Gesture recognition in a smart room environment

by

Christiaan Coenraad Joubert Smit

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Computer Engineering)
Department of Electrical and Electronics Engineering
University of Pretoria

February 2001

Supervisor: Prof EC Botha
Co-supervisor: Dr GS Cox

Abstract

Much of our interaction with the environment is physical. We use our bodies for nonverbal expression or to augment or emphasize verbal communication. In other cases we use our bodies to execute tasks such as walking or picking up an object. A human observer can easily recognise these activities. For example, it is the job of a security officer in a supermarket to observe people and check that articles are not stolen. If a person does steal, the security officer recognises the act and takes appropriate action.

The problem addressed in this study is the automatic recognition of human gestures by means of video image analysis. For this purpose a computer-based system with similar recognition capabilities as a human observer is investigated. The system uses cameras that correspond to the eyes and algorithms that resemble abilities of the human visual system. Automatic gesture recognition is a complex problem and the focus here is to develop algorithms that will solve a subset of the problem. This involves the recognition of simple gestures such as walking and waving of arms.

The approach taken in this dissertation is to represent body shape in camera images with a simple model called a bounding box. This model has the appearance of a rectangle that encapsulates the extremities of the human body and resembles the coarse structure of body shape. From a representation point of view, the model is an abstraction of body pose. A gesture consists of a sequence of poses. By employing pattern recognition techniques, a sequence of pose abstractions is recognised as a gesture.

Various aspects of the bounding box model are explored in this study. Perception experiments are conducted to gain a conceptual understanding of the behaviour of the model. Other aspects include investigation of two- and three- dimensional spatial representations of the model with a neural network classifier as well as the model's temporal properties through the use of hidden Markov models. These aspects are tested using gesture recognition systems implemented for this purpose. The gesture vocabularies of these systems range from four to ten gestures, while recognition rates vary from 84.7% to 96.3%.

Keywords: human gesture recognition, computer vision, smart room, pattern recognition, neural network, hidden Markov model.

Uittreksel

Baie van ons interaksie met die omgewing is fisies. Ons gebuik ons liggame onder andere vir nie-verbale kommunikasie of om verbale kommunikasie te beklemtoon. In ander gevalle gebruik ons ons liggame om sekere bewegings mee uit te voer soos om te loop of om 'n voorwerp op te tel. Vir 'n mens is dit maklik om sulke aktiwiteite te herken. Byvoorbeeld, 'n sekuriteitsbeampte in 'n winkel hou mense dop om te kyk dat items nie gesteel word nie. Indien iemand wel iets steel, herken die sekuriteitsbeampte dit en neem die nodige aksie.

Die probleem wat in die studie ondersoek word, is die automatiese herkenning van menslike gedrag of aksies. Vir hierdie doeleinde word 'n rekenaar gesteunde stelsel ondersoek wat soortgelyke vermoëns het as dié van 'n mens. So 'n stelsel gebruik kameras wat die funksie van oë naboots en algoritmes wat die menslike visuele stelsel naboots. Outomatiese aksie herkenning is 'n komplekse probleem en daarom word net 'n substel daarvan hier ondersoek. Dit behels die herkenning van eenvoudige aksies soos loop and arms waai.

Die benadering wat hier gevolg word, is om die liggaamsvorm in kamerabeelde te modelleer met 'n eenvoudige model wat 'n omtrekreghoek genoem word. Die model is 'n reghoek wat die hele liggaam van die mens omsluit. So 'n model stel rofweg die vorm van 'n mens se liggaam voor en is dus 'n abstrakte voorstelling van liggaamshouding. 'n Aksie is 'n aantal opeenvolgende liggaamshoudings. Deur gebruik te maak van patroonherkenningstegnieke word die sekwensie van die abstrakte voorstellings herken as 'n aksie.

Verskillende aspekte van die omtrekreghoek model word in die studie ondersoek. Dit sluit persepsie eksperimente in wat die konsepsionele werking van die model beskryf. Ander aspekte wat ondersoek word is na twee- en drie- dimensionele ruimtelike voorstellings van die model met 'n neurale netwerk as klassifiseerder, sowel as die tyd aspekte van die model deur middel van verskuilde Markov modelle. Dié aspekte word ondersoek deur gebruik te maak van verskeie gedrags herkenning stelsels wat geïmplementeer was vir die doeleinde. Aksie woordeskatte van hierdie stelsels wissel van vier tot tien aksies, terwyl herkennigsvermoëns wissel van 84.7% tot 96.3%.

Sleutelwoorde: menslike aksie herkenning, rekenaar visie, intelligente omgewing, patroonherkenning, neurale netwerk, verskuilde Markov model.

Acknowledgements

I would like to thank the following people for their help and support, without which this project would not have been possible:

- My wife, Soria, for her support and motivation during the course of this work,
- Dr. Greg Cox for his valuable advice, suggestions and supervision,
- DebTech for sponsoring the work,
- Anton de Beer for the drawings in Figures 3.1(a), 4.1(a) and 5.1(a).
- Friends, family and fellow DebTech employees who kindly participated in the various experiments conducted during the course of this study,
- Prof. Liesbeth Botha for her quality inputs and supervision,
- Family for their support.

List of Abbreviations

2D	Two dimensions or two dimensional
3D	Three dimensions or three dimensional
CCD	Charge coupled device
CD	Compact disc
CV	Computer vision
DOF	Degrees of freedom
DTW	Dynamic time warping
FIFO	First-in-first-out
HVS	Human visual system
HMM	Hidden Markov model
MBB	Moving bounding box
MLD	Moving lights display
OI	Object of interest
RBF	Radial basis function
SFM	Structure from motion
SNR	Signal to noise ratio

List of Symbols

b	3D bounding box true centroid height (scalar, meters)
$\mathbf{b}$	3D bounding box true centroid position (3-element vector, meters)
h	2D and 3D bounding box height (pixels) and (scalar, meters) respectively
n	Image frame number
s	3D bounding box perimeter (scalar, meters)
w	2D bounding box width (pixels)
$\mathbf{x}$	3D bounding box lower centroid position (3-element vector, meters)
$\mathbf{y}$	3D bounding box intersection position of lower vertices (3-element vector, meters)

Contents

Chapter 1

Introduction

> "George is blissfully unaware that a crime is about to be committed right under his nose. Partially obscured by a bag of doughnuts and a half-read newspaper is one of the dozens of security monitors he is employed to watch constantly for thieves and vandals. On the screen in question, a solitary figure furtively makes his way through a car park towards his target. The miscreant knows that if the coast is clear it will take him maybe 10 seconds to get into the car, 15 to bypass the engine immobiliser and 10 to start the engine. Easy. But before he has even chosen which car to steal, an alarm sounds in the control room, waking George from his daydream. A light blinking above the screen alerts him to the figure circling in the car park and he picks up his radio. If his colleagues get there quickly enough, they will not only catch a villain but also prevent a crime.
>
> The unnatural prophetic powers of the security team would not exist but for some smart technology. The alarm that so rudely disturbed George is part of a sophisticated visual security system that predicts when a crime is about to be committed." - *Warning! Strange behaviour* - New Scientist, 11 December 1999.

The above extract describes a surveillance system that automatically detects suspicious human behaviour in a security context. Although the system is a fictitious one, advances in computer technology and pattern recognition during the past three decades have brought it closer to reality. The research field that studies this and related technologies is known as computer vision. In this dissertation computer vision is investigated for the purpose of automatically recognising human gestures such as walking, waving and crouching in camera images. A system capable of detecting gestures can be used as a building block for the scenario sketched above.

1.1 Motivation

Research in the automatic recognition of gestures can be motivated by its diverse applications. Example applications are:

- **Safety and security**: The introductory paragraph described a security application. A related application is safety, where the system detects human activities that might have health and safety implications. The system can even attempt to predict such activities and initiate preventative actions.
- **Sports training**: The performance of athletes can be improved by means of gesture analysis. In such an application the motion of an athlete's body is reconstructed and presented as a computer graphic model. The athlete's trainer analyses the reconstruction and recommends improvements in technique if necessary. An example is javelin throwing, where body motion is analysed to optimise the javelin's angle of attack and velocity.
- **Natural man-machine interfaces**: The original man-machine interface to a computer is the keyboard. This was later augmented with pointing devices such as the mouse and tracker ball. Pointing devices provided a more natural interface than a keyboard and simplify navigation in graphical user interface environments. During the 1990's speech technology matured. Commercial software packages appeared on the market and offer the best natural interface to computers so far. A great deal of interaction with our world is by means of vision. A truly natural man-machine interface should therefore integrate visual information as well.
- **Medical applications**: By analysing the gait of a patient, location and movement of joints are tracked and analysed for abnormalities. From this analysis corrective procedures (e.g. physiotherapy) can be recommended.

The above examples are only a few potential applications, but are sufficient to warrant research in this field.

1.2 What is computer vision?

Computer vision plays an important role in the previously discussed examples. These systems use computer vision techniques to function and is also central to the work conducted in this study. A definition of computer vision is therefore appropriate. Trucco and Verri [1, p. 2] define computer vision as a set of computational techniques aimed at estimating or making explicit the geometric and dynamic properties of the 3D world from digital images. Wechsler [2, p. 19] calls it the process of seeking to produce useful descriptions of visual input to allow an artificial or natural system to safely negotiate its environment. Schalkoff [3, p. 2] simply calls it the science (or art?) of making robots 'see'.

A typical computer vision setup consists of one or more cameras interfaced to a computer. Images are captured and digitized by the cameras and suitable hardware and processed by the computer. The computer outputs a description of the scene - the content of which depends on the application. In this study the captured images contain humans executing gestures and the computer outputs the gesture type or class. The process of achieving this is the focus of this dissertation.

1.3 Objectives

Machine-based gesture recognition is a difficult problem. In this study a novel and simple model is proposed as an approach to the problem. This model is dubbed the "bounding box" model. Its purpose is to abstract gesture information in the form of body pose, which is one of the first steps required for gesture recognition. Within this framework the objectives are:

- To investigate the bounding box model by means of visual perception experiments. The outcomes of these experiments provide a conceptual understanding of the capabilities of the bounding box model.

- To investigate the performance of the proposed model in a machine-based gesture recognition scenario by employing computer vision and pattern recognition techniques. This includes:

  - **Two dimensional (2D) representation**: The human body is modeled as a 2D spatial representation. This is the simplest form of the proposed bounding box model and makes a number of assumptions. It is, however, a good starting point to test the concept.

  - **Three dimensional (3D) representation**: Here the 2D model is extended to a 3D spatial representation. This is a better approximation of the human body and therefore has more practical value.

  - **Temporal properties**. A temporal model is investigated capable of modeling the temporal properties of the bounding box.

1.4 Contributions

Contributions made in this study are in the field of computer vision and automatic gesture recognition. They are:

- **Bounding box model**: In this study the bounding box model is proposed, investigated, implemented and tested in a gesture recognition system. It is a simple model that can be applied in a scenario where the aim is to recognise *coarse* human gestures. The parameters of this model are simple to determine and it offers an alternative to complex models often encountered in the literature. It was found that the model works well in a 2D and 3D environment. The strength of the model lies in the extension from a 2D to 3D representation. The parameters of many 3D models are difficult to calculate. In contrast, calculation of the 3D bounding box parameters are of a comparable complexity to that of the 2D bounding box.

- **Recognition strategy**: A simple recognition strategy based on first principle component approximation is used to classify gestures. This technique is used in the 2D and 3D recognition systems and permits real-time and continuous recognition of gestures. A second technique that uses hidden Markov models (HMMs) is also used for recognition. It is more complex than the first principle component approximation scheme, but makes better use of the temporal information contained in the gesture signatures.

1.5 Overview

To conclude this chapter, an overview of the subsequent chapters is given. Chapter 2 gives a review of the literature applicable to machine-based gesture recognition. This is followed by an introduction of the bounding box model in Chapter 3. A conceptual discussion of the model is given and its usefulness investigated by means of a visual perception experiment. Given the insight gained from this investigation, the proposed model is tested in a machine-based system in Chapter 4. A simple scenario is chosen, namely a 2D system and a vocabulary of only four gestures. The 2D system limits motion to a plane and is of limited practical value. The aim is, however, to obtain an initial estimate of the proposed model's performance in a machine-based application. Chapter 5 extends the recognition system from 2D to 3D and therefore removes the plane motion constraint. In addition the gesture vocabulary is extended to a total of eight gestures. Chapter 6 improves further on recognition capabilities by introducing an alternative temporal model. The bounding box model is rich in temporal information, which has up to this point not been properly exploited. The alternative temporal model is a hidden Markov model and it is tested for ten gestures using the 3D bounding box model. This dissertation is concluded with Chapter 7, which puts this work into perspective and recommends future research.

Accompanied with this dissertation is a CD that contains computer viewable video sequences (see inside of front cover). The videos are in the AVI video file format and contain demonstrations of the various recognition systems developed during the course of this study.

To conclude, a note on the meaning of the words "gesture" and "pose" used in this dissertation: A gesture is the execution of body motion to accomplish a task (e.g. walking and waving of hands). It requires the movement of the body and limbs. When reference is made to "gesture", it is implied that the gesture contains motion. A gesture is made up of a sequence of "poses", which are motionless or static. A gesture itself can also be static, for example standing still, and this is referred to as a "pose". Depending on the context of a sentence, "gesture" can be a collective noun for both moving and static gestures (that is gestures and poses). Since this can be confusing the phrase "dynamic gesture" is sometimes used to emphasize that only gestures containing motion are implied.

Chapter 2

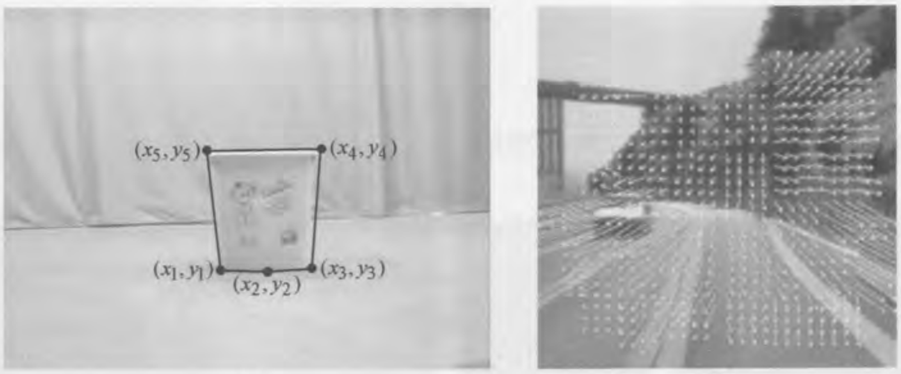
Literature overview and background

In Section 1.2 of the previous chapter the computer vision problem was formulated as follows: Given one or more camera images, the aim is to describe the content of the images within the context of a particular problem. Images are presented as a sequence of digitised pixel values. In its raw form an image is unsuitable for computer vision interpretation. The question is: How does one proceed to make sense of raw image data to produce a meaningful interpretation? This chapter reviews the literature that addresses this question in the context of machine-based gesture recognition.

