the estimated velocity tends to zero. Assume that the hands are moving toward each other with almost constant velocities. The acceleration is zero. When a pause occurs the acceleration increases in the negative direction in order to push the velocity to zero. The graph of acceleration looks approximately like a negative quadratic polynomial (see Figure E.1). The velocity is the integral of the acceleration,

$$v_t = \int a_t \, dt \qquad \text{(E.1)}$$

Therefore, the graph of the velocity obeys a polynomial of power 3 (see Figure E.2).

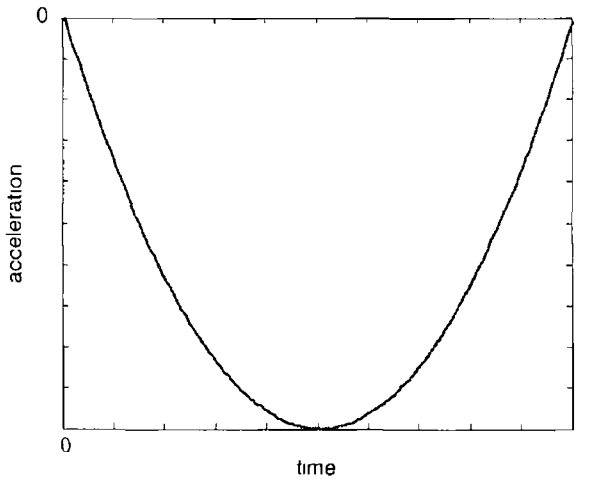


Figure E 1 Acceleration changes in a movement where pause is detected (approximate graph)

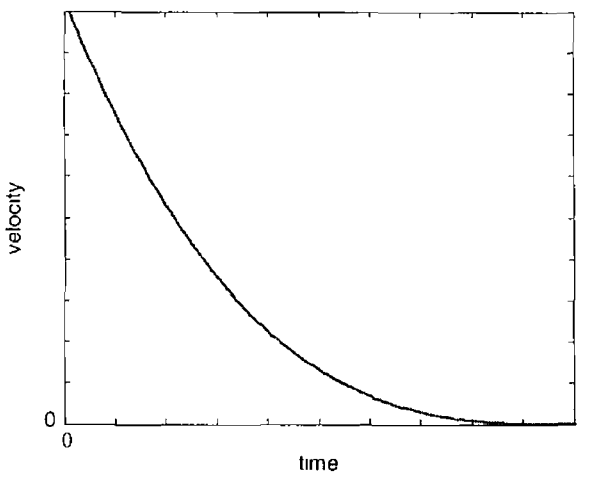


Figure E 2 The graph of the velocity changes in a movement where a pause is detected (from the beginning of occlusion to the pause point)

After the pause the rectangle sides move in opposite directions The velocity changes in the same fashion but in the negative direction Therefore, the graph of the velocity during the occlusion period looks like Figure E 3

We may approximate this graph by a polynomial of power three or a logarithmic function like negative arc tangent hyperbolic,

$$f_1 = \alpha x^3$$
$$f_2 = \frac{\ln(1+x) - \ln(1-x)}{2} \qquad \text{(E 2)}$$

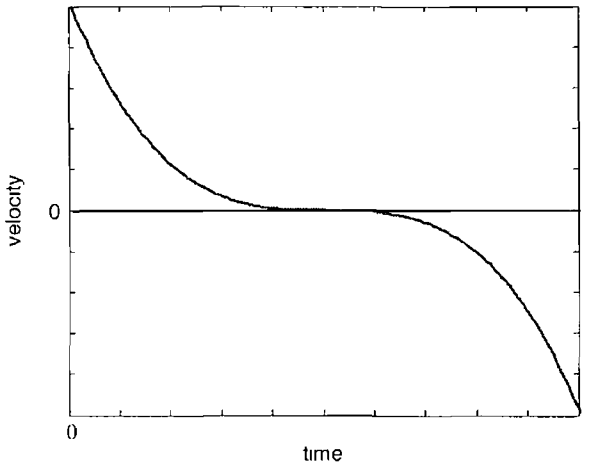


Figure E 3 The graph of velocity changes in a movement where a pause is detected in the whole period of occlusion

Also, in the cases where hands pass each other the velocity of a rectangle side looks like Figure E 4 The rapid sign change in the graph is due to the fact that when one hand passes the other, it pushes the rectangle sides in opposite direction This graph looks like a logistic function,

$$f_v = \frac{1}{1+e^{\alpha v}} \qquad \text{(E 3)}$$

One may conclude that by approximating the velocity changes with the above equations we can make a decision on hand pauses A velocity change that better matches the negative arc tangent hyperbolic is more probable to be a hand pause And a velocity change that better matches the logistic function is more probable to be a hand pass

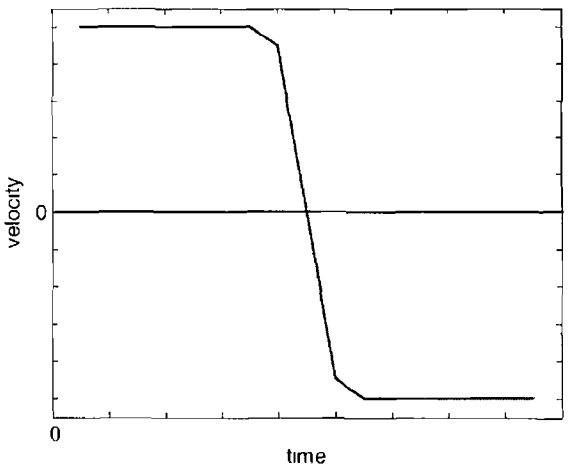


Figure E.4 The velocity changes in a movement where hands pass each other

Although this seems true theoretically, our experiments did not demonstrate a good performance for the algorithm. This is due to velocity variations in different problems and speed of movement that prevents using a constant analytic model.

According to a neuroscience theory [Harris 1998], there exists noise in the motor commands. In the presence of such noise the intended motor commands will generate a probability distribution over the hand positions and velocities if repeated several times [Wolpert 2001].

In accordance with this theory, we model the velocity changes by gaussian distributions. In this model the graphs of the velocity changes are approximated by a sequence of gaussian distributions for the two main categories hand-pause and hand-pass.

As in Chapter 4, 2-dimensional gaussian distributions are constructed by a set of training data. The training data set is the velocity changes of the occlusion-rectangle sides.