The application scope of computer vision is large. The introductory section discusses the current paradigm that deals with the general computer vision problem (Section 2.1). This is followed by a discussion of motion recognition, where the aim is to recognise the movement type of an arbitrary object in an image sequence (Section 2.2). The main body of this chapter overviews a special case of motion recognition, namely gesture recognition (Section 2.3).

2.1 The current computer vision paradigm

Vision is of interest to two groups of scientists: Neuropsychologists and psychophysicists study biological vision with the aim of understanding how it works. The second group, engineers and computer scientists, study vision in order to develop vision systems [4]. It is hoped that if biological vision is understood, the problem of computer vision can also be solved. Despite decades of research, understanding biological vision is still in its infancy [2, p. 493]. The same is true for computer vision [4, 5]. This can be illustrated by the classical problem of invariant object recognition - a requirement for a robust



(a) (b)

Figure 2.1: Example models to represent various targets. Figure (a) shows a chain code model suitable to represent rigid objects. Figure (b) shows an optical flow field that represents motion (after Trucco and Verri [1, p. 255]).

vision system [1, p. 255][2, pp. 95-96]. From a computer vision point of view it is a complex problem, although many examples of successful biological vision systems are in existence (e.g. the human visual system (HVS)). Invariant object recognition is a topic often discussed in the literature (e.g. Welcher [2, Chapter 3] and Würtz [6]), but to date no accurate model or solution exists.

According to Cedras and Shah [7, p. 130] a computer vision system needs to perform the following two steps :

- **Representation**: The first step transforms the target[1] in one or more images to a suitable model. The model is determined beforehand and the basis for choice is often the application. For example, a system aiming to recognise an arbitrary shape might choose a chain-code representation as illustrated in Figure 2.1(a). If the aim is to detect moving objects, a motion model such as the one in Figure 2.1(b) is perhaps more suitable. The model serves as abstract representation of the information to be interpreted by the system. An important property of such a representation is that it should be easy to manipulate by computer. For example, a chain-code representation describes the shape of an object as a list of nodes, which is suitable for manipulation. To summarise: the representation phase extracts information from the images that is meaningful to the problem.
- **Recognition**: Once the model parameters are known, the target (e.g. object

[1] A target does not need to be a physical object. As discussed soon, motion can also be abstracted and therefore reference is made to a target instead of object.

motion or object type) is recognised or interpreted. This usually means that the target is classified as one of many known classes and this is achieved by means of standard pattern recognition techniques. Not all machine vision applications require a recognition phase. The sports training example given in Section 1.1 does not require recognition. In that case it serves as an analysis tool and interpretation is done by the athlete's trainer.

Much of computer vision research focuses on the representation phase of the problem and the infrastructure to facilitate it [1, 3]. It is regarded as a complex problem that depends on a number of properties such as the target, it's environment, the application's objectives and the model chosen.

2.2 Motion recognition

A subset of computer vision problems is concerned with the analysis and recognition of objects in motion. For example, a meteorological vision system tracking a tropical cyclone from satellite images might calculate its speed and rate of expansion.

Motion recognition has been influenced by visual perception experiments such as the work of Johansson [8, 9]. Johansson attempted to uncover the mechanisms used by the human visual system to recognise motion. His discoveries were based on experiments he dubbed the Moving Lights Display (MLD). In these experiments, an actor was dressed in black and light bulbs attached to his joints. He was then placed in a dark room to ensure his body shape was invisible. The actor was then asked to perform a number of gestures such as walking and dancing. An audience unfamiliar with the experiment was asked to identify the gestures or motion. They managed to easily identify moving or dynamic gestures, even if the gestures were complex (e.g. dancing). However, stationary gestures (resulting in a stationary lights display) or upside-down gestures were perceived as meaningless [10]. Johansson concluded that motion information directly contributes to its recognition.

Interpretation of the MLD experiments can lead to two motion recognition approaches [7, p. 129]:

- **Configuration-based recognition**: First, the object's structure is recovered and used to recognise what the object is. Once this is achieved, motion is recognised by how it moves.

- **Motion-based recognition**: Here the object type and its motion are recognised by the characteristics in its motion. Structure plays little or no role in this process

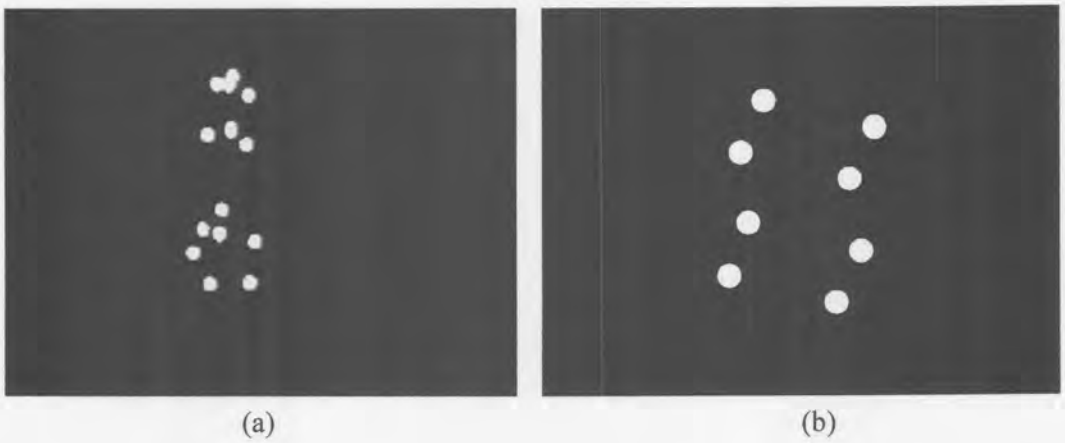


Figure 2.2: Example of MLD representations. It consists of blobs located at strategic locations on the structure of an object. The static MLD objects shown in Figures (a) and (b) are difficult to recognise. Once these objects are in motion, the object and motion type can be recognised (see accompanied video sequences).

- a direct inference from the MLD work. Figure 2.2(a) and 2.2(b) show objects presented within the framework of the MLD experiment - lights or blobs are placed on strategic locations on a structure, while the remainder of the structure is invisible. The reader is invited to identify these objects. Most people cannot identify the objects, since they lack structure [8, 9]. By observing how the blobs or lights move it is, however, possible to identify the objects and their motion. This is demonstrated by watching the two video sequences[2] that shows these objects in motion. The videos are located in the *chap2* directory of the CD that accompanies this document.

The fundamental property of motion-based recognition is that a motion field must be present - the object must be moving over the period of observation. It is assumed that the manner in which it moves is unique to the object and if recognised, identifies the object and the motion type. Configuration-based recognition adds an additional complexity, namely that of structure. Recovering it might not be easy, but it allows the recognition of motionless activities or poses, for example a person sitting or the

[2] The video sequence named *mld_2.avi* was compiled from images scanned in from Johannson's publication on his MLD work [9]. Unfortunately only every sixth frame was published. This resulted in the video sequence being temporally subsampled. A clue to help the observer is that two objects are present in the sequence. Johannson claims that naive observers recognise the activity in 0.2 seconds.

orientation of a static cube.

Gesture recognition systems are usually biased towards one of the above two approaches. The bounding box model that is investigated in subsequent chapters can be considered to fall in the configuration-based category: It recovers the coarse structure of the human body and recognition is then based on the recognition of the signatures of the model parameters. Details of gesture recognition systems that can be found in the literature are given in the next chapter.

2.3 Gesture recognition

The framework discussed in the previous section is generic and applicable to motion recognition of any object. Gestures are performed by humans and to recognise them, properties of the human body have to be taken into account. Two important properties are that the human body is nonrigid and body parts can be occluded. This excludes direct application of motion recognition theory that assumes rigidness. Fortunately the human body has unique properties that are often used to aid in object modeling [11]. One such property is that limbs are constrained in movement or that they have to adhere to certain kinematics [12, 13]. For example, the elbow can only bend approximately 180 ° and in one degree of freedom (DOF), which in turn constrains movement of the hand. By taking such properties into account the state space[3] is decreased, making the problem easier to solve [14].

In Section 2.1 it was pointed out that a machine vision system requires two steps, namely representation and recognition. Section 2.2 revealed that to recognise motion two techniques can be used, namely a configuration-based or motion-based approach. This section investigates gesture recognition within this framework. Configuration-based and motion-based representations applicable to gesture recognition are discussed first. This is followed by techniques that recognise gestures given one of these two representations.

2.3.1 Configuration-based models

This model type represents the structure or configuration of the human body (see Gravila[11] for an overview). Representation is based on either explicit shape information (appearance models) or low level features such as points or lines (feature models). An appearance model resembles the human body in appearance and the objective is

[3] This is the space that defines all possible configurations of the object.

to register the model to the person in the image. Feature models ignore high level shape information and instead seek out low level features and track these from frame to frame. It requires solving for feature correspondence between successive frames. Correspondence is implicit for appearance models, since it is implied by registration[4].

The first step of configuration-based modeling is segmentation. Here, the person is masked from the background in the camera images. Strategies to accomplish this vary widely in the literature. Often, the problem is simplified by using a special chroma-keying background to enforce a high foreground/background contrast [16, 17, 18]. Another approach is to use special markers on the clothing to simplify location of features [19]. Sophisticated segmentation techniques are also used to improve segmentation. For example, the Pfinder system of Wren *et al.* [20] exploits the spatio-temporal properties of the human body in order to track and segment it.

The bounding box model presented in this study also requires a segmentation procedure before it can be constructed. A background substraction segmentation scheme that utilises chroma-keying is chosen for this purpose. Details of the algorithm are given in Chapter 4.

Appearance models

Appearance models use atomic components to build a representation that resembles the human body. These components are typically sticks, contours or volumes and are related to the human body [21]: Sticks resemble the skeleton, contours the body projection in camera images and volumes the 3D detail of the human body. The complexity of model to image registration is determined by the number of model parameters - a high DOF model is usually more difficult to register. The remainder of this section overviews some literature published in this regard.

Chen and Lee [13] used a 17 segment stick figure model shown in Figure 2.3. The objective of their work was to recover 3D pose of a simulated stick figure, given a sequence of perspective projected images of the stick figure. They formulated the problem using a graph, where the optimal graph is the solution to the pose at a given time instant. The graph is pruned by using knowledge of gestures. A first phase eliminates candidate graphs that do not comply to instantaneous poses. For instance, two arms can not be both in front or behind the torso simultaneously. The second phase pruned the graph to a single one by requiring that the movement of joints should be

[4] Correspondence is the process of locating the same feature of an object in successive images or locating the same feature in images shown from different viewpoints. Registration is the process of computing the transformations which bring those corresponding features into alignment [15].

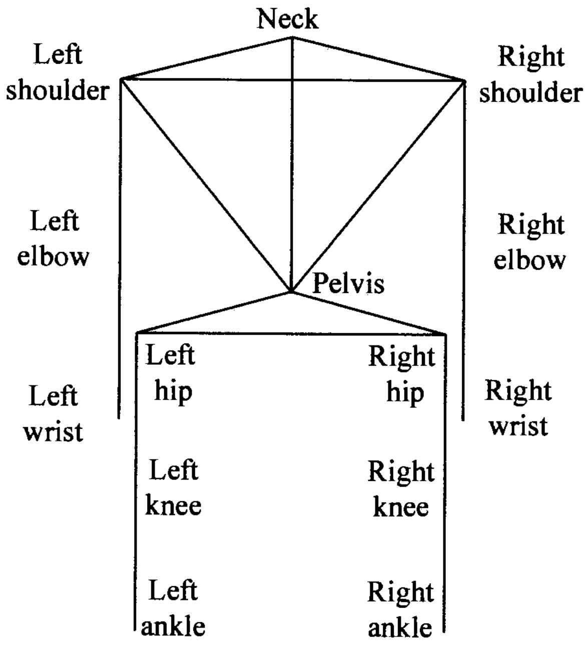


Figure 2.3: A appearance stick figure model. This model was used by Chen and Lee [13] to recover 3D gestures of a simulated figure.

smooth from one frame to the next. This approach requires the length of the line segments and the location of the neck to be known.

Rohr [22] used a 14 object volumetric model to represent the human body. A similarity measure is defined based on the geometric properties of the volumetric model's contours and the detected edges of the person in the camera image. The measure takes edge length, projection angle and the distance between the model's contour and image edge into account. Matching is accomplished by maximising the similarity measure. To minimise the search space, model pose in future frames is predicted by means of Kalman filtering. Gavrila and Davis [23] used a 22 DOF super-quadrics model and a similar measure for matching. However, their system required the person to wear a tight-fitting suit. The suit had colour coded limbs to aid in model registration.

Rehg and Kanade [14] used kinematic constraints to simplify matching of a 27 DOF human hand model to a hand observed in two stereo camera views. Their system, DigitEyes, requires the hand to be in a known initial pose. Once this condition is satisfied hand gestures are tracked by using local features on the hand and applying the above constraint. This system has applications in sign language recognition.

As stated earlier the appearance models resemble the actual appearance of the

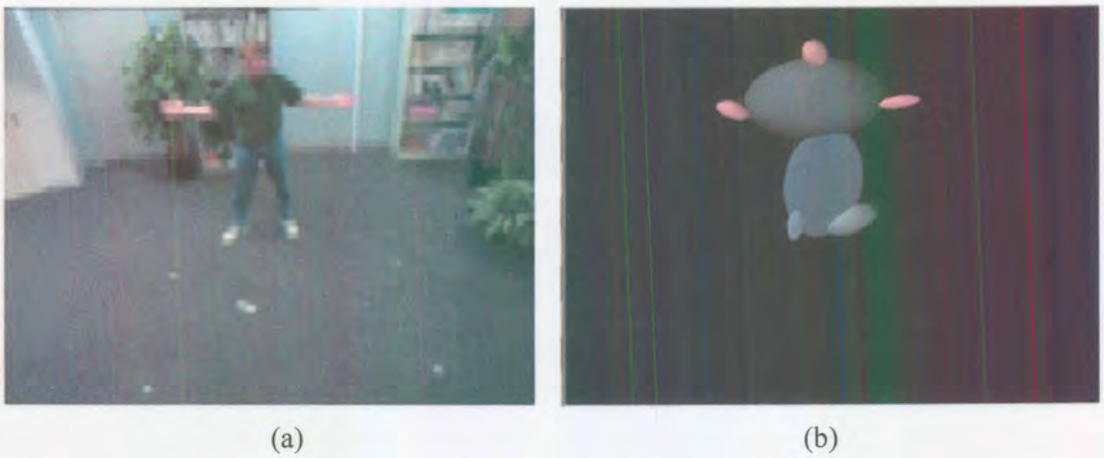


Figure 2.4: The Pfinder system breaks features on the person down into blobs. Features are based on colour and pixel location and tracked in successive frames using a Kalman filter (after Wren *et al.* [20]).

human body. This type of model uses *a priori* information about the object to be modeled (in this case the human body) and the configuration needs to be determined beforehand. The bounding box model actively models the shape of the human body. Models closer related to the bounding box are described in the next section.

Feature models

Feature-based models do not use *a priori* information about the structure, but instead model the human body based on low level features. Popular features are derived from shape, location and colour. For example, Wren *et al.* [20] grouped image pixels belonging to the person based on colour and location in their Pfinder system. The result was a combination of blobs that coarsely represents the structure of a human body (see Figure 2.4). One of the primary functions of this system is the blob correspondence from frame to frame. This was accomplished by using Kalman filtering that estimates the location of the blobs in the next frame. The Kalman filter learns the dynamics of the blobs by taking their history into account.

Yamato, Ohya and Ishii [24] avoided the feature correspondence problem by dividing the segmented image into a static array of M by N blocks. Array entries were set equal to the foreground to image pixel ratio. A feature vector was then constructed that represents the pose of a person at a given time instant. This technique was used to recognise various tennis strokes [24].

Darrell and Pentland [25] and Darrell, Essa and Pentland [26] describe a method where a gesture is defined as a predetermined number of key frames. Key frames are camera images of the actual gesture that serve as templates for the gestures. The idea is that a full gesture is represented by interpolating between key frames. In essence a gesture in a test sequence is recognised by determining the similarity between the test sequence and the template sequence. Similarity is measured using a normalised correlation-based metric.

Instead of using the camera images directly, Murase [27] extracted image variation by means of principle component analysis. Pose is then represented by the lower order eigenvectors obtained from this method. Image features can also be derived using wavelets [28] and splines [29].

The above models model body pose using shape, location, colour or a combination of these features. The proposed bounding box model is closely related to this group of models - it attempts to describe shape by enclosing the body of the person in a camera image in the smallest possible rectangle or bounding box. The result is a coarse representation of body shape. The bounding box is calculated for every frame and is static with respect to time for poses, but active for dynamic gestures. To recognise a gesture a system that uses this type of model should therefore recognise the sequence of bounding box representations.

2.3.2 Motion-based models

In Section 2.2 it was mentioned that motion is a prerequisite for motion-based recognition. Gesture types that can be represented by this model therefore need to be dynamic. This is the case for many gesture recognition systems, for example human activity surveillance, lip reading recognition and sign language recognition. Although motion-based recognition is not directly applicable to the proposed bounding box model, it is often discussed in the gesture recognition literature and therefore worthwhile to mention.

A motion-based representation transforms temporal events into a spatial representation. A good example is the binary motion region images by Davis [30]. This approach monitors consecutive image frames for change. Any pixel that deviates from its previous value is flagged and represented as a white pixel as shown in Figure 2.5 for a person sitting. The final image (bottom right figure of Figure 2.5) shows the accumulated change of the gesture sequence and is the abstraction of the gesture.

A problem with motion-based approaches is that it is difficult to segment the rep-

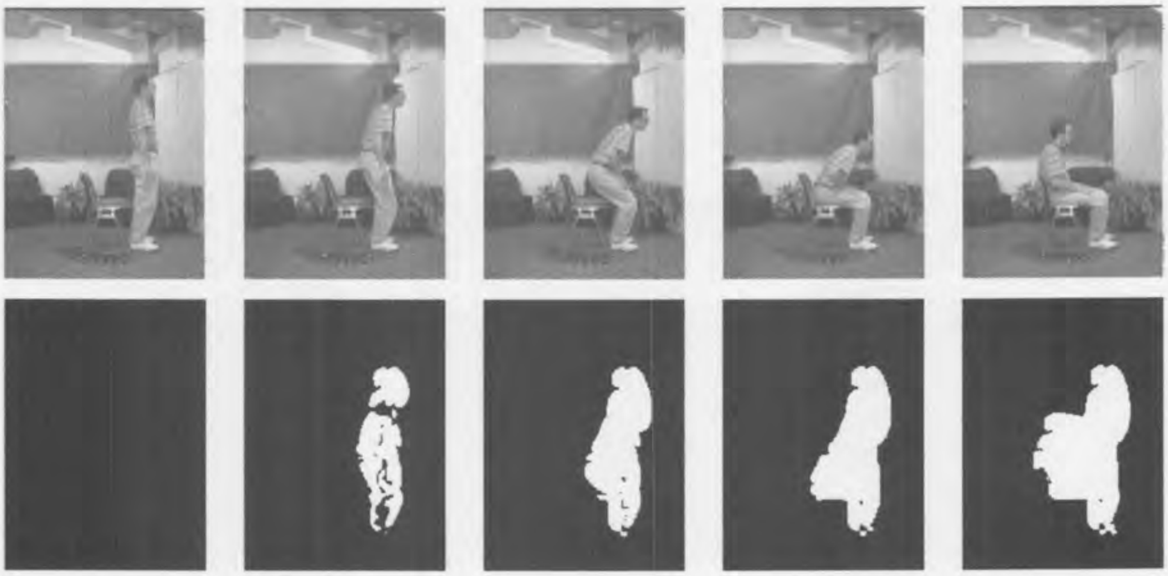


Figure 2.5: A motion model that accumulates changes in an image sequence. The top sequences shows a person sitting and the bottom the accumulated change in pixel values. Davis [30] called it binary motion region images. The final image is an abstraction of the gesture.

resentation and determine the start and stop instances of the motion calculation. If the calculation period is too short only part of the gesture is represented in the model, or if it is too long it might contain more than one gesture. Polana [31] and Polana and Nelson [32] solved this for cyclic activities (e.g. walking, running and jumping jacks): Activity is analysed by calculating motion in a grid laid over the person. A motion value is based on the change of each cell value in the grid with respect to previous values. Cycles in an activity are detected by searching for re-occurrences in the sequence.

Another type of motion representation is XT-slices [33, 34]. A line (X-slice) is chosen in the image sequence such that it intersects the person at a region of interest. A motion image (XT-slice) is then built by placing the X-slices of successive images in sequence (e.g. top to bottom). For example, Figure 2.6(a) shows two example images of a person walking from right to left. The X-slice is chosen to be located at the ankles of the person and the resulting XT-slice of the sequence is shown in Figure 2.6(b). The resulting pattern - in this case a crisscross pattern - can also be used for detecting periodic motion [33] or for gait recognition [34].

Human motion can also be presented by optical flow fields [35] [36, pp. 17-37][36,

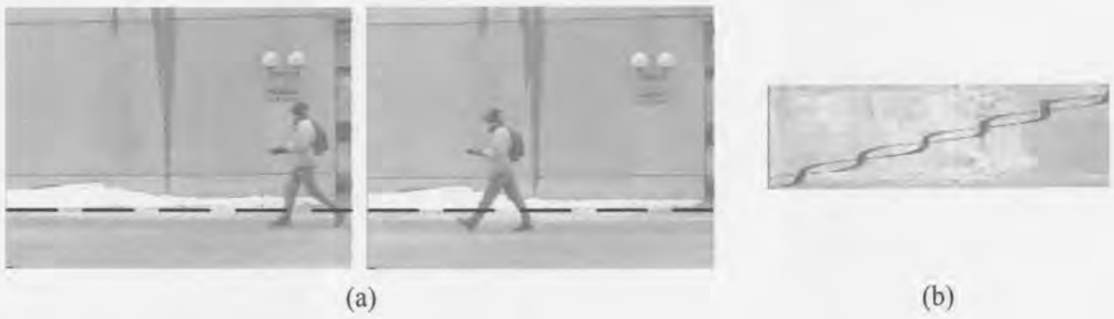


Figure 2.6: Example of an XT-slice motion model. Figure (a) shows two frames of a sequence containing a person walking. The dashed line shows the location of the X-slice. Figure (b) is the resulting XT-slice compiled by placing the X-slices in sequence (after Lui [33]).

pp. 245-269]. The aim here is to track scene-based features (e.g. edges or corners) or image-based features (e.g. image intensities) in consecutive frames. Given that the feature correspondence from frame to frame is successfully solved, a displacement or instantaneous velocity field is calculated. An example for a moving car is shown in Figure 2.1(b). This optical field characterises activity in the scene.

2.3.3 Recognition

Recognition aims to classify an unknown gesture representation as one of a set of known classes. The known classes represent prototype gestures and are compiled from a predetermined gesture vocabulary. Recognition is based on motion-based or configuration-based features of the representation. Methods used for recognition are commonly found in the pattern recognition literature and include neural networks, hidden Markov models (HMMs) and template matching techniques. Details of these methods are given in the remainder of this section.

Polana [31] and Polana and Nelson [32] used a feature vector that reflects motion in a period of cyclic gestures. This was achieved by adding all the motion values of corresponding motion cells in the detected gesture period. They then used a nearest centroid classifier to recognise the unknown feature vector as one of seven gestures. Davis [30] based a feature vector on Hu shape moments as descriptors for motion in his binary motion region images. The Mahalanobis distance is calculated between the feature vector and each gesture class. The class with the shortest distance is then chosen.

The above two recognition techniques collapse temporal information observed over a fixed period into a single feature vector. Configuration-based models often produce a sequence of features over a period of time, which correspond to snapshots of gestures or poses. Recognition can therefore be achieved by detecting a sequence of snapshots or features. A property of gestures is that they can be executed at various rates. This can be a problem and the recognition method must deal with it. Two popular techniques that address this problem are dynamic time warping (DTW) and hidden Markov models (HMM).

DTW is a matching technique commonly encountered in earlier speech recognition literature [37, 38]. It matches an unknown pattern to a prototype or reference pattern by warping the unknown pattern to fit the reference pattern (asymmetric warping) or by warping both the unknown and reference pattern (symmetric warping). Darrell, Essa and Pentland [26] used DTW to temporally align the correlation scores of a novel sequence to that of a known template sequence. The endpoints of the feature sequence were kept fixed while starting points were elastically matched. The gesture belonging to the template model with the shortest accumulated distance is chosen to be the executed gesture. Other work where DTW was applied for gesture recognition is by Takahashi *et al.* [39] and Bobick and Wilson [40].

Another recognition method that has become popular in gesture recognition is the use of hidden Markov models [41, 42]. An HMM represents the stochastic properties of an observation sequence by means of a Markov random process. The Markov random process is hidden and is indirectly observed through features of the process. Yamato, Ohya and Ishii [24] were the first to employ HMMs for the purpose of gesture recognition. As discussed earlier, they used a feature vector constructed based on the foreground pixel ratio obtained from a fixed grid. A gesture consists of a sequence of feature vectors calculated from the camera images. For each gesture an HMM is generated by a training procedure that requires a number of example gesture sequences. To determine the class of an unknown sequence, the probability of the sequence belonging to each of the models is calculated. The assigned gesture is the one with the largest probability. HMMs have also been used for gesture recognition by Pentland and Liu [43], Starner and Pentland [44], Wilson and Bobick [45] and Vogler and Metaxas [46].

A bounding box based gesture recognition system also requires a recognition phase. Features are derived from the bounding box model, which serves as input to the classifier. Two types of classifiers are used for this purpose namely neural networks and HMMs.

2.4 Conclusion

This chapter highlighted the various elements required for a gesture recognition system. Section 2.1 stated that the system should have a representation and recognition stage. According to Section 2.2 gesture abstraction models can be based on the configuration of the object or its motion. Configuration-based models can again be divided into appearance and feature based models. The next chapter discusses the bounding box model, which is the key element of the gesture recognition system discussed in this study. This model can be categorised as a configuration-based model, specifically a feature-based model. It will be shown in the next chapter that the bounding box model represents the coarse structure of the human body. The second ingredient of a gesture recognition system is the recognition phase. In this study two techniques are investigated, namely recognition based on a neural network (Chapters 4 and 5) and a more advanced HMM based classifier (Chapter 6).

Chapter 3

The bounding box model

The literature survey of the previous chapter revealed two important paradigms used for machine-based gesture recognition, namely motion-based [7, 30, 31, 32, 33, 34, 35, 36] and configuration-based [11, 13, 22, 23, 14, 20, 24, 26, 27, 29, 28] approaches. This chapter describes a very simple configuration-based model that forms the basis of the remainder of this study.

The chapter starts by introducing this model and relating it to existing models (Section 3.1). Being simple, it has limitations. These are then investigated by a visual perception experiment (Section 3.2). From the results of this experiment insight can be gained into the capabilities of the proposed model (Section 3.3). Once these are understood, subsequent chapters use the model to abstract gesture information in various machine-based gesture recognition systems.

3.1 Bounding box model overview

Configuration-based approaches use models that represent the structure of the actual object and therefore have the same physical appearance as the object. Figure 3.1(a) shows a typical model of the human body. This model consists of a number of geometrical shapes (circles and rectangles) configured such that it looks like a human. The objective is to register the model, in other words relating the elements or shapes of the model to their counterparts in the body of the person (see Figure 3.1(b)). This is usually not a simple problem: Registration algorithms do not always converge to the correct solution, typically due to image noise, occlusion of body parts and the fact that the model is only an approximation of the real body shape [47, 48, 17].

In this dissertation a very simple configuration-based model is proposed - the bound-

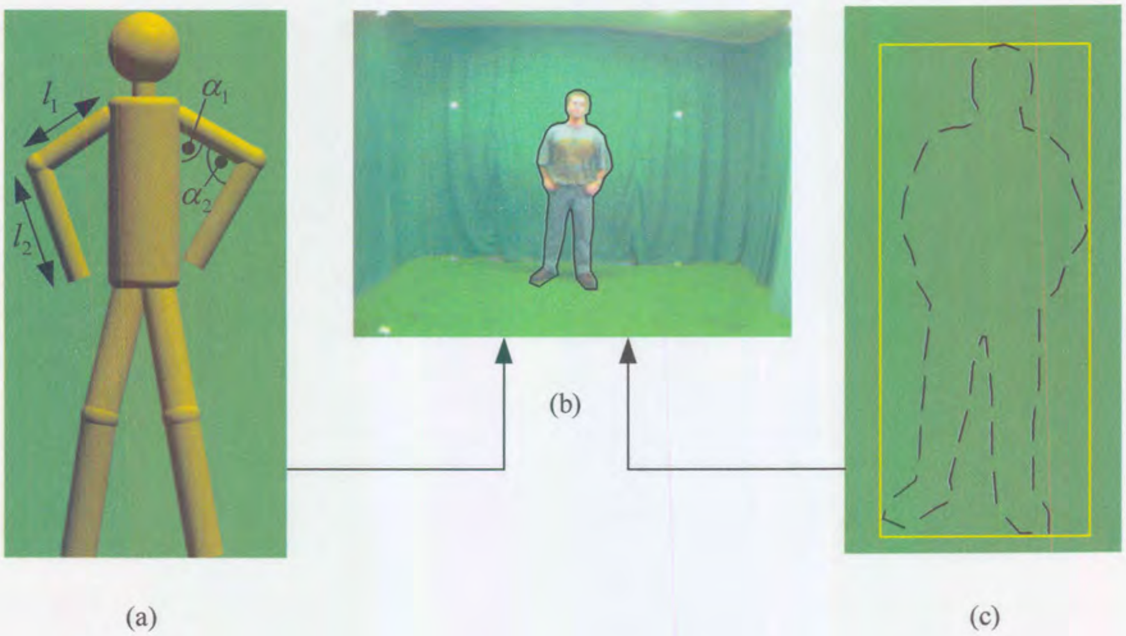


Figure 3.1: Two possible abstractions of a person's body pose. The model in (a) is a high degree of freedom (DOF) model typically found in the literature (only a few parameters are shown). The model in (c) is the proposed bounding box model. In both cases the problem is to register the model to the person in (b).

ing box[1]. This type of model consists simply of a rectangle that encapsulates the structure of the person. Figure 3.2 shows a number of body poses and their bounding box representations. For each pose the corresponding bounding box appears different. It seems to contain some information about the pose and can perhaps be used as a model in a recognition system. The advantage of such a model is that it is simple to calculate: Given that the person is located or segmented in an image, (s)he is encapsulated within the smallest possible rectangle.