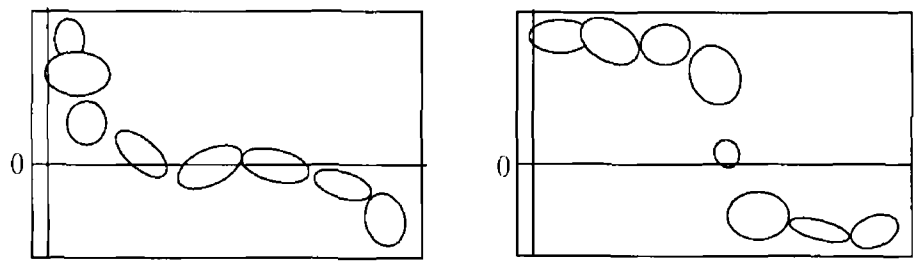
The following function is defined in order to represent a pair of parallel sides of the occlusion-rectangle,

$$v(t) = v_1(t) - v_2(t) \qquad \text{(E.4)}$$

where $v_1(t)$ and $v_2(t)$ are the velocities of two parallel sides at time t. When the hands are negatively synchronised, this function results in a velocity equal to the sum of the individual absolute velocities. An important feature of this function is that it makes the algorithm

independent of the actual velocities Therefore, in some applications (e g active vision) the effect of a constant value added to the both velocities is eliminated

In the movements where a pair of parallel sides are positively synchronised the velocity-synchronisation model (System 7 11) captures the synchronisation The gaussian models of velocity changes of Function E 4 are shown in Figure E 5 Figure E 5 (a) shows the pattern of Function E 4 for a pair of parallel sides in the movements where a pause is detected Figure E 5 (b) shows this pattern for the movements where the hands pass each other In these figures, each ellipse shows a 2-dimensional gaussian distribution These models approximate a given sequence of velocity changes A decision on whether the hands have passed each other or paused and returned is made based on the probabilities that the pattern of Function E 4 matches each of these patterns



(a) (b)

Figure E 5 The sequences of gaussian distributions to model the occlusion rectangle velocities during the two main categories (a) hand pause (b) hand pass

$$P(v_o \mid H_t) = \prod_J \max_k (P(v_o^J \mid H_t^k)) \qquad \text{(E 5)}$$

where v_o^J is the observed velocity at time J during occlusion, H_t^k is the k^{th} class of gaussian distribution in the HyperClass H_t, and $P(v_o^J \mid H_t^k)$ is calculated using the probability density function of Equation 4 18

In order to train the distributions we must classify the data points for each gaussian distribution H_t^k in the pattern H_t Vector Quantisation (VQ) as an unsupervised clustering technique can cluster the data points By applying VQ to a set of training velocity data points in each pattern the data points of each distribution are classified Then by using Principal Component Analysis the parameters of the gaussian distributions are determined

Using this pattern matching technique, we can reliably detect the hand pauses even if the velocities do not reach zero

Experimental Results

We use the proposed model in order to track all types of the movements Figure E 6 shows the velocity changes of a large set of moderate and fast movements In Figures E 6 (a) and (b) the hands face either collisions or pauses In Figures E 6 (c) and (d) the hands pass each other In order to get a better view how the graphs differ from each other the graphs of the horizontal pause (see Figure E 6 (a)) and the horizontal pass (see Figure E 6 (c)) are plotted together in Figure E 7

Using four training sets each of which contained thousands of data points taken in fast, moderate and slow movements, four models were constructed Each of the models includes eight gaussian distributions We employed the Vector Quantisation algorithm to cluster the data points to make 8 clusters in each case

1 Performance

Using the trained model we performed two hundred experiments for each type of movement, one hundred in the class of moderate/fast movements and one hundred in the class of slow/moderate movements The measured performances of the algorithm are presented in Tables E 1

In both classes of speed the proposed algorithm works very well with 12% and 5% error rate which is a good performance given that the algorithm is independent of the camera view direction, changing hand shapes, and the type of movements This is an improvement to the algorithm of Chapter 7 because it has a better performance and a wider range of applications due to the independence of the algorithm from the actual velocities

In the next section we present two sets of experiments to demonstrate the independence of the algorithm from the camera view direction and its application in active vision such as mobile robots

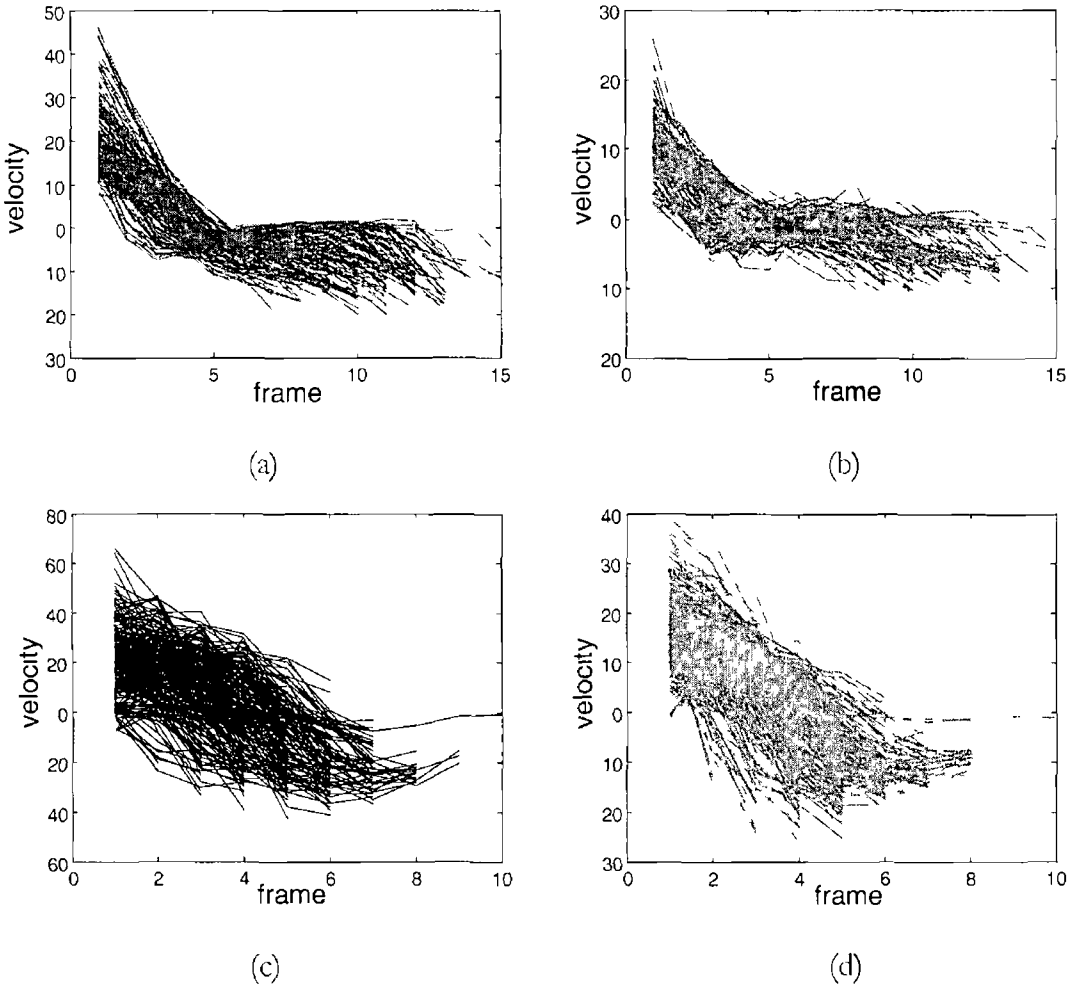


Figure E 6 Velocity changes of Equation 7 11 for (a) and (b) horizontal and vertical pause or collision (c) and (d) horizontal and vertical hand pass

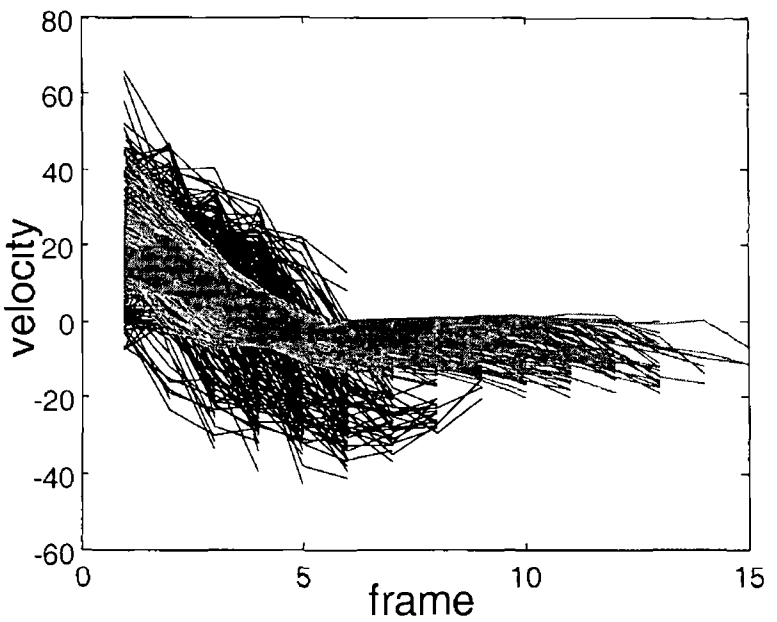


Figure E 7 The horizontal pause and horizontal hand pass

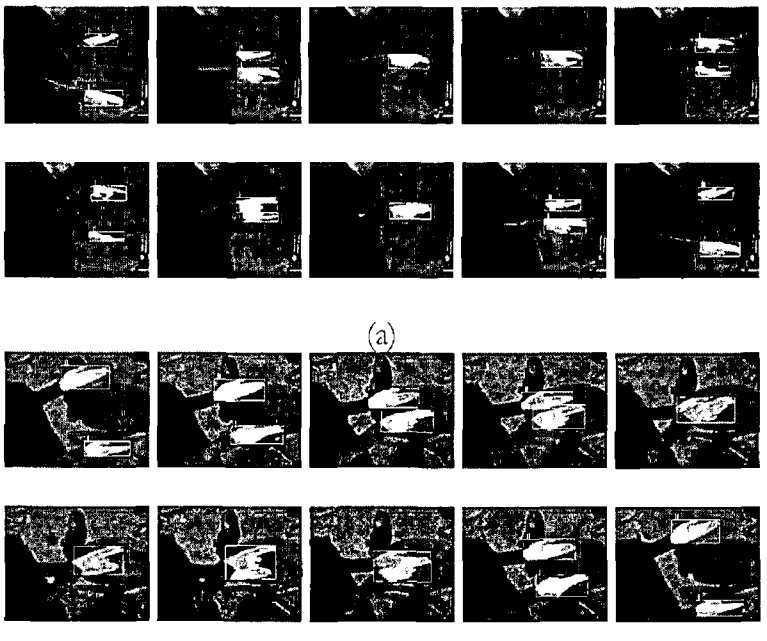
Table E 1 Performance of the tracking algorithm in the two classes of speeds

speed class	# movements	# fails	% correct tracking
slow/moderate	800	39	95 13
moderate/fast	800	95	88 13

2 An Untrained View Direction

The tracking algorithm was developed as a direction-independent algorithm In this experiment, we change the camera view direction from the side-frontal view to a top-corner view as in Figure 7 26 of Chapter 7 The algorithm has been only trained in the original view direction Given that we defined the movement models to cover almost every angle of view, we test the tracking algorithm from the direction in which the algorithm is not trained for

As an example a movement similar to the example in Chapter 7, Section 7 4 is performed twice in which from the original view it is represented by the model *e* and from top-corner view by the model *j* of Figure 7 10 Some images of the two experiments are presented in Figure E 8 Despite the new untrained view direction, the algorithm tracks the hands properly in the both experiments



(a)

(b)

Figure E 8 A bimanual movement in which (a) the hands pass each other in the vertical direction from a side view, model *e*, (b) the hands do not pass but return to their previous sides from a top corner view model *j*

3 Active Vision Applications

In active vision, where the position of the camera changes, the view direction, the background, and illumination are changed In this case, the tracking algorithms that assume these parameters constant cannot cope with the changes and may fail to track the hands correctly

We did not make any assumption regarding the constant background In the active vision applications, where the visual system moves, a constant value is added to the velocities of the pairs of parallel sides of the occlusion-rectangle at each time unit However, the Function E 4 makes the constant added values ineffective Also, we considered the cases that the velocities do not reach zero in the movements that hands pause or collide All these enable us to use the proposed tracking algorithm in active vision applications

If we assume that the speed of the moving visual system is constant the proposed models can be applied to the hands movements with a constant positive or negative added value which is the speed of the visual system However, this can be a very restrictive assumption as in the real-world applications (e g mobile robots) the speed of the visual system cannot be kept always constant Therefore, we have to assume that the visual system speed is lower than the speed of hands movements In other words, the speed of visual system should not be almost equal to the speed of hands, because if the hands and the visual system have quite the same speed, a little variation in the speed of visual system can cause a wrong model to be matched with the behaviour of the hands during the occlusion period This happens when the variation in the speed of moving camera causes the occlusion detection subroutine to bounce between occlusion and non-occlusion

Three experiments are presented in order to demonstrate the ability of the tracking algorithm in tracking the hands in active vision In Figure E 9, the path of the moving visual system (the camera) with respect to the room and the subject is shown from a top-view and a side view In these movements the camera moves horizontally