The human body has a very large number of possible poses or degrees of freedom (DOF). DOF directly translates to the number of independent parameters that uniquely represent body pose. This means that all parameters need to be specified to determine a body pose[2]. The bounding box has only four parameters, namely centroid

[1] For the purpose of this discussion only a 2D bounding box is considered.

[2] Some parameters are unique and fixed for a particular person (e.g. the length of upper arm) and other parameters are constrained, for example, under normal conditions the elbow has a limited angle range.

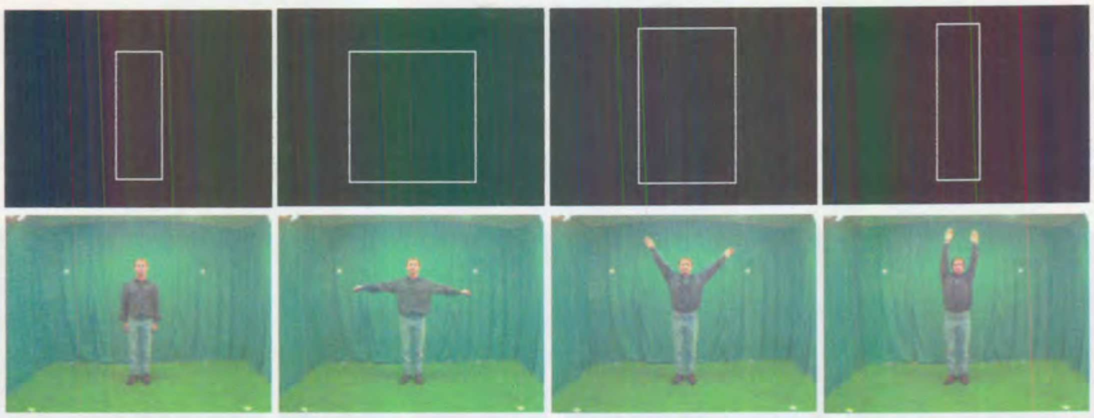


Figure 3.2: A person executing different arm poses is shown in the bottom sequence. The corresponding bounding boxes are shown in the top sequence. The bounding box appears different for each pose suggesting that it contains some information about pose.

coordinates (x, y), width and height (w, h). A bounding box representation therefore reduces the number of parameters and the exact body pose can therefore not be recovered from it. In other words, information is lost during conversion to the bounding box presentation. Visually this can also be demonstrated: It is not possible, for example, to determine where the hands of the person in Figure 3.3 are by only observing the bounding box. This is in contrast to a high DOF appearance model such as the one in Figure 3.1 that is capable of representing the above poses.

An important question arises from the above discussion: To what extent can a bounding box represent actual body pose? Furthermore, is this model of any use in a machine-based gesture recognition system? These questions are addressed in the remainder of this chapter.

3.2 Moving bounding box experiment

Humans have a remarkable ability to recognise complex visual patterns [49, p. 34][32]. Recognition of visual patterns is an important topic in the cognitive sciences and a number of theories attempt to explain the processes involved [49, pp. 34-42]. Many of these theories are based on observations of human participants in cognitive experiments in an attempt to understand this ability. Ideas for a computer vision system are often first tested on human participants (e.g. Cedras [7, p. 129] and Davis [30, pp. 6-8]). The assumption is that if humans cannot recognise a certain visual pattern, then most

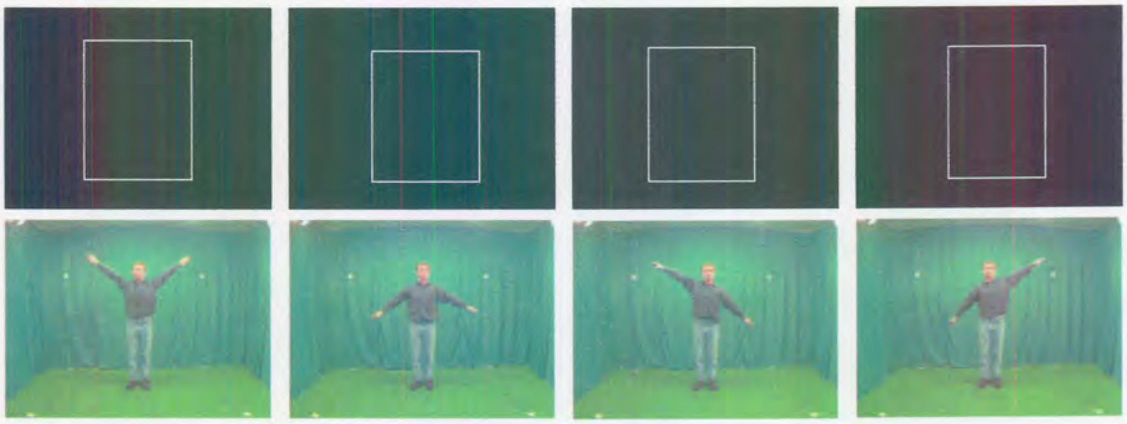


Figure 3.3: Various arm poses are shown in the bottom sequence. Their corresponding bounding boxes are in the top sequence. It is easy to determine arm pose from the bottom sequence, but not the top sequence.

probably, neither will a machine-based system.

The above approach was also used in this study to determine the usefulness of the bounding box model. A bounding box has a characteristic signature or behaviour for a particular gesture. For example, if a person crouches, it has a predominantly vertical stretching and shrinking motion combined with a slight horizontal stretching and shrinking motion. If a person, unfamiliar with this work recognises the gesture by only observing the bounding box, then a bounding box might be an acceptable model for crouching. The following sections describe an experiment, dubbed the "Moving Bounding Box" or MBB experiment. It was conducted to explore the possibilities and limitations of a bounding box as model through human perception. Although observations from such an experiment cannot be accepted as ground truth, it does give some insight into the potential performance of a machine-based system.

3.2.1 Objectives

The MBB experiment investigates the notion of the bounding box to represent gestures within the context of human visual perception. In other words the objective is to determine whether a bounding box contains sufficient visual information to identify various gestures. This investigation is conducted by keeping in mind that the application is a computer vision-based one. Interesting psychological conclusions can perhaps also be made from the experimental observations, but none are attempted here.

For the purpose of this experiment, gestures have been chosen that are assumed

Exp. dims.	Wk	Wv	Cr	Hd	In	Nd	Total gestures
2D	1	1	1	2	1	2	8
3D	1	1	1	2	2	1	8

Table 3.1: Account of the gestures in the 2D and 3D test sets. Legend is Wk —*walking*, Wv — *arms waving*, Cr — *crouching*, Hd — *hand waving*, In — *inactive*, Nd — *nodding*.

to be recognisable by a machine-based system using a high DOF model. These are *walking*, *arms waving*, *hand waving*, *crouching*, *inactivity* and *nodding*. The gestures are presented as two primary sets: A 2D bounding box set and a 3D bounding box set. The 2D set is applicable to a 2D gesture recognition system (later implemented in Chapter 4) and the 3D set applicable to a 3D gesture recognition system (Chapter 5). For the 2D set, motion is limited to a plane perpendicular to the camera's optical axis. For example, *arms waving* was recorded by waving at the camera. Doing this, one forces the problem (modeling of human body) to take on the same dimensionality as the representation (the 2D bounding box) [22]. No movement restrictions apply to the 3D set.

Researchers often investigate only 2D models, since it is simpler than the equivalent 3D representation. We are living in 3D space and therefore a practical model should consider 3D representations as well. Fortunately, it is easy to extend a 2D bounding box model to a 3D one and for this experiment both are investigated.

3.2.2 Experimental protocol

Eleven participants took part in the experiment, all of whom had no prior knowledge of the project. Participants were in the age group 21 to 61. The experiment was conducted on three levels - each level gives more information about the gestures than the previous level.

Four sets of computer viewable videos were compiled of the experiment's six gestures. The video sets are supplied on the accompanying CD and they are:

- A test set consisting of eight videos representing gestures with a 2D bounding box. This is referred to as the 2D test set and can be viewed in *chap3\2d\test\*. The key of this test set are given in file *2d_key.txt* located in *chap3\2d\test\*.

- A test set consisting of eight videos representing gestures with a 3D bounding box. This is referred to as the 3D test set and can be viewed in *chap3\3d\test\*.

The key of this test set are given in file *3d_key.txt* located in *chap3**3d**test*\.

- An example set consisting of six videos showing a person executing one of the six gestures as well as its 2D bounding box representation. This is referred to as the 2D example set and can be viewed in *chap3**2d**example*\.
- An example set consisting of six videos showing a person executing one of the six gestures as well as its 3D bounding box representation. This is referred to as the 3D example set and can be viewed in *chap3**3d**example*\.

All videos are approximately 20 seconds long. Some gestures are repeated in the test sets to prevent participants from consciously or subconsciously determining answers by a process of elimination. An account of the test sets are given in Table 3.1.

The experimental procedure is as follows:

- The participant is given a basic overview of machine-based gesture recognition and its objective. This is done without revealing any details about the gesture types.
- An explanation about 2D and 3D bounding box models is given and how it is being used in machine-based recognition. To familiarise the participant with the concepts and presentation, he or she is shown a few of the 2D and 3D test set videos.
- The following information is then given to the participant:
  - A single person is in the video.
  - Only one gesture is performed for the duration of a particular video.
  - The person in the video does not interact with any object.
  - A gesture can be repeated in another video of the same set.
  - *Unknown* is also a valid answer.
- The following three experiments are then conducted:
  - **Experiment 1**: The participant is asked to classify the 2D test set and his or her answers are noted. The participant does not know the gesture set at this point. This is repeated for the 3D test set.

- **Experiment 2**: The participant is told that there are six gestures and what they are. These gestures are also visually demonstrated by acting the gestures (the videos are not shown). The participant is asked again to classify the 2D and 3D test sets and his or her answers are noted.
- **Experiment 3**: The participant is shown the 2D and 3D example sets. The participant is asked for a third time to classify the 2D and 3D test sets and his or her answers are noted.

3.2.3 Results

In the case of Experiment 1 the gestures of the test set were unknown to participants. Answers can therefore be any gesture type and not necessarily from the test set. To be able to compare the results of the three experiments, answers were interpreted as belonging to one of the six classes in the case of Experiment 1. The criteria used for this classification was: Is the answer given by a participant for a particular gesture a reasonable description of the actual gesture? If the answer is yes, then it was assumed that the perceived gesture was recognised as the actual gesture.

The experimental results of the three experiments are given in the form of confusion matrices (Tables 3.2 to 3.7) and the overall recognition rate is then summarised in Table 3.8. Abbreviations used in these tables are Wk — *walking*, Wv — *arms waving*, Cr — *crouching*, Hd — *hand waving*, In — *inactive*, Nd — *nodding*, Un — *unknown*, Or — *other* (none of the above). Detailed answers of each participant are also given in Appendix A.

A number of observations were also made while conducting the experiments. The observations were independent of the representation dimensionality (2D or 3D) and are:

- Participant response time for Experiment 1 was usually more than 10 seconds irrespective of the gesture type,
- Response times for Experiment 2 and 3 were:
  - *Walking*, *arms waving* and *crouching* response times were around 2 to 5 seconds,
  - *Nodding*, *hand waving* and *inactive* response times were usually more than 10 seconds,

Particpant	True Gesture Class					
Answers (%)	**Wk**	**Wv**	**Cr**	**Hd**	**In**	**Nd**
Wk	**81.8**	0.0	0.0	0.0	0.0	0.0
Wv	0.0	**27.3**	0.0	0.0	0.0	0.0
Cr	0.0	0.0	**90.9**	0.0	0.0	0.0
Hd	0.0	0.0	0.0	**0.0**	9.1	0.0
In	0.0	0.0	0.0	40.9	**45.5**	63.6
Nd	0.0	0.0	0.0	0.0	0.0	**4.6**
Un	0.0	0.0	0.0	18.2	9.1	13.6
Or	18.2	72.7	9.1	40.9	36.3	18.2

Table 3.2: Confusion matrix of the 2D test set of Experiment 1.

Participant	True Gesture Class					
Answer (%)	**Wk**	**Wv**	**Cr**	**Hd**	**In**	**Nd**
Wk	**100.0**	0.0	0.0	0.0	0.0	0.0
Wv	0.0	**18.2**	0.0	0.0	0.0	0.0
Cr	0.0	0.0	**100.0**	0.0	0.0	0.0
Hd	0.0	0.0	0.0	**0.0**	0.0	0.0
In	0.0	9.1	0.0	68.2	**59.1**	63.6
Nd	0.0	0.0	0.0	0.0	0.0	**0.0**
Un	0.0	0.0	0.0	13.6	18.2	9.1
Or	0.0	72.7	0.0	18.2	22.7	27.3

Table 3.3: Confusion matrix of the 3D test set of Experiment 1.

- Participants often remarked that *nodding*, *hand waving* and *inactive*, looked alike and were difficult to recognise.

3.2.4 Discussion

A glance at Table 3.8 reveals that *walking*, *arms waving* and *crouching* were easily recognised given that the gesture vocabulary is known (Experiments 2 and 3). The remaining three gestures have low recognition rates. For these particular gestures, the question is whether the participants guessed the answer or actually recognised the gesture. Guessing means that the bounding box model does not contain the required

Participant	**True Gesture Class**					
Answer (%)	**Wk**	**Wv**	**Cr**	**Hd**	**In**	**Nd**
Wk	**100.0**	0.0	0.0	0.0	0.0	0.0
Wv	0.0	**100.0**	0.0	4.6	0.0	0.0
Cr	0.0	0.0	**100.0**	0.0	0.0	0.0
Hd	0.0	0.0	0.0	**31.8**	63.6	40.9
In	0.0	0.0	0.0	31.8	**27.3**	36.4
Nd	0.0	0.0	0.0	22.7	0.0	**18.2**
Un	0.0	0.0	0.0	9.1	9.1	4.5

Table 3.4: Confusion matrix of the 2D test set of Experiment 2.

Participant	**True Gesture Class**					
Answer (%)	**Wk**	**Wv**	**Cr**	**Hd**	**In**	**Nd**
Wk	**100.0**	0.0	0.0	0.0	0.0	0.0
Wv	0.0	**100.0**	0.0	0.0	0.0	0.0
Cr	0.0	0.0	**100.0**	0.0	0.0	0.0
Hd	0.0	0.0	0.0	**54.5**	45.4	9.1
In	0.0	0.0	0.0	27.3	**36.4**	36.4
Nd	0.0	0.0	0.0	18.2	18.2	**54.5**
Un	0.0	0.0	0.0	0.0	0.0	0.0

Table 3.5: Confusion matrix of the 3D test set of Experiment 2.