The three experiments are as follows,

1 A movement of type *g* (without collision), the camera moves from the point *a* to point *b*

2 A movement of type *h*, the camera moves from the point *a* to point *b*

3 A movement of types *b* and *d*, the camera moves from the point *a* to point *b* and returns to point *a*

Some images of the three experiments are shown in Figure E 10

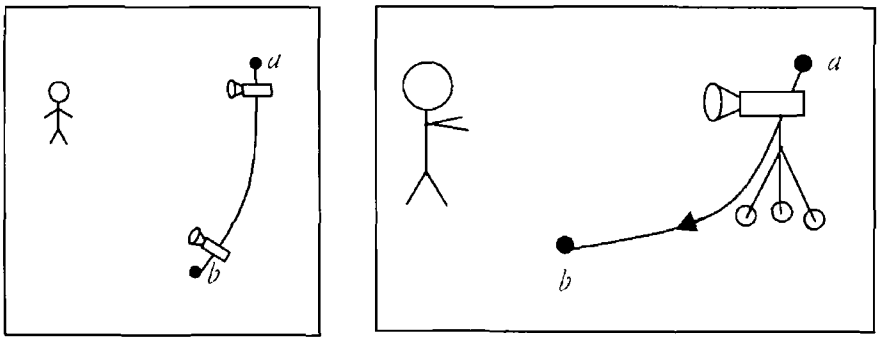


Figure E 9 The path of camera moving horizontally from a top view and a side view

Figure E 10 Tracking in a mobile camera applications (a) A movement of type *g* (b) a movement of type *b*, and (c) a movement of types *b* and *d*

In some of the images (e g the 8th image of Figure E 10(a) and 5th and 6th images in Figure E 10(b)) the extracted hands blobs are connected to some objects in the background Due to the fact that we have used a monochrome camera the connected regions of hands and the background objects are extracted as the hands' blobs However, the mis-extraction of hands' blobs has no negative effect on the tracking algorithm as the hands are correctly tracked and reacquired all over the movements

In all the movements one of the hands moves partially or completely out of image frame In Figure E 10(c), in the 8th frame the right hand is totally out of frame and there is only one hand visible in the image When the hand returns to the scene the algorithm labels it correctly and keeps tracking both hands through the rest of movement

Figure E 10(c) shows a natural movement of the visual system and the hands, which is an example of the reliability of the algorithm in tracking the hands in a natural environment

APPENDIX F

UNIMANUAL COORDINATION

In this experiment the hand moves in a circular fashion (see Figure F 1) While it is circulating the fingers are opening and closing A few images of the experiment are shown in Figure F 2

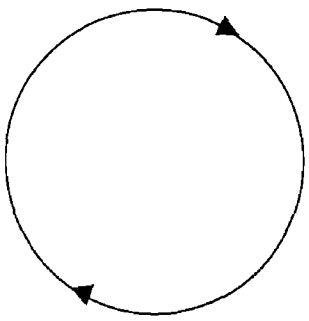


Figure F 1 The path of the hand during circular movement

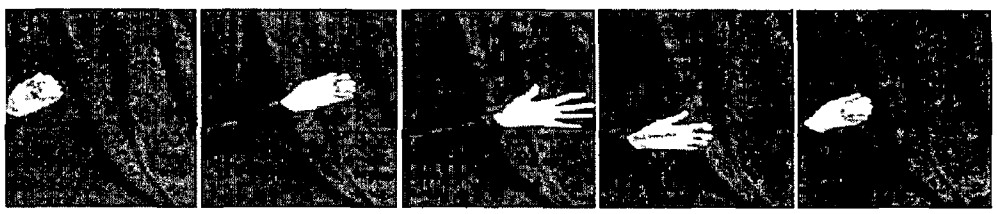


Figure F 2 The circular movement of the hand and the movement of the fingers

By constructing a rectangle around the hand (see Figure F 3) we monitor the vertical sides of this rectangle, X1 and X2 The vertical sides show the horizontal positions of the palm and the fingers The velocity of the first vertical side, X1, is the horizontal velocity of the palm, and the velocity of the second vertical side, X2, is the horizontal velocity of the palm plus the velocity of the fingers

The results of this experiment shows that the velocities are highly synchronised (see Figure F 4)

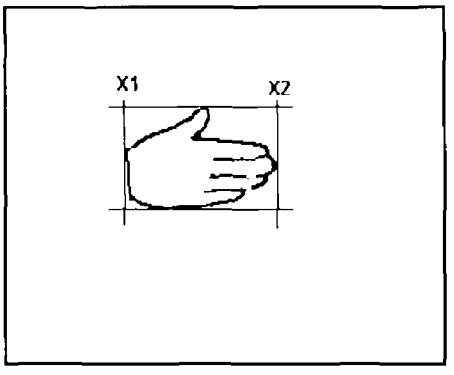


Figure F 3 A rectangle is constructed around the hand

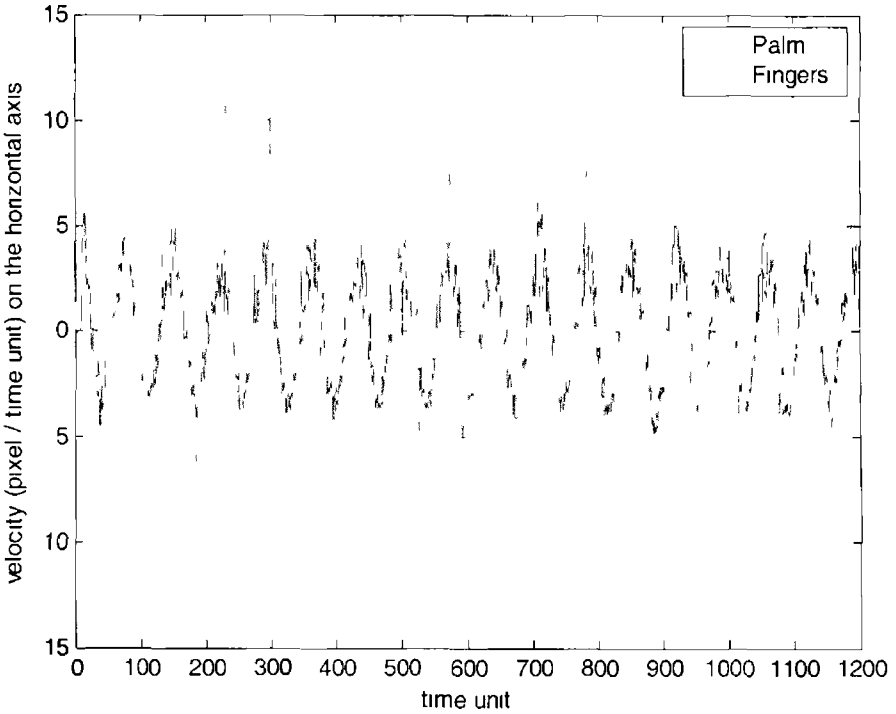


Figure F 4 The velocity of palm and fingers during a circular movement

A closer view of this graph is shown in Figure F 5 As can be seen in this figure the positive and negative peaks of the velocity of the fingers matches accurately the corresponding peaks of the velocity of the palm

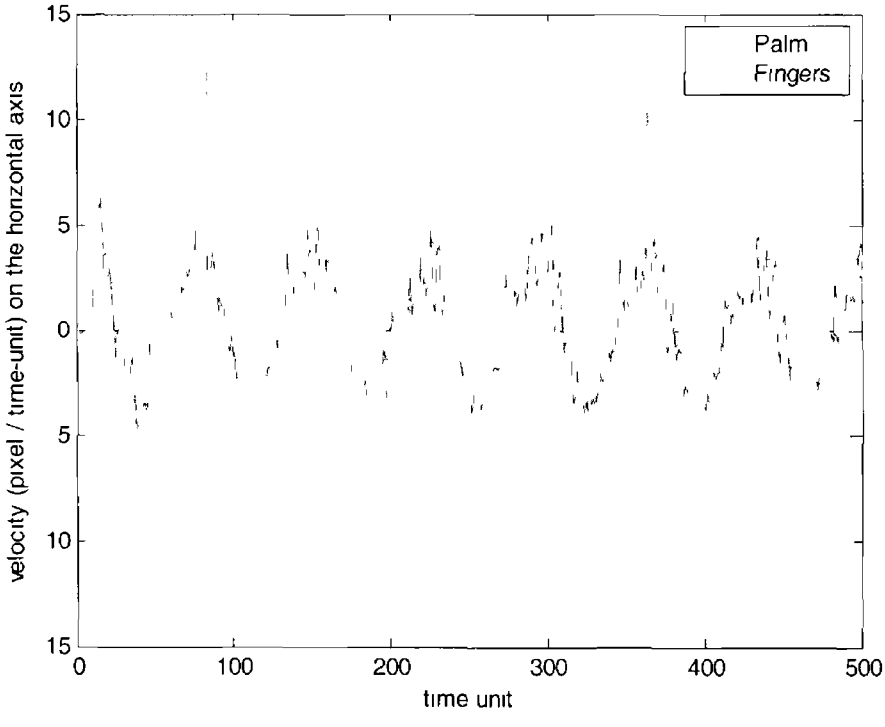


Figure F 5 The velocities of palm and fingers in vertical movement

This synchronisation enables us to track the hands in bimanual movement using the proposed algorithm of Chapter 7 even in the cases that the hand shape is changing during the movement In other words, the change of the hand shape and the movement of hand are synchronised In some of the bimanual movements where both the horizontal or vertical sides of the rectangle during occlusion are connected to one hand (see Figure F 6) the unimanual synchronisation results in concurrent pauses in both the parallel rectangle sides

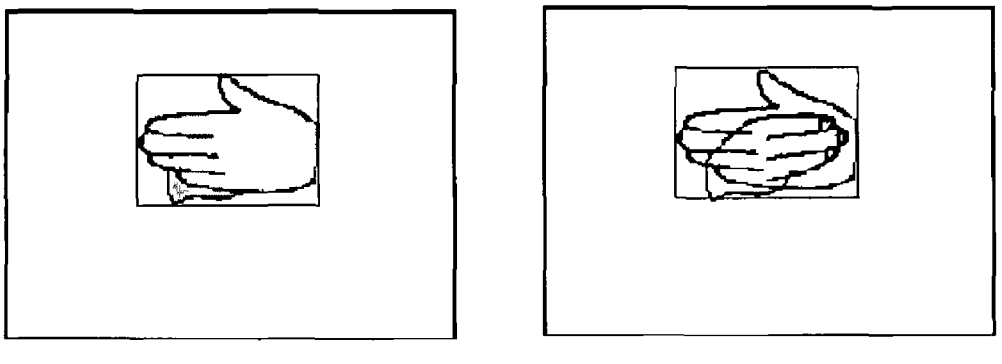


Figure F 6 The vertical sides of the rectangle are both connected to the left hand The unimanual coordination guarantees that the pauses in the vertical sides of the rectangle occur simultaneously even when the hand shape is changing

REFERENCES

Black 1996

Black M J , Jepson A D , "EigenTracking Robust Matching and Tracking of Articulated Objects Using a View-Based Representation", Int'l Journal of Computer Vision, Vol 26, No 1, 1996, pp 63-84

Bobick 1995

Bobick A , Wilson A , "Using Configuration States for the Representation and Recognition of Gesture", M I T Media Lab Technical Report, No 308, MA, 1995

Bourke 1996

Bourke P , Cross Correlation, http://astronomy.swin.edu.au/~pbourke/analysis/correlate/, August 1996

Bowden 2000

Bowden R , Sarhadi M , "Building Temporal Models for Gesture Recognition", Proc British Machine Vision Conference, BMVC 2000, Bristol, September 2000

Brand 1997

Brand M , Oliver N , Pentland A , "Coupled Hidden Markov Models for Complex Action Recognition", Proc Conf Computer Vision and Pattern Recognition, Puerto Rico, 1997

Brennan 2000

Brennan V , Principe J , "Multiresolution Using Principal Component Analysis", Proc 2000 IEEE Int'l Conf Acoustics, Speech, and Signal Processing, ICASSP'00, Istanbul, June 2000

Brown 1997

Brown R G , HwangP Y C , Introduction to Random Signals and Applied Kalman Filtering, 3rd Ed , John Wiley & Sons, USA, 1997

Bunke 2000

Bunke H , "Recent Developments in Graph Matching", Proc Int'l Conf Pattern Recognition, ICPR'00, Barcelona, 2000

Bunke 2001

Bunke H , Caelli T , Hidden Markov Models Applications in Computer Vision, World Scientific Publishing Co , Singapore, 2001

Campbell 1995

Campbell L , Bobick A , "Recognition of Human Body Motion Using Phase Space Constraints", Proc 5th Int'l Conf Computer Vision, Cambridge, MA, 1995

Chen 2003

Chen J , Yeasin M , Sharma R , "Visual Modelling and Evaluation of Surgical Skill", Pattern Analysis and Applications, Vol 6, 2003, pp 1-11

Chui 1999

Chui C K , Chen G , Kalman Filtering with Real-Time Applications, 3rd Ed , Springer Verlag, Berlin Heidelberg, 1999

Cootes 1992

Cootes T F , Taylor C J , Cooper D H , Graham J , "Training Models of Shapes from Sets of Examples" , Proc British Machine Vision Conference, 1992