Participant	**True Gesture Class**					
Answer (%)	**Wk**	**Wv**	**Cr**	**Hd**	**In**	**Nd**
Wk	**100.0**	0.0	0.0	0.0	0.0	0.0
Wv	0.0	**100.0**	0.0	0.0	0.0	0.0
Cr	0.0	0.0	**100.0**	0.0	0.0	0.0
Hd	0.0	0.0	0.0	**27.3**	54.5	27.3
In	0.0	0.0	0.0	22.7	**27.3**	45.4
Nd	0.0	0.0	0.0	40.9	18.2	**18.2**
Un	0.0	0.0	0.0	9.1	0.0	9.1

Table 3.6: Confusion matrix of the 2D test set of Experiment 3.

Participant	**True Gesture Class**					
Answer (%)	**Wk**	**Wv**	**Cr**	**Hd**	**In**	**Nd**
Wk	**100.0**	0.0	0.0	0.0	0.0	0.0
Wv	0.0	**90.9**	0.0	0.0	0.0	0.0
Cr	0.0	9.1	**100.0**	0.0	0.0	0.0
Hd	0.0	0.0	0.0	**40.9**	27.3	18.1
In	0.0	0.0	0.0	22.7	**40.9**	36.4
Nd	0.0	0.0	0.0	36.4	31.8	**36.4**
Un	0.0	0.0	0.0	0.0	0.0	9.1

Table 3.7: Confusion matrix of the 3D set of Experiment 3.

Exp. #	**Dims.**	**Gesture Recognition Rate(%)**					
		Wk	**Wv**	**Cr**	**Hd**	**In**	**Nd**
1	2D	81.8	27.3	90.9	0.0	45.5	4.6
2	2D	100.0	100.0	100.0	31.8	27.3	18.2
3	2D	100.0	100.0	100.0	27.3	27.3	18.2
1	3D	100.0	18.2	100.0	0.0	59.1	0.0
2	3D	100.0	100.0	100.0	54.5	36.4	54.5
3	3D	100.0	90.9	100.0	40.9	40.9	36.4

Table 3.8: Gesture recognition rate for the set of MBB perception experiments.

information to abstract the particular gesture and is of little or no use in a machine-based recognition system.

For the above reason the possibility that a participant guessed the answer for *hand waving*, *inactive* or *nodding* is explored. This is determined by inspecting the confusion matrices of Experiments 2 and 3 (Tables 3.4 to 3.7). It seems that if any of these three gestures are presented to the participant, the chosen answer is also from this set. In a few cases *unknown* was also given as an answer, but this is ignored for the purpose of the following discussion. Therefore, if a participant was guessing, his or her chances of guessing correct is $\frac{1}{3}$ assuming equal probability. In the 2D case of Experiments 2 and 3 the recognition rate is below $\frac{1}{3}$ and one can infer that the participants have guessed the answers. Results are slightly better for the 3D case. Recognition rates for Experiments 2 and 3 are above $\frac{1}{3}$ suggesting that a 3D bounding box does contain some information about these gestures. One should, however, also keep in mind that the participant sample is small, which causes noise in the results. The recognition rate is, however, still too low for many practical computer vision applications. The low recognition rate (2D and 3D cases) is also supported by participant remarks that these gestures are difficult to recognise as well as their long response times to these gestures.

The results of Experiment 1 are also very interesting. Many participants recognised *walking* and *crouching* in the 2D and 3D cases without any prior knowledge of the gestures. *Arms waving* was, however, often mistaken as *star jumps*. *Inactive* has a fairly high recognition rate, because many participants classified *hand waving*, *inactive* and *nodding* as *inactive*. Once participants were informed about the gesture types, its recognition rate fell. This again illustrates that participants cannot distinguish between these three gestures.

Of more importance for Experiment 1 is the fact that participants were able to distinguish between some of the gestures. *Walking*, *arms waving* and *crouching* were always described as different gestures. In other words, they are distinctive. They are also distinctive from *hand waving*, *inactive* and *nodding* as a group. The significance of this is discussed in the next section.

A final remark about the trends in the results. The recognition results are the worst for Experiment 1 where the participant has the least information. This is the case for the 2D and 3D bounding box representations. For this experiment the confusion matrices show that participants are often unsure what the gesture type is (see Tables 3.2 and 3.3): A significant percentage of gestures are classified as *unknown* or *other*. Coarse gestures do however have a higher recognition rate than the fine gestures. Once the participant was informed about the gesture types as required for Experiment 2, the coarse gestures have a 100% overall recognition rate. The fine gesture recognition rate still under performs and gestures are mostly confused with each other (see Tables 3.4 and 3.5). Even when the maximum information about the gestures is given by showing example videos (Experiment 3), participants cannot recognise the fine gestures. Again these gestures are confused for each other as shown in Tables 3.6 and 3.7.

3.3 Conclusion

Given the results and from the above discussion the following conclusions are drawn:

- *Walking*, *arms waving* and *crouching* are modeled well by the bounding box as opposed to *hand waving*, *inactive* and *nodding*. The first set of gestures all have large movements in common as opposed to the latter set, which have little or no movement. We refer to large movement gestures as coarse gestures and small movement gestures as fine gestures. Given the results, it seems that coarse gestures are better recognised within the context of a bounding box model. One should, however, be careful not to generalise since this was only investigated for three gestures. For example, in Experiment 1 participants have often mistaken *arms waving* for *star jumps*. Both of these gestures fall into the coarse gesture category.

- The results of Experiment 1 have shown that coarse gestures are distinctive. Although a participant did not always recognise the actual coarse gesture, (s)he has never confused different coarse gestures. Also, coarse gestures are distinct from fine gestures. This observation can be used by a computer vision system to do unsupervised labeling of unknown gestures. Such a system, might for example, group all the fine gestures in a cluster and different types of coarse gestures in separate clusters.

- Experiment 3 suggests that the proposed model can also be used in a supervised labeling environment. In this experiment the human participant is first “trained” by presenting him or her with example gestures. The participant is then “tested” by being asked to classify an unknown gesture. This is the same procedure used in many pattern recognition systems.

- Experiment 2 also falls within the supervised labeling category. Of more importance here is that it indicates whether participants correctly interpreted the videos and the appearance of the bounding box. We as humans are familiar with the six gestures from experience and can visualise these gestures. The fact that *most* of the gestures can be recognised in the videos from historical training implies that the representation is adequate for the purpose of the experiment. Reference is made to *most* of the gestures, since the recognition rate of Experiment 2 should be viewed within the context of the results of Experiment 3. Experiment 3 has already shown that fine gestures cannot be recognised except in classifying them as a fine gesture. The fact that the fine gestures also have a low recognition rate in Experiment 2 can therefore be ignored when motivating the adequateness of the representation.

- The 2D and 3D bounding box representations relate respectively to 2D and 3D recognition systems. A 3D system extends the dimensionality of the event space, thereby removing some of the limitations of a 2D system. In this experiment both representations have similar recognition rates suggesting that one should expect similar performance of the two systems.

This chapter introduced the bounding box to abstract gestures. The quality of this model was explored by means of a perception experiment. An analysis of the experiment's results revealed that the proposed model has promise in automatic gesture recognition. In particular its strength is in the recognition of coarse gestures in a 2D or 3D environment. In the subsequent chapters the model is applied in various gesture recognition systems. The simplest case - the 2D representation - is investigated in the next chapter. A 3D system is then explored in Chapter 5. The systems developed in Chapters 4 and 5 use a simple temporal model to represent the temporal characteristics of gestures. In Chapter 6 a more advanced model is investigated that better utilises the temporal information offered by a sequence of bounding boxes.

Chapter 4

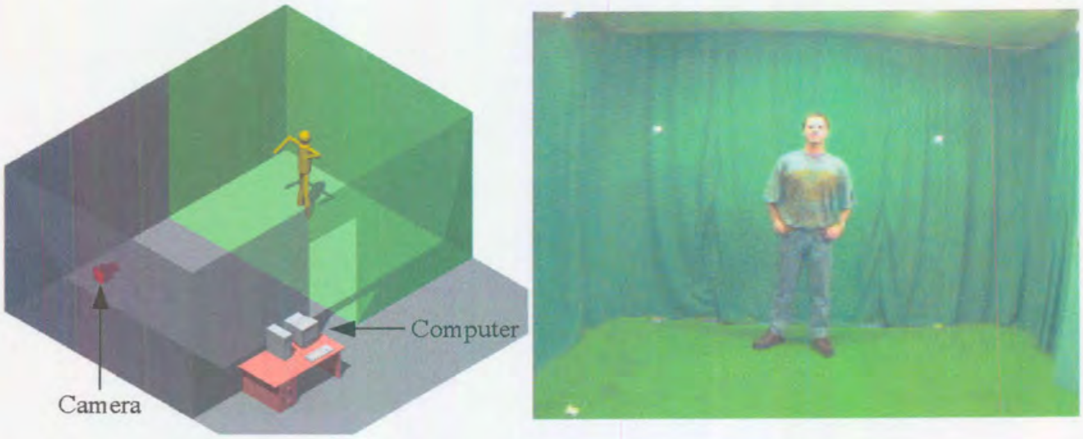
Coarse 2D gesture recognition

The Moving Bounding Box (MBB) experiment of the previous chapter suggested that humans find coarse gestures fairly distinctive in a bounding box framework. It also suggested that given this model, fine gestures are hard to recognise. Based on this observation, a computer vision system is described in this chapter that aims to automatically recognise gestures by utilising the bounding box model.

The objectives of the proposed machine-based gesture recognition system are given in Section 4.1. An outline of the approach to the problem is given in Section 4.2. Due to the complexity of the problem a number of assumptions have to be made. These are given in Section 4.3. This is followed by a discussion of the proposed system in Section 4.4. The system was tested by classifying gestures of a number of participating people. Results of the tests are given in Section 4.5 and discussed in Section 4.6. The chapter is concluded with Section 4.7 by putting the results into perspective.

4.1 Objectives

The aim of this chapter is to explore the bounding box model for abstracting gestures as required for automatic gesture recognition (see Section 2.2). From an implementation point of view, the 2D bounding box is simpler than the 3D bounding box. For this reason it is chosen as model for the first machine-based gesture recognition system of this study. A small gesture vocabulary set is also chosen, in particular the gestures that were identified by the MBB experiment to be easily recognisable. These gestures are *walking*, *waving*, *crouching* and *fine gesture.*



(a)

(b)

Figure 4.1: The setup for a machine-based gesture recognition system is shown in (a). A camera is connected to a computer which observes an area of activity. The computer runs an algorithm designed to recognise human gestures. (b) is a typical camera image. The green chroma-keying material aids with the segmentation process.

4.2 Approach

The literature survey of Chapter 2 highlighted two important steps required for motion recognition. First, the target as observed in an image needs to be abstracted. The abstraction is expressed as a predetermined model. The parameters of this model usually have some characteristic temporal signature related to a particular behaviour of the object. The second step is to recognise the actual behaviour of the object by recognising the characteristic signature of the model. Often, this is achieved by means of some pattern recognition algorithm trained to recognise the signature.

The object to be abstracted here is the shape of the human body. The model chosen to represent it is a 2D bounding box. Bounding box parameters behave in correspondence to the gestures executed by the person[1]. For the purpose of the gesture recognition system proposed in this chapter, recognition is based on a simple concept: Over a time period each gesture causes a dominant variation in one or more bounding box parameters. For example, arms waving has a dominant variation in bounding box width. By detecting this, a gesture is recognised. A detailed description of this technique is given in Section 4.4.3.

[1] This occurs only within the capabilities of the bounding box model.

Figure 4.1 shows the setup of the system. A camera is mounted halfway between the ceiling and floor in one side of the room. It faces an activity area covered mostly by green chroma-keying material[2]. The camera is connected to a frame grabber in a computer. An algorithm is executed on the computer designed to recognise certain gestures performed by the person in the activity area.

4.3 Assumptions

The objective of many computer vision systems are ambitious, since they aim at mimicking the human visual system. Currently, computer vision capabilities are not nearly as advanced and generalised in application as the human visual system. It is therefore imperative to make assumptions in order to make such systems viable. This system is no exception and the following assumptions are made or constraints imposed:

- **2D motion**: Motion is limited to a plane perpendicular to the optical axis of the camera. This is the same restriction that was applied to the 2D bounding box of the MBB experiment.

- **Gesture vocabulary**: Gestures are limited to *walking*, *arms waving* and *crouching*. A fourth gesture class called *fine gesture* is a collective class for all fine gestures. All other gestures should be classified as *unknown*.

- **Single foreground object**: Only a single foreground object[3] is allowed within the field of view of the camera. It is assumed that this object is a person. A single foreground object also implies that the person does not interact with any other objects, e.g. picking something up.

- **Environment**: To simplify image segmentation, the physical environment is carefully controlled. The room is well lit and kept at a constant light level. Care is also taken to ensure that surfaces are nonspecular and excessive cast shadowing is minimised. Good colour contrast between the person in the scene and background is ensured by using chroma-keying material and paint that covers the walls and floor. Finally, the person is never occluded by any object.

[2] Chroma-keying is a technique used in the motion picture industry to segment people or objects from the background in camera images. Once segmented, the people or objects are merged with another background, creating the impression that they are at the location of the background. Usually the chroma-keying colour is green or blue.

[3] This is also called an object of interest (OI). All other objects or surfaces are called background objects.

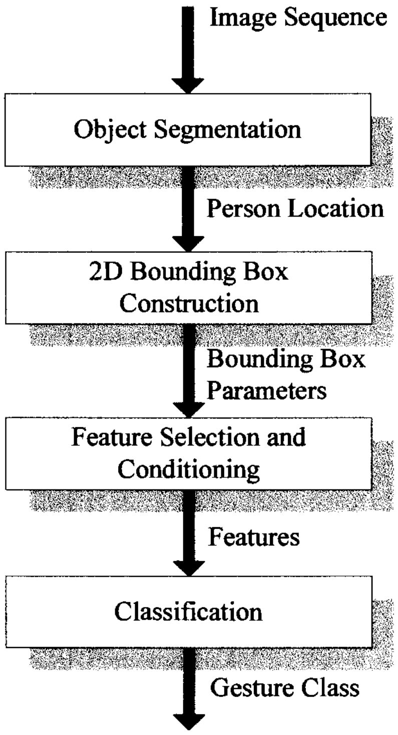


Figure 4.2: The gesture recognition algorithm is shown here as a high level flow diagram. It receives a sequence of images, processes these images and displays a gesture according to the algorithm.

Enforcing the above allows one to focus on the problem at hand - the machine recognition of gestures. This is, however, only a starting point for a more practical system and the aim is to gradually eliminate the above constraints by means of further research. Also considering the status quo of systems with similar objectives, the above constraints are not unreasonable. Some machine-based recognition systems found in literature often implicitly assume these constraints [19, 48].

4.4 Gesture recognition in 2D

In the following sections a machine-based 2D gesture recognition algorithm is discussed. Figure 4.2 shows a high level description of the algorithm. A person is first located in the activity area by segmenting the camera image. Next, a 2D bounding box is constructed that approximates the person's body pose. The third step is to select and condition features from the bounding box. Classification is then based on the value of the feature vector.