Cowell 1999

Cowell R G , Dawid A P , Lauritzen S L , Spiegelhalter D J , Probabilistic Networks and Expert Systems, Springer-Verlag, New York, 1999

Dave 1992

Dave R N , Bhaswan K , "Adaptive Fuzzy c-Shells Clustering and Detection of Ellipses", IEEE Trans Neur Net Vol 3, No 5, September 1992

Davies 1997

Davies E R , Machine Vision Theory, Algorithms, Practicalities, Academic Press, 1997

Davis 1999

Davis J , Shah M , "Toward 3D Gesture Recognition", Int'l Journal of Pattern Recognition and Artificial Intelligence, Vol 13, No 3, May 1999

Diedrichsen 2001

Diedrichsen J , Hazeltine E , Kennerley S , Ivry R B , "Moving to Directly Cued Locations Abolishes Spatial Interference During Bimanual Actions", Psychological Science, Vol 12, No 6, November 2001

Diestel 1997

Diestel R , Graph Theory, Springer-Verlag, New York, 1997

Dockstader 2000

Dockstader Sh L , Tekalp A M , "Tracking Multiple Objects in the Presence of Articulated and Occluded Motion", Workshop Human Motion, HUMO'00, December 2000

Donchin 1998

Donchin O , Gribova A , Steinberg O , Bergman H , "Primary Motor Cortex is Involved in Bimanual Coordination", Nature, Vol 395, September 1998

Edwards 1997

Edwards J , Murase H , "Appearance Matching of Occluded Object Using Coarse-to-fine Adaptive Masks", Proc Conf Computer Vision and Pattern Recognition, CVPR'97, Puerto Rico, 1997

Eisenstein 2001

Eisenstein J , Ghandeharizadeh S , Huang L , Shahabi C , Shanbhag G , Zimmermann R , "Analysys of Clustering Techniques to Detect Hand Signs", Int' Symp Intelligent Multimedia, Video and Speech Processing, Hong Kong, May 2001

Ellis 2001

Ellis T , "Image Formation And Acquisition", EPSRC School in Computer Vision, University of Surrey, 2001

Erenshteyn 1999

Erenshteyn R , Saxe D , Laskov P , Foulds R , "Distributed Output Encoding for Multi-Class Pattern Recognition" , 10th IEEE International Conf Image Analysis and Processing, Venice, September 1999

Fine 1998

Fine S , Singer Y , Tishby N , "The Hierarchical Hidden Markov Model Analysis and Applications", Machine Learning, Vol 32, No1, 1998, pp 41-62

Friedman 1989

Friedman J H , "Regularized Discriminant Analysis", Journal of the American Statistical Association, Vol 84, No 405, 1989, pp 165-175

Fukushima 2000

Fukushima K , "Recognition of Occluded Patterns A Neural Network Model", Proc IEEE Int'l Joint Conf Neural Networks, IJCNN'00, Como, Italy, 2000

Ghahramani 2001

Ghahramani Z , "An Introduction to Hidden Markov Models and Bayesian Networks", in Hidden Markov Models Applications in Computer Vision, World Scientific, Singapore, 2001

Gong 2002

Gong S , Ng J , Sherrah J , "On the Semantics of Visual Behaviour, Structured Events and Trajectories of Human Action", Image and Vision Computing, Vol 20, 2002, pp 873-888

Habili 2001

Habili N , Lim C C , Moini A , "Hand and Face Segmentation using Motion and Color Cues in Digital Image Sequences", IEEE Int'l Conf Multimedia & Expo 2001, Tokyo, 2001

Hager 1988

Hager W W , Applied Numerical Linear Algebra, Prentice-Hall, 1988

Harris 1998

Harris C M , Wolpert D M , "Signal-dependent Noise Determines Motor Planning", Nature, Vol 394, August 1998

Hauck 1997

Hauck A , Lanser S , Zierl C , "Hierarchical Recognition of Articulated Objects from Single Perspective Views", Proc Conf Vomputer Vision and Pattern Recognition, CVPR'97, Puerto Rico, 1997

Heap 1997

Heap I , Hogg D , "Improving Specificity in PDMs Using a Hierarchical Approach", British Machine Vision Conference, 1997

Heckerman 1996

Heckerman D , "A Tutorial on Learning with Bayesian Networks", Microsoft Research Technical Report, MSR-TR-95-06

Heidemann 2000a

Heidemann G , Lucke D , Ritter H , "A System for Various Visual Classification Tasks Based on Neural Networks", Proc Int'l Conf Pattern Recognition, Barcelona, 2000

Heidemann 2000b

Heidemann G , Lucke D , Ritter H , "Segmentation of Partially Occluded Objects by Local Classification", Proc IEEE Int'l Joint Conf Neural Networks, IJCNN'00, Como, Italy, 2000

Hill 2000

Hill A , Taylor C J , Brett A D , "A Framework for Automatic Landmark Identification Using a New Method of Nonrigid Correspondence", IEEE Trans Patt Anal Mach Int , Vol 22, No 3, March 2000

Huang 2000

Huang C L , Wu M S , Jeng S H , "Gesture Recognition using the Multi-PDM Method and Hidden Markov Models", Image and Vision Computing, Vol 18, 2000, pp 865-879

Illingworth 1990

Illingworth P J , Kittler J , Yuen H K , "Shape Analysis Using Hough Transform", in Computer Vision and Image Processing, edited by A Barret, Chapman & Hall, 1990

Intille 1994

Intille S S , Bobick A F , "Disparity-Space Images and Large Occlusion Stereo", Third European Conf Computer Vision, Stockholm, 1994

Isard 1998a

Isard M , Blake A , "CONDENSATION – Conditional Density Propagation for Visual Tracking", Int'l Journal Computer Vision, Vol 29, No 1, 1998, pp 5-28

Isard 1998b

Isard M , Blake A , "A Smoothing Filter for CONDENSATION", Proc 5th European Conf Computer Vision, Vol 1, 1998, pp 767-781

Ishibuchi 1993

Ishibuchi K , Takemura H , Kishino F , "Real Time Hand Gesture Recognition using 3D Prediction Model", Int'l Conf Systems, Man, and Cybernetics, Le Touquet, France, October, 1993

Ivry 2003

Ivry R , Personal Communication, University of California, Berkeley, 2003

Jackson 2000

Jackson G M , Jackson S R , Husain M , Harvey M , Kramer T , Dow L , "The Coordination of Bimanual Prehension Movements in Centrally Deafferented Patient", Brain, Vol 123, 2000

Jain 1989

Jain A , Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs NJ, 1989

Jain 2000

Jain A , Duin R P W , Mao J , "Statistical Pattern Recognition A Review", IEEE Trans Patt Anal Mach Int , Vol 22, No 1, January 2000

Jelinek 1997

Jelinek F , Statistical Methods for Speech Recognition, Massachusetts Institute of Technology, MA, 1997

Jojic 1999

Kojic N , Turk M , Huang T S , "Tracking Self-Occluding Articulated Objects in Dense Disparity Maps", Proc IEEE Int'l Conf Computer Vision, Gorfu, Greece, September 1999

Jolliffe 1986

Jollifte, I T , Principal Component Analysis, Springer-Verlag, New York, 1986

Karpinski 1998

Karpinski M , Rytter W , Fast Parallel Algorithms for Graph Matching Problems, Oxford University Press, Oxford, 1998

Kennerley 2002

Kennerley S W , Diedrichsen J , Hazeltine E , Semjen A , Ivry R B , "Callasotomy Patients Exhibit Temporal Uncoupling During Continuous Bimanual Movements", Nature Neuroscience, 2002

Kohler 1997

Kohler M R J , "System Architecture and Techniques for Gesture Recognition in Unconstraint Environments", Int'l Conf Virtual Systems and MultiMedia, Geneva, September 1997

Lamar 1999

Lamar M , Bhuiyan Md Sh , Iwata A , "Hand Gesture Recognition using Morphological Principal Component Analysis and an Improved CombNET-II", Proc IEEE Int'l Conf System, Man, and Cybernetics, Tokyo, October 1999

Leavers 1992

Leavers E , Shape Detection in Computer Vision Using the Hough Transform, Springer-Verlag, 1992

Lee 1999

Lee H K , Kim J H , "An HMM-Based Threshold Model Approach for Gesture Recognition", IEEE Trans Patt Anal Mach Int , Vol 21, No 10, October 1999, pp 961-973

Li 2001

Li Y , Gong S , Liddell H , "Recognising Trajectories of Facial Identities Using Kernel Discriminant Analysis", Proc British Machine Vision Conference, BMVC 2001, Manchester, September 2001

Lin 1998

Lin D T , "Spatio-Temporal Hand Gesture Recognition Using Neural Networks", Proc IEEE World Congress on Computational Intelligence, 1998

Linde 1980

Linde Y , Buzo A , Gray R M , "An Algorithm for Vector Quantizer Design", IEEE Trans Communications, January 1980, pp 702-710

Lyons 1999

Lyons M , Budynek J , Akamatsu S , "Automatic Classification of Single Facial Images", IEEE Trans Patt Anal Mach Int , Vol 21, No 12, December 1999

MacLean 2001

MacLean J , Herpers R , Pantofaru C , Wood L , Derpanis K , Topalovic D , Tsotsos J , "Fast Hand Gesture Recognition for Real-Time Teleconferencing Applications", 2nd Int'l Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, Vancouver, July 2001

Mammen 2001

Mammen J P , Chaudhuri S , Agrawal T , "Simultaneous Tracking of Both Hands by Estimation of Erroneous Observations", Proc British Machine Vision Conference, 2001

Martinez 2001

Martinez A M , Kak A C , "PCA versus LDA", IEEE Trans Patt Anal Mach Int , Vol 23, No 2, February 2001, pp 228-233

McAllister 2002

McAllister G , McKenna S J , Ricketts I W , "Hand Tracking for Behaviour Understanding", Image and Vision Computing, Vol 20, 2002, pp 827-840

McKenna 1998

McKenna S J , Gong S , "Gesture Recognition for Visually Mediated Interaction using Probabilistic Event Trajectories", Proc British Machine Vision Conference, BMVC'98, Southampton, September 1998

Mechsner 2001

Mechsner F Kerzel D , Knoblich G , Prinz W , "Perceptual Basis of Bimanual Coordination", Nature, Vol 414, November 2001

Mechsner 2002

Mechsner F , "Why are We Particularly Good at Performing Symmetrical Movements", MaxPlackResearch, Januray 2002

Moghaddam 1998

Moghaddam B , Wahid W , Pentland A , "Beyond Eigenfaces Probabilistic Matching for Face Recognition", 3rd IEEE Int'l Conf Automatic Face and Gesture Recognition, Nara, Japan, 1998

Moghaddam 1999a

Moghaddam B , "Principal Manifolds and Bayesian Subspaces for Visual Recognition", Int'l Conf Computer Vision, ICCV'99, Kerkyra, Greece, 1999

Moghaddam 1999b

Moghaddam B , Pentland A , "Probabilistic Visual Learning for Object Detection", Pro 5th Int'l Conf Computer Vision, Cambridge, MA, June 1995

Morguet 1999

Morguet P , Lang M , "Comparison of Approaches to Continuous Hand Gesture Recognition for a Visual Dialog System", Proc of 1999 IEEE Int'l Conf Acoustics, Speech, and Signal Processing, ICASSP 99, Phoenix, 1999

Morrison 2003

Morisson K , McKenna S J , "An Experimental Comparison of Trajectory-based and History-based Representation for Gesture Recognition", 5th Int'l Gesture Workshop, Genova, Italy, April 2003

Murphy 1999

Murphy K P , Weiss Y , Jordan M I , "Loopy Belief Propagation for Approximate Inference An Empirical Study", 15th Conf Uncertainty in Artificial Intelligence, Stockholm, 1999

Nam 1996

Nam Y , Wohn K Y , "Recognition of Space-Time Hand-Gestures using Hidden Markov Models", ACM Symposium on Virtual Reality Software and Technology, Hong Kong, 1996

Neapolitan 1990

Neapolitan R E , Probabilistic Reasoning in Expert Systems Theory and Algorithms, John Wiley and Sons, USA, 1990

Ng 2000

Ng C W , Ranganath S , "Gesture Recognition via Pose Classification", Proc Int'l Conf Pattern Recognition, ICPR'00, Barcelona, 2000

Pavlidis 1977

Pavlidis T , Structural Pattern Recognition, Springer Verlag, New York 1977

Pavlovic 1997

Pavlovic V I , Sharma R , Huang T S , "Visual Interpretation of Hand Gestures for Human-Computer Interaction A Review", IEEE Patt Anal Mach Int Vol 19 No 7, July 1997

Pearl 1988

Pearl J , Probabilistic Reasoning in Intelligent Systems Network of Plausible Inference, Morgan Kaufmann Publishers, California, 1988

Pentland 2000

Pentland A , "Looking at People Sensing for Ubiquitous and Wearable Computing", IEEE Trans Patt Anal Mach Intell Vol 22, No 1, January 2000, pp 107-119

Persoon 1977

Persoon E , Fu K S , "Shape Discrimination using Fourier Descriptors", IEEE Trans Syst Man Cyber Vol 7, 1977