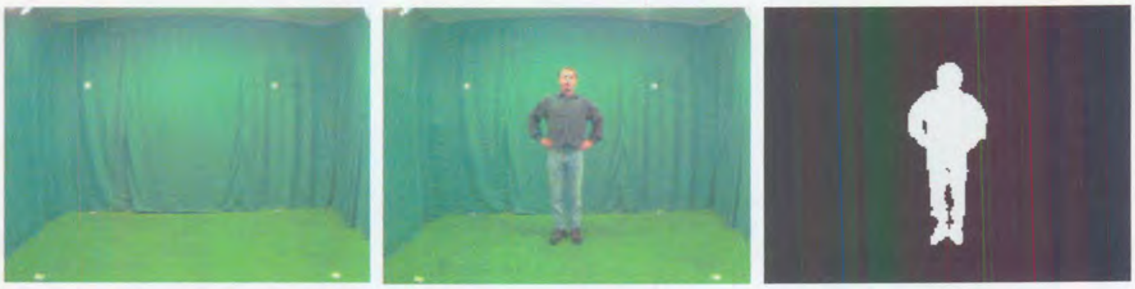


Figure 4.3: This figure shows how the person is segmented in an image. Each pixel in the active image (b) is compared to the respective pixel in the background image (a). A significant deviation as governed by a hypothesis test results in a foreground pixel and ultimately an object being detected. This is illustrated in (c).

4.4.1 Segmentation

Segmentation is the process of locating or isolating an object of interest (OI) in an image. Generally, it is a complex problem [15, p. 413] [50]. Successful segmentation depends on many factors such as the scene environment and its dynamics. The emphasis of this work is gesture recognition. It was therefore decided to implement a simple, but adequate segmenter by controlling the environment. A short conceptual description of this segmenter is given next.

One approach to the segmentation problem is to employ a background substraction scheme [51, 52]. It is an attractive approach if the camera is stationary in space (free of ego-motion) and background images are available[4]. A stationary camera always observes the same spatial location in a scene. OIs are identified by simply comparing an active image to a background image[5]. If the pixel value deviates significantly from its corresponding background value it is labeled as being part of an OI. The camera used here is fixed in space and background images are easily obtainable. It was therefore decided to implement the above segmentation strategy.

The above approach suffers from the problem that environmental factors such as changing light levels, specular surfaces and cast shadows can cause active pixels to sig-

[4] Even if background images are unavailable a background generation scheme can be employed to estimate the background [53].

[5] An active image is one that might contain an object of interest in it. A background image never contains such an object.

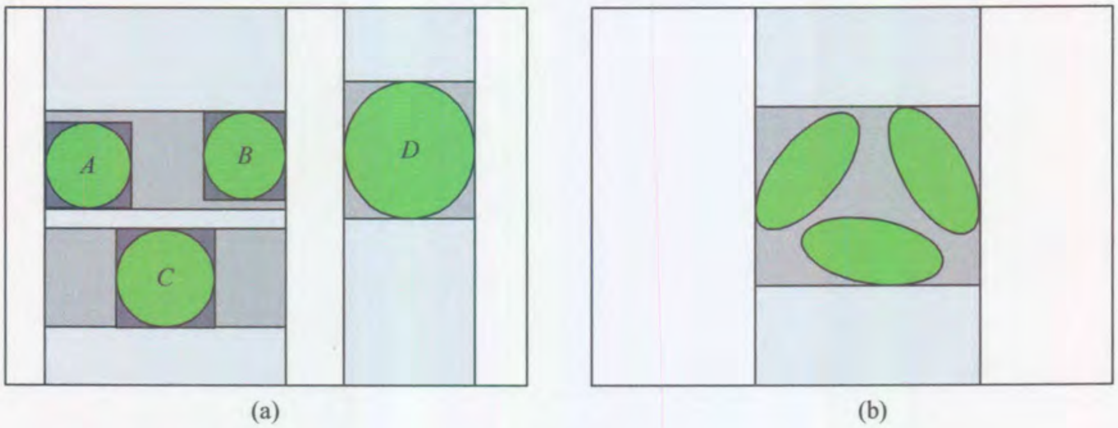


Figure 4.4: Segmentation of muliple objects. Figure (a) shows how four objects are isolated by the simple location algorithm. This algorithm requires horisontal and vertical gaps between objects, otherwise objects are not located individually as shown in Figure (b).

nificantly deviate from the background values. The result is that background surfaces can potentially be segmented as OIs. To limit such occurrences, constraints such as constant lighting are imposed on the environment. Section 4.3 discussed these constraints in more detail. Unfortunately, camera noise and cast shadowing are difficult to control. Noise is an inherent property of CCD cameras and cast shadowing a result of the person's presence in the scene. By explicitly modeling this, it is possible to compensate to some extent for these phenomena and hence avoid false classification of pixels. Shadowing is modeled by regarding it as an attenuation of radiated light. In other words, if the light intensity of a pixel in an active image is lower than its intensity in the background image and its colour is unchanged, then this is probably caused by shadowing. In this case the pixel is classified as background. Camera noise is modeled statistically. To do this a number of background images are acquired so that the distribution parameters of each pixel value can be estimated. Each pixel is treated independently and 50 to 100 images are sufficient to estimate the distribution parameters. Taking all the above into account, a hypothesis test is devised that tests whether each pixel belongs to an OI or background.

Figure 4.3 shows how a person is located in an example frame. The result is a binary level image - white pixels denoting foreground and black pixels denoting background.

4.4.2 Bounding box construction

The 2D bounding box is calculated such that it encloses the segmented person in the smallest possible rectangle. For this purpose the following simple procedure is used: Start at the left most column of the segmented image and search column by column from left to right for the first white pixel. Once found, the left line of the bounding box is denoted by this pixel's x-value less one. Continue searching until all pixels in the column are black. This column denotes the right line of the bounding box. The same procedure applies for the top and bottom bounding box lines, but the rows are searched instead of the columns.

Despite a carefully controlled environment, it sometimes happens that the segmenter detects multiple objects. Objects in the scene might move or the light level might change slightly. As already discussed, this can cause some background pixels to be misclassified making it appear as if an object is detected. The algorithm described above cannot cope with multiple objects. Instead it encapsulates all objects in a single bounding box or only detects a single object which might not be the object of interest.

To cope with the possibility of multiple objects, the above algorithm is extended [54]: By alternating the search along horizontal and vertical directions, all objects can eventually be detected. An example containing four circular objects is shown in Figure 4.4(a). The algorithm first searches the image columns from left to right. The first iteration detects two object groups that are marked by the light grey blocks. Next, it independently searches the image rows of each object group from top to bottom but only within the previously determined vertical bounds. This search results in the location of three object groups (medium grey blocks). The process continues until no new object group is detected. Each object group then contains a single object. Objects A, B and C of Figure 4.4(a) were located after three iterations and object D after two. This algorithm was used by Wohlberg and Cox [54] to track multiple people. If the original algorithm described earlier is to be used in the scenario presented in Figure 4.4(a), it will detect only two objects, namely A, B and C as the first object and D as the second object.

In some cases multiple objects are grouped as a single object even if they are spatially separated. An example is shown in Figure 4.4(b). To distinguish between the objects more advanced techniques than the simple horizontal and vertical search algorithm described above are required to check for object connectivity [55]. Since the pixel area of the person in a scene is usually much larger than any other possible falsely detected object, this is ignored. The reason for this will be apparent soon.

When multiple objects are detected in an image one needs to determine which

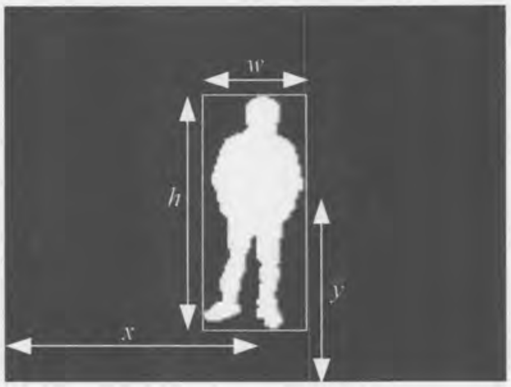


Figure 4.5: A bounding box encapsulates the segmented person in the smallest possible rectangle. It has four parameters, (x, y, w, h).

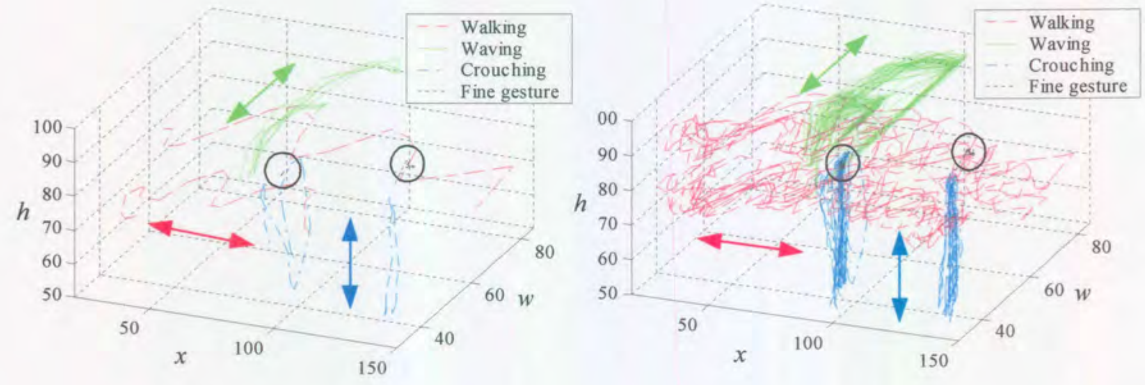
object is the person. It is assumed that the person is the object with the largest pixel surface area. In most cases very few pixels are misclassified as OI objects and even if they are misclassified, the falsely detected objects have small pixel areas. The above assumption is therefore valid in most cases.

It is now possible to determine the bounding box of the person in the scene. Referring to Figure 4.5, this model is represented by its centroid coordinates, (x, y), and its dimensions, (w, h).

4.4.3 Feature selection and conditioning

The next step is to select features from the model. This is done by visually inspecting the model parameters of the various gestures to be recognised and then deriving a meaningful feature vector. Each of the gestures has a temporal signature in the model's parameter space. The distinctiveness of these signatures determines how easily they are recognised. Figure 4.6(a) shows typical signatures for the four gestures. Only (x, w, h) is shown, since y is mostly constant, except for crouching in which case it has a similar signature to h. For each gesture two signatures are shown in the figure. These were obtained from two people of different build. Each curve consists of 20 points, which corresponds to two seconds for the particular gesture. This is sufficient for visual recognition of the gesture in a video sequence and the curves should therefore contain the required information for recognition. Figure 4.6(b) shows the curves for the same two people and gestures, but repeated over a period of 30 seconds.

The location of these curves in parameter space is related to the location of the person in the image sequence. For example, crouching was performed by the two people



(a) (b)

Figure 4.6: (a) shows the temporal parametric curves of the model parameters for two people. Each curve consists of 20 points which corresponds to 2 seconds of a particular gesture. Figure (b) shows the same gestures but over a 30 second period. For both these figures the *fine gesture* curves are encircled for easy identification. The arrows indicate the principle direction of the curves for particular gestures.

at pixel locations $x = 100$ and $x = 140$ respectively. The absolute location of a curve is, however, not important, but how it changes over time. This suggests that one needs to use a recognition technique that explicitly models for such a process e.g. a Markov random process. A much simpler recognition technique can be devised if one ignores the temporal or sequential properties of the parameters for the moment and considers them to be a distribution of data. Then, for each class set (corresponding to a gesture) the data has a large variance along a specific direction. For example, the distribution of *walking* is predominantly along the x-direction and that of *crouching* along the h-direction as shown in Figure 4.6(a) and (b). In other words, the distributions of the various gestures have prominent first principle components and these principle components are fairly distinctive for all the classes. This fact can be used to derive a suitable feature vector for classification. The exception to this is the *fine gesture* class that has an unpredictable first principle component (see encircled distribution of Figure 4.6(a) and (b)). This gesture class also has a much smaller variance than the other classes. A suitable feature vector might then be the first principle component's eigenvector and its eigenvalue. The eigenvalue is a measure of the principle component's variance.

The above procedure is summarised as follows: The last c observations of (x, w, h) of a gesture sequence is saved in a first-in-first-out (FIFO) buffer. These observations are treated as a distribution and its principle components are calculated. The first principle component's eigenvector and eigenvalue are measures of the gesture performed during the last c image frames. By repeating this process one should be able to classify gestures as time progresses.

Calculating the principle components can be simplified considerably by noting that the "derivative vector" (vectors with component-wise derivatives) of the curve in Figure 4.6(a) points approximately in the same direction as the first principle component. This is illustrated in Figure 4.7(a) for *walking* where the derivative is being calculated at three different time instances. In each case the vectors have large components in *walking*'s first principle component direction. Figure 4.7(b) shows the derivative of the gesture sequences of Figure 4.6(b). From this figure one can see that the derivatives of the curve indeed have large components along their respective gesture's principle components. Of greater importance from a classification point of view is that the classes are fairly well separated, the exception being at the origin of the graph. The *fine gesture* class is distributed around the origin, resulting in a low recognition rate for this class. This can be improved by considering the derivatives for the last c observations. If one visually averages the three derivative vectors of Figure 4.7(a), the resulting vector better approximates the first principle component of *walking*. The following three simple equations can therefore be used to approximate the principle direction of a gesture and hence serve as feature vector:

$$v_1(n) = \frac{1}{\mu_w c}\sum_{i=0}^{c-1}|x_{n-i} - x_{n-i-1}| \qquad \text{with } \mu_w = \frac{1}{m}\sum_{j=0}^{m-1} w_{n-j} \text{ and } n \geq 1 \tag{4.1}$$

$$v_2(n) = \frac{1}{\mu_h c}\sum_{i=0}^{c-1}|h_{n-i} - h_{n-i-1}| \qquad \text{with } \mu_h = \frac{1}{m}\sum_{j=0}^{m-1} h_{n-j} \text{ and } n \geq 1 \tag{4.2}$$

$$v_3(n) = \frac{1}{\mu_w c}\sum_{i=0}^{c-1}|w_{n-i} - w_{n-i-1}| \qquad \text{for } n \geq 1 \tag{4.3}$$

where n is the frame number and μ_w and μ_h are scaling factors introduced to provide some measure of invariance with respect to distance from camera. The perspective projection property of a camera causes the apparent size of a person to change as he or she moves away or closer to the camera. Both μ_w and μ_h are calculated over a one second period and therefore m is set equal to the frame rate.

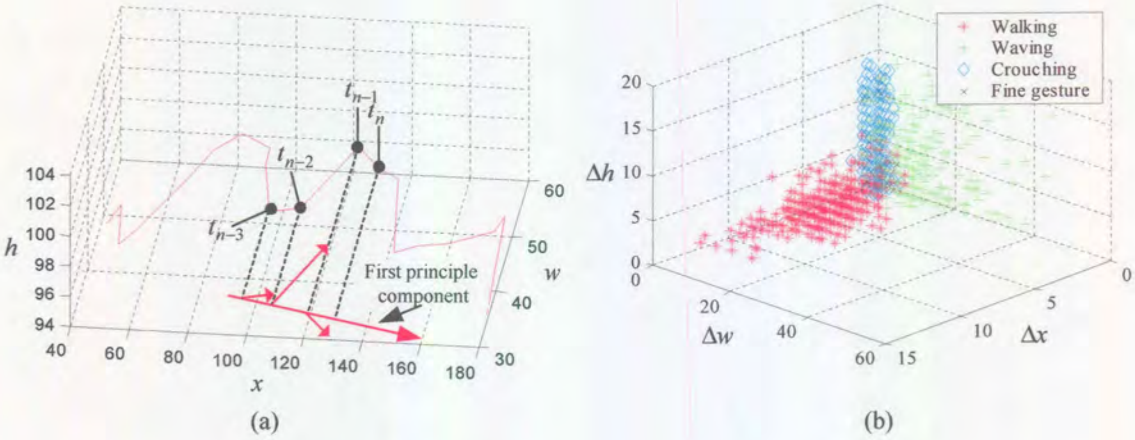


Figure 4.7: (a) shows the temporal signature of *walking* for 20 samples. The derivative of this signature has large components in the direction of the first principle component as shown for three points. (b) shows the derivatives of the temporal signatures of Figure 4.6(b).

4.4.4 Classification

In the classification stage one tries to determine the class of an unknown feature vector, **v**. To determine the class a radial basis function (RBF) neural network is used [56]. It is trained on the data distribution shown in Figure 4.8. Details of the data sets generation are given in the next section. A network with seven Gaussian basis functions was chosen. The number of basis functions was determined by inspecting the feature distribution of Figure 4.8 and visually fitting Gaussian basis functions with full covariance matrices over the data. The aim is not to over fit the data with too many kernels or under fit it with too few kernels. Seven basis functions seemed to be a good trade-off and this was confirmed by testing the neural network.

The network has four outputs - one for each gesture class. It is trained to output 1 if the class is true and 0 if it is false. During testing of an unknown feature vector the network output can take any value from 0 to 1. The selected gesture class is the one with the largest output and on condition that it exceeds a threshold of 0.5. If these conditions do not hold the class is classified as *unknown*. The *unknown* class is therefore not recognised explicitly, but rather as the absence of the other classes. Final classification is implemented as a voting system that bases classification on the last 10 network values of each class. The class with the most 1 votes is the final selected class. This voting mechanism helps to eliminate spurious outlier classifications of the

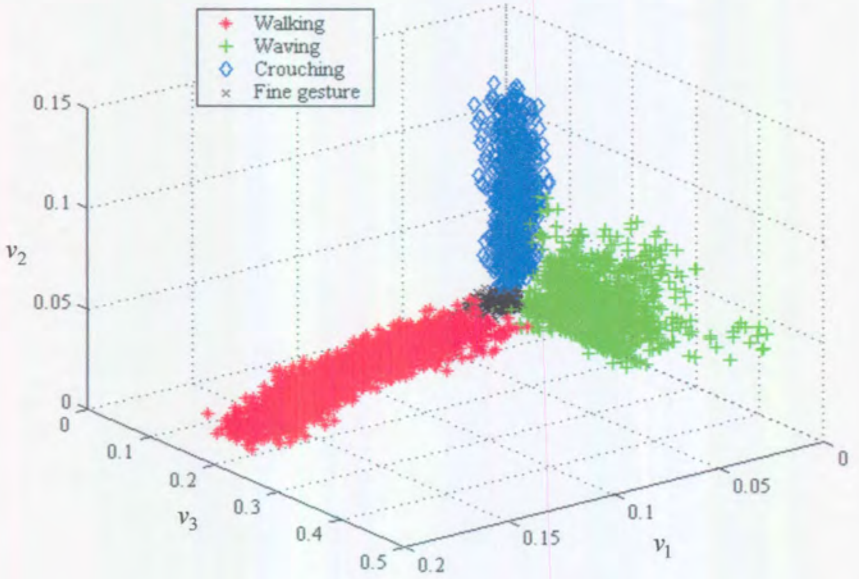


Figure 4.8: The data set distribution of the four gestures after conditioning by equations 4.1 to 4.3.

neural network. An outlier classification usually causes the wrong class to be selected by the neural network. If this does not happen too often, final classification is still unaffected as a result of the voting system. In addition, the voting system is also used to calculate a type of *a posteriori* probability. This is simply the percentage of votes for a particular class. It should be noted that the voting system is effectively a filter. The queue size is therefore kept small at 10 (approximately the image frame rate) to minimise lag. This means that the system's gesture verdict slightly lags that of the actual gesture.

Person	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
Height (m)	1.85	1.89	1.77	1.60	1.50	1.70	1.74	1.70	1.75
Weight (kg)	80	94	68	73	52	73	70	80	108

Table 4.1: Physical profiles of the participants used for training and testing the system.

4.5 Results

The system was trained on five participants, a subset of the nine participants on which it was tested. Physical profiles of these participants are given in Table 4.1. Training data was generated of the first five participants (A to E in the table) and separate testing data generated of all nine participants. Figure 4.8 shows the feature distribution of the training data. It was generated from the five people who participated in training: Four video sequences were captured of each participant. Each video sequence contains a single gesture. A video sequence is approximately 30 seconds and captured at a rate of 10 frames per second. Therefore, in total 150 seconds of video was used to generate the data for each class. This corresponds to 1500 data points per gesture. To capture gesture variability as well as possible within the discussed constraints of Section 4.3, the five participants were asked to perform the gestures at different rates and at various distances from the camera.

The test set was generated by capturing a video sequence of each of the nine participants who took part in testing. Each sequence is approximately 50 seconds and captured at a frame rate of 10 frames per second. All four gestures of a particular participant are contained in the sequence, where each gesture varies between 10 to 15 seconds.

To test the system a ground truth of the test sequences was first obtained. This was done by hand classifying the test sequences frame by frame. The recognition system was then used to classify the same set of sequences. The ground truth and machine classified verdicts were then compared. A frame is correctly classified if the system and ground truth have the same verdict[6]. The results are shown in Table 4.2. Average system performance is 96.3%.

Two gesture recognition systems were implemented, namely an online and offline system. The offline system performs recognition on recorded video data and is primarily used for training and testing of the recognition system. The online system performs recognition of live video data in real-time at approximately 14 frames per second on a Pentium III-600 PC. Two example videos of the online gesture recognition system can be viewed at *chap4\recognition\* on the CD.

[6] The classification verdict of the system is given by the output of the voting system (see Section 4.4.4)

Person	Gesture recognition rate (%)				Avg(%)
	Walking	Arms Waving	Crouching	Fine gest.	
A	96.8	95.2	93.3	95.8	95.3
B	97.4	95.9	94.9	97.4	96.4
C	98.0	95.8	91.4	95.5	95.2
D	97.9	97.2	91.1	99.1	96.3
E	98.7	94.1	94.0	99.2	96.5
F	96.1	93.9	96.5	97.6	96.0
G	98.4	97.0	93.8	100	97.3
H	98.3	92.3	96.7	98.2	96.4
I	94.7	96.2	95.7	100	96.7
Avg(%)	**97.4**	**95.3**	**94.2**	**98.1**	**96.3**

Table 4.2: Gesture recognition rate results of the 2D gesture classifier tested on 9 participants

4.6 Discussion

Given the little class overlap of the class distributions, one expects a good recognition rate. This is reflected in the results summarised in Table 4.2. False positive classifications mostly occur during the initial period of a new gesture. During this period the new gesture is classified as the previous gesture. This phenomenon is attributed to a delay in the response of the classifier. The implication of this delay is shown in classification confusion matrix (see Table 4.3). For the purpose of acquiring test data, participants were asked to perform gestures in the sequence *walking*, *waving*, *crouching* and then *fine gesture*. During system testing, the initial 1 to 2 seconds of the respective gestures were classified as *unknown*, *walking*, *waving* and *crouching*.

The classification delay is caused by the classifier voting system and filtering operations in Equations 4.1 to 4.3. The filtering operations are required to construct a feature vector, since the system uses the last c (where $c = 10$) observations to calculate the vector. c impacts the delay and the choice for its value is described next: A larger c means that more observations are used to approximate the first principle component. This causes the class separation in Figure 4.8 to be better and leads to an improved recognition rate. The drawback of a larger c is that the system has a longer response time, or lag. The lag can be so large that the system might miss detection of gestures if, for example, the person moves rapidly from one gesture to the next. The opposite of this explanation is true for too small values of c. A value of $c = 10$ (equals the video

System	True Class (%)			
Classified	Walking	Waving	Crouching	Fine gesture
Walking	**97.4**	4.4	0.0	0.0
Waving	0.0	**95.3**	5.8	0.0
Crouching	0.0	0.0	**94.2**	1.9
Fine gesture	0.4	0.33	0.0	**98.1**
Unknown	2.5	0.0	0.0	0.0

Table 4.3: Classification confusion matrix of the 2D gesture classifier.

frame rate), seems to be a good trade-off. Practically this means that the system has a response of around 1 to 2 seconds to an input.

4.7 Conclusion

It is clear from the results that a bounding box can be used as basis of an automatic gesture recognition system for the given gesture vocabulary. The system was able to discriminate between the four gestures 96.3% of the time and false positive classification is a result of classification dynamics. The current system is however limited to the recognition of coarse gestures, has a small vocabulary of four gestures and has to comply to the plane motion constraints. As gathered from the MBB experiment results, coarse gesture recognition is a property of the bounding box representation. The gesture vocabulary can however still be extended and the 2D constraints can possibly be removed by a 3D bounding box representation. These possibilities are investigated in the next chapter.

Chapter 5

Coarse 3D gesture recognition

In the previous chapter it was demonstrated that coarse gestures can be recognised by modeling the human body with a simple 2D bounding box. A single camera was used and gestures executed by a person were recognised under certain restrictions. Specifically, the person had to face the camera or some gestures were incorrectly classified. For example, *walking* had to be executed in a plane perpendicular to the camera's optical axis. If this was not the case, the system's classification alternated between *walking* and *crouching*. In a larger gesture vocabulary, one would probably find that recognition of other gestures have similar undesirable results. This is due to the representation - a 3D spatial problem is represented by a 2D spatial model.

The 2D motion restriction severely limits the practical application of the system, since we are living in 3D space. In this chapter a system is presented that extends the 2D bounding box concept to a 3D one. The objectives of the proposed 3D system are given in Section 5.1 and the approach to the problem in Section 5.2. As was the case for 2D system, a number of assumptions are made here in order to make the system viable. These are given in Section 5.3. This is followed by the main body of the chapter, which describes the operation of the 3D system in Section 5.4. The test results of the 3D system are given in Section 5.5, which is followed by a discussion in Section 5.6.

5.1 Objectives

The objective here is to investigate a 3D bounding box model for the purpose of gesture abstraction in an automatic gesture recognition system. It is expected that such a model will remove the plane motion constraint imposed by a 2D system (see assumptions for the 2D system in Section 4.3). In addition, recognition invariance to facing direction is investigated. If invariance is achieved, the system will be totally

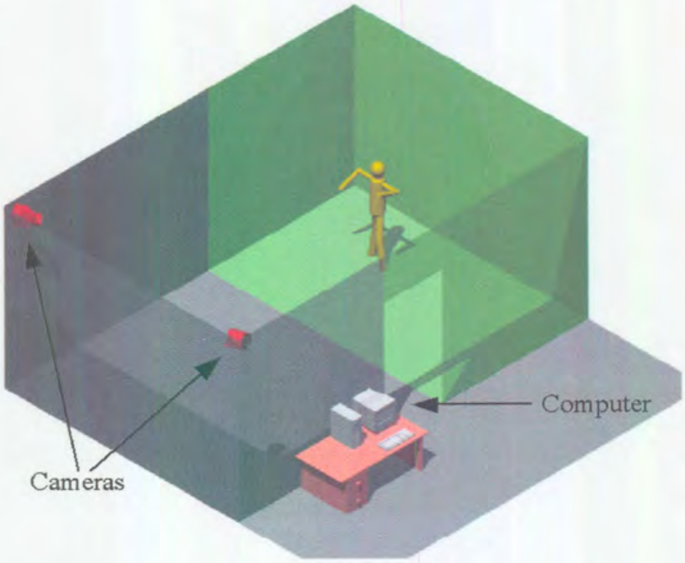


Figure 5.1: The setup for a 3D gesture recognition and tracking system. Two cameras are located at a wide-baseline, looking downwards to the activity area. These cameras are interfaced to a computer that runs the gesture recognition and tracking algorithms.

unconstrained from a spatial representation point of view.

The above objectives are to be tested on an extended gesture vocabulary. Gesture vocabulary is increased from the four gestures of the 2D system to the following eight gestures: *walking*, *waving*, *crouching*, *standing-stretch*, *standing-star*, *standing-normal*, *sitting* and *lying down*.

5.2 Approach

To investigate the 3D bounding box model a way must be found to construct it. Two techniques that are often used to recover 3D object structure are stereopsis [1] and structure from motion (SFM) [57]. Stereopsis uses the views of two or more cameras to recover object structure. These cameras are spatially separated and observe the object from different viewpoints. A triangulation procedure enables recovery of the structure. The advantage of SFM techniques is that object structure can be recovered from monocular vision and require only a single camera. To recover object structure, relative motion is required between the camera and object. Either the camera moves and observes the object from various viewing locations or the object rotates displaying different views to the camera. SFM is best suited for rigid objects and since the human body is nonrigid SFM cannot directly be applied to gesture recovery problems. By including *a priori* information of the nonrigid object (e.g. body shape of the person) into the model, it is possible to recover object pose. For example, Chen and Lee [13] were able to track the joints of an animated stick figure in 3D from a single camera. Their system requires that the dimensions of the stick figure are known, that the figure is walking and that a complete gesture sequence is acquired.

SFM makes too many assumptions about the behaviour of the object to be of use here. For this reason it was decided to use stereopsis in this system to recover body pose. This requires an extension to the setup of the existing 2D system camera. A second camera was added to the experimental area and is located at a wide base-line relative to the first camera. For practical reasons, the cameras were installed in the corners of the room near the ceiling (see Figure 5.1).

The primary difference between the 2D gesture recognition algorithm and the 3D extension discussed here is the construction of a 3D bounding box. Given images from the cameras, two 2D bounding boxes are independently calculated for each image. A 3D bounding box is then constructed from the two 2D bounding boxes. Once this is achieved, a procedure similar to that of the 2D system is used to recognise the gestures.

5.3 Assumptions

The same assumptions and constraints are imposed on this system as for the 2D one (see Section 4.3). The exception is that motion is unconstrained (not restricted to a plane) and the gesture vocabulary consists of only the following: *walking*, *waving*, *crouching*, *standing-stretch*, *standing-star*, *standing-normal*, *sitting* and *lying down*. Any other gestures presented to the system should be classified as *unknown*.

5.4 Gesture recognition in 3D

Figure 5.2 shows a diagram describing the 3D gesture recognition algorithm. It consists of steps similar to the 2D recognition algorithm. The primary addition to this algorithm is to relate the bounding boxes of the two camera views. This is achieved by a camera calibration procedure (see Appendix B). An advantage of using stereopsis is that the person's location in space can easily be determined, i.e. it is possible to track the person. This part of the algorithm is performed by the Person Tracker module of Figure 5.2.