Petrou 2001

Petrou M , "Fusion of Information Fuzze Logic versus Bayesian Networks", EPSRC School in Computer Vision, University of Surry, UK, 2001

Pitas 1993

Pitas I , Digital Image Processing Algorithms, Prentie-Hall, New York, 1993

Ron 1996

Ron D , Singer Y , Tishby N , "The Power of Amnesia Learning Probabilistic Automata with Variable Memory Length", Machine Learning, Vol 25, 1996, pp 117-149

Ruiz 2001

Ruiz A , Lopez-de-Teruel P E , "Non-linear Kernel-Based Statistical Pattern Analysis", IEEE Trans Neural Networks, Vol 12, No 1, January 2001

Salvendy 2001

Slavendy G , "Human-computer Interaction", Ercim News, No 46, July 2001

Schlenzig 1994

Schlenzig J , Hunter E , Jain R , "Recursive Identification of Gesture Inputs Using Hidden Markov Models", Proc 2nd IEEE Workshop on Applications of Computer Vision, Sarasota, December 1994

Scholkopf 1996

Scholkopf B , Smola A , Muller K R , "Non-linear Component Analysis as a Kernel Eigenvalue Problem", Technical Report, Max Planck Institute fur biologische Kybernetik, December 1996

Scholkopt 1998

Scholkopf B , Mika S , Smola A , Ratsch G , Muller K R , "Kernel PCA Pattern Reconstruction via Approximate Pre-Images", Proc 8th Int'l Conf Artificial Neural Networks, Skovde, Sweden, 1998

Shamaie 1997

Shamaie A , Fakhraie S M , Benhabib B , Parallel Processing Techniques in Accurate Estimation of Partly Occluded-Ellipse Parameters in CIM Environments, M Sc Dissertation, University of Tehran, Tehran, 1997

Shamaie 1999

Shamaie A , Fakhraie S M , Benhabib B , "A New Transformation Technique for Accurate Estimation of Partly-Occluded Ellipse Parameters", Scientia Iranica, Vol 6, No 1, January 1999, pp 43-50

Shamaie 2001

Shamaie A , Sutherland A , "Graph-based Matching of Occluded Hand Gestures", Applied Imagery Pattern Recognition Workshop, Washington D C , 2001

Shamaie 2003

Shamaie A , Sutherland A , "Accurate Recognition of Large Number of Hand Gestures", 2nd Iranian Conf Computer Vision and Image Processing, Tehran, Iran, February 2003

Sharifi 2003

Sharifi S , Shamaie A , Crane M , "Noise in Correlation Matrix A Simulation Approach", 23rd Conf Applied Statistics in Ireland, Mullingar, Ireland, 2003

Sherrah 2000

Sherrah J , Gong S , "Resolving Visual Uncertainty and Occlusion through Probabilistic Reasoning", Proc British Machine Vision Conference, BMVC 2000, Bristol, September 2000

Sozou 1994

Sozou P D , Cootes T F , Taylor C J , Di-Mauro E C , "A Non-linear Generalisation of PDMs Using Polynomial Regression", British Machine Vision Conference, 1994

Sozou 1995

Sozou P D , Cootes T F , Taylor C J , Di-Mauro E C , "Non-linear Point Distribution Modelling Using Multi-layer Perceptron", British Machine Vision Conference, 1995

Starner 1995a

Starner T , Pentland A , "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models", M I T Media Lab Technical Report, No 375, MA, 1995

Starner 1995b

Starner T , Pentland A , "Visual Recognition of American Sign Language Using hidden Markov Models", Proc Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, June 1995

Starner 1996

Starner T , Weaver J , Pentland A , "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video", M I T Media Lab Technical Report, No 466, MA, 1996

Stauffer 1999

Stauffer C , "Automatic Hierarchical Classification Using Time-Based Co-occurrences", IEEE Conf Computer Vision and Pattern Recognition, CVPR'99, Fort Collins, Colorado, June 1999

Su 1998

Su M , Huang H , Lin C , Huang C J , Lin C D , "Application of Neural Networks in Spatio-temporal Hand Gesture Recognition", Proc IEEE World Congress on Computational Intelligence, 1998

Theodoridis 1999

Theodoridis S , Koutroumbas K , Pattern Recognition, Academic Press, London, 1999

Warakagoda 1996

Warakagoda N D , A Hybrid ANN-HMM ASR system with NN based adaptive preprocessing, M Sc Thesis, Norges Tekniske Hogskole, Institutt for Teleteknikk, Transmisjonsteknikk, Hungary, 1996

Weiss 1997

Weiss Y , "Belief Propagation and Revision in Networks with Loops", M I T Dept Brain and Cognitive Sciences Technical Report, No 155, November 1997

Weiss 1999

Weiss Y , Freeman W T , "Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology", University of California Berkeley Technical Report UCB/CSD-99-1046, June 1999

Weiss 2000

Weiss Y , "Correctness of Local Probability Propagation in Graphical Models with Loops", Neural Computation, Vol 12, 2000, pp 1-41

Wilson 1999

Wilson A , Bobick A , "Parametric Hidden Markov Models for Gesture Recognition", IEEE Trans Patt Anal Mach Int , Vol 21, No 9, September 1999, pp 884-900

Wolpert 2001

Wolpert M D , Ghahramani Z , Flanagan J R , "Perspectives and Problems in Motor Learning", TRENDS in Cognitive Sciences, Vol 5, No 11, November 2001, pp 487-494

Wu 2002

Wu H , Gesture recognition using principal component analysis, multi-scale theory, Ph D Thesis, Dublin City University, 2002

Yanez-Suarez 1999

Yanez-Suarez O , Azimi-Sadjadi M R , "Unsupervised Clustering in Hough Space for Identification of Partially Occluded Objects", IEEE Trans Patt Anal Mach Int , Vol 21, No 9, September 1999

Ying 1999

Ying Z , Castanon D , "Statistical Model for Occluded Object Recognition", Proc IEEE Int'l Conf Information, Intelligence and Systems, Bethesda, MD, November 1999

Zadeh 1992

Zadeh L A , Fuzzy Logic for the Management of Uncertainty, Wiley, New York, 1992

Zhou 2003

Zhou H , Huang T S , "A Bayesian Framework for Real-Time 3D Hand Tracking in High Clutter Background", 10th Int'l Conf Human-Computer Interaction, Crete, Greece, 2003

Zieren 2002

Zieren J , Unger N , Akyol S , "Hands Tracking from Frontal View for Vision-Based Gesture Recognition", Lecture Notes in Computer Science LNCS 2449, Springer, 2002 , pp 531-539

Zuech 2000

Zuech N , Understanding and Applying Machine Vision, Michael Dekker, New York, 2000