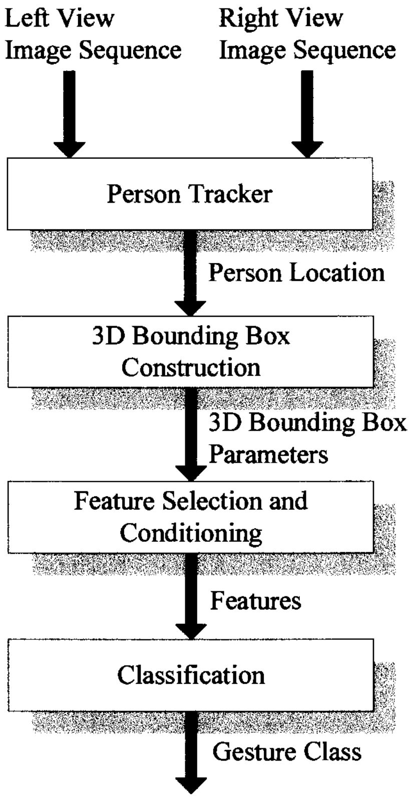


Figure 5.2: A high level flow diagram of the algorithm used to recognise gestures and track people in 3D. The inputs to the algorithm are two camera image sequences. The output of the system is the person's location and the gesture class.

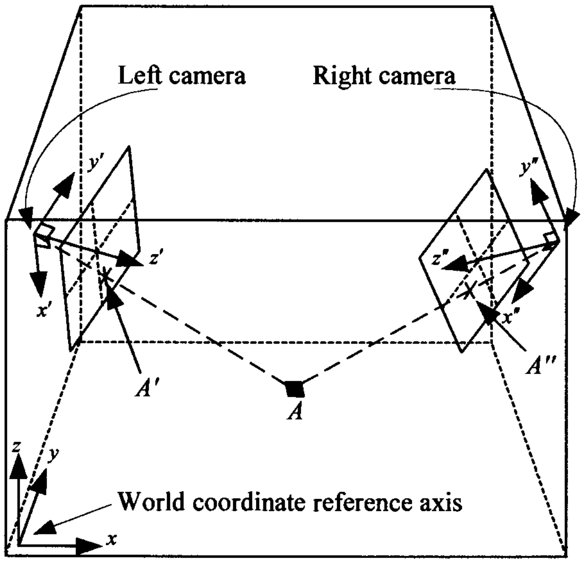


Figure 5.3: This figure illustrates how a point object (A) is tracked in space using stereopsis. The projected locations of the point object (A' and A'') on the camera's image planes are known. Given this, the object's location in space can be determined by means of triangulation.

5.4.1 Person tracking

This section starts by explaining how to track a point object in space using two wide-baseline cameras. Once this is accomplished it is easy to extend the concept to track a person in 3D or construct a 3D bounding box from two 2D bounding boxes.

Figure 5.3 shows a point object denoted by A in 3D space. The aim is to follow or track this object as it moves in space. A' and A'' represent the perspective projection of A onto the image planes of the left and right cameras respectively. Given the projection of the point object on a camera view, say the left camera, we know that the object is located somewhere on a line (dashed line AA' in the figure) in space. To uniquely determine the object's location, a line is traced from a second camera through the object. In other words the object's location is determined by means of triangulation. The equations of the two trace lines are given relative to the respective reference axes of the cameras. These reference axes are, however, unrelated making triangulation impossible. For this reason the cameras need to be calibrated. This essentially means that their location (described by a translation vector ($\mathbf{t}$)), orientation (described by a rotation matrix ($\mathbf{R}$)) and focal length (f) have to be known relative to some common reference axis. The common reference axis is chosen to be a corner of the room (world

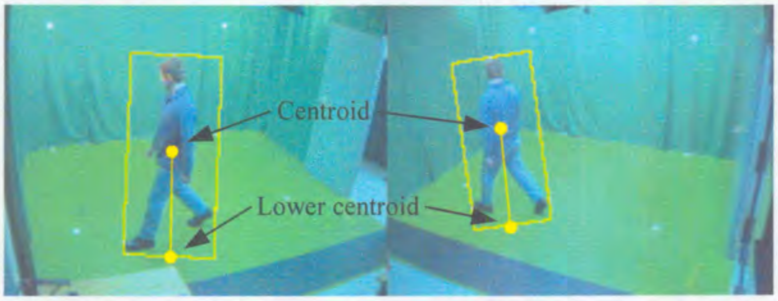


Figure 5.4: To track a person, the lower centroid serves as input to the triangulation procedure. This point is approximately where the person's feet are.

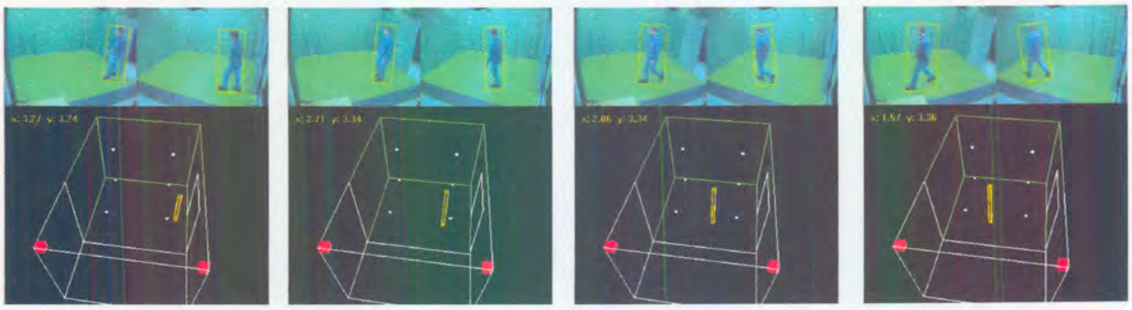
reference axis in Figure 5.3). A detailed description of the calibration procedure and 3D object tracking is given in Appendix B.

To track a person a point on the person is selected. This point is chosen to be the person's lower centroid as illustrated in Figure 5.4. The lower centroid is defined as the point that intersects the bottom bounding box line and the line that is perpendicular to this line and passes through the true centroid. In each view the person's lower centroid is therefore calculated and used as input to the above triangulation procedure. Figure 5.5 shows how tracking is accomplished for an example frame. Example tracking videos can be viewed at *chap5\tracking\*.

5.4.2 Bounding box construction

Each camera view produces a 2D bounding box. The first step is to correct for the apparent rotation of these bounding boxes. This phenomenon is demonstrated in Figure 5.5. As the person moves from left to right, his or her observed orientation changes (see also Section B.3 of Appendix B). A rotation correction algorithm has been applied to the bounding boxes in this figure - this algorithm tracks the apparent rotation and corrects for it as the person moves about. This phenomenon is caused by the cameras being nonlevel to the floor surface. By correcting for it, a 3D bounding box is constructed that better approximates the coarse shape of the human body[1]. From a pattern recognition point of view this means that noise in the feature space is

[1] The apparent rotation correction algorithm was not applied to the 2D gesture recognition system. It requires the position of the person in space to be known. This is not available in the 2D case, but is eliminated by ensuring that the camera is level to the floor surface. This is, however, not always practical. Cameras mounted near the ceiling and looking downwards as in the case of the 3D system are preferred in which case apparant rotation should be corrected.



$x = 3.27; y = 3.24$ Frame 21

$x = 2.71; y = 3.34$ Frame 26

$x = 2.06; y = 3.34$ Frame 31

$x = 1.57; y = 3.36$ Frame 36

Figure 5.5: This example sequence illustrates how a person can be tracked using stereopsis. Top images are the left and right camera views respectively (the cameras are represented by the red cubes in the bottom wire frame drawing). The (x, y) position of the person is given relative to the lower bottom left corner of the room. This sequences also illustrates the apparant rotation of the person as he or she moves around. For example, as the person walks from right to left in this sequence, his body translates and appears to be rotating in the camera views. The bounding box algorithm tracks this apparant rotation. Another example of this phenomenon is given in Figure B.3 of Appendix B.

reduced. The rotation correction procedure is described in detail in Appendix B.

Given the two orientation corrected bounding boxes, a 3D bounding box can be constructed. This problem is in essence the same as discussed in the previous section where the aim was to track a point in 3D space, given two 2D projection of the point. In this case each vertex of the 3D bounding box (3D points) has two 2D projections onto the camera views. The projections correspond to the vertices of the 2D bounding boxes. In order to determine the vertices of the 3D bounding box, the appropriate vertices of the 2D bounding boxes are used. This is illustrated in Figure 5.6(a) and 5.6(b). The bottom left vertices of both the 2D bounding boxes map to the vertex of the 3D bounding box marked A. Vertex B is constructed from the bottom right vertex of the left camera view's bounding box and the bottom left vertex of the right camera view's bounding box.

Once construction is completed, the bounding box is represented by its parameters. These are height (h) and the perimeter (s). It should be noted that h and s do not uniquely define a 3D bounding box, since $s = s' + s'' + s''' + s''''$ (see Figure 5.6(c)). They do, however, relate conveniently to the 2D bounding box parameters of height

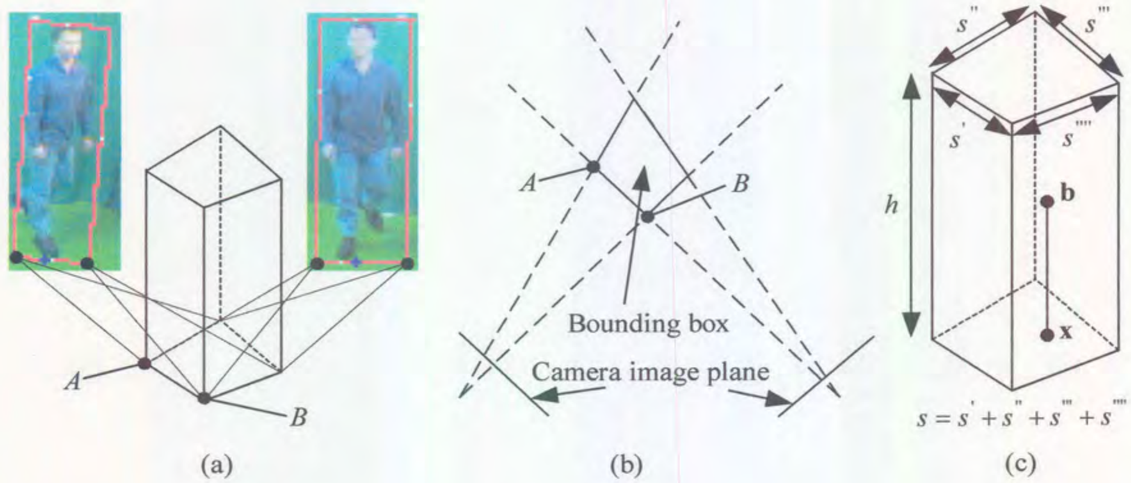


Figure 5.6: Construction of a 3D bounding box from two 2D bounding boxes. Each vertex on the 3D bounding box is calculated from two appropriate points - one from each camera view. For example, point A in (a) is determined from the bottom left vertices of the two camera views. Figure (b) shows the reconstruction as seen from the plan view. Figure (c) shows the bounding box parameters. h and s are respectively the height and perimeter of the base. $\mathbf{b}$ and $\mathbf{x}$ are the respective true and lower centroids.

and width: Height for both cases has the same meaning, while width in the 2D case is related to perimeter in the 3D case. Two other useful parameters are the true centroid ($\mathbf{b}$) and lower centroid ($\mathbf{x}$).

5.4.3 Feature selection and conditioning

One of the objectives of the 3D system is to increase the gesture vocabulary with respect to the 2D system. The vocabulary consists of two gesture groups namely dynamic gestures (*walking*, *waving*, *crouching*) and poses (*standing-stretch*, *standing-star*, *standing-normal*, *sitting*, *lying down*). The 3D bounding box parameters have similar signatures to that of the 2D bounding box (see Figure 4.6). The same principle component-like features can therefore be used for the gestures. Poses have no temporal signature and need different features. Feature selection is discussed in the remainder of this section.

The dynamic gesture features are similar to those used in the 2D recognition system, but are adapted for the 3D model. They are expressed by:

$$v_1(n) = \frac{1}{c}\sum_{i=0}^{c-1} ||\mathbf{x}_{n-i} - \mathbf{x}_{n-i-1}|| \qquad \text{for } n \geq 1 \tag{5.1}$$

$$v_2(n) = \frac{1}{c}\sum_{i=0}^{c-1} |h_{n-i} - h_{n-i-1}| \qquad \text{for } n \geq 1 \tag{5.2}$$

$$v_3(n) = \frac{1}{c}\sum_{i=0}^{c-1} |s_{n-i} - s_{n-i-1}| \qquad \text{for } n \geq 1 \tag{5.3}$$

where $\mathbf{x}_n$, h_n and s_n are the person's lower centroid (position), height and bounding box perimeter at frame n respectively. c is the number of observations used to approximate the principle components and is chosen to be equal to the frame rate ($c = 10$). The major differences between these features and that of the 2D recognition system are that bounding box perimeter is used in Equation 5.3 instead of width as in the case of Equation 4.1 and that the features are not normalised for depth. A property of stereopsis is that the true dimensions of the bounding box are recovered directly. The depth problem caused by the perspective projection property of a camera does not occur here and it is therefore not necessary to accommodate for it.

To derive pose features, we notice that the bounding box model parameters are constant over time if noise is ignored. Features can therefore be derived by considering the ratio of one parameter to another. A number of features were evaluated by means of inspection and those with the best class separability were chosen:

$$v_4(n) = \frac{h_n}{s_n} \qquad \text{for } n \geq 1 \tag{5.4}$$

and

$$v_5(n) = \frac{b_n}{b_{avg}} \qquad \text{for} \geq 1 \tag{5.5}$$

v_4 is an aspect ratio measure of a particular gesture and v_5 a measure of how "stretched out" the gesture is. b_n is the height of the person's true centroid and b_{avg} is the person's average centroid height. By normalising b_n using b_{avg} some invariance with respect to the height of the person is achieved. b_{avg} is a constant that depends on a person's height and is determined by the system when the person is in an upright position.

When *walking* is executed the person is in an upright position and it was therefore decided to calculate b_{avg} when the system detects *walking*. This can, however, lead to unstable classification, since this approach requires the output of the classifier to be passed to its input. The problem is solved by using two separate classifiers for the dynamic gestures and poses. *Walking* is detected by the dynamic gesture classifier and whenever this is the case, b_{avg} is also updated by averaging b_n over the last c frames.

5.4.4 Classification

The final stage of the algorithm classifies an unknown feature vector as belonging to one of the eight gesture classes. The classifier has a similar architecture to that of the 2D recognition system (see Section 4.4.4). A two stage classifier consisting of a neural network and a voting system is used (see Figure 5.7). The neural network classifies the features on a frame by frame basis. The result is binarised by a threshold of 0.5 and passed to a FIFO queue that comprises the voting system. The gesture class with the most votes is selected by the system as the performed gesture.

Gestures and poses are mutually exclusive and classification can therefore be based on two separate neural networks - one for each gesture group. The gesture network has an *inactive* class trained on all the poses (see Figure 5.7). When this class is detected by the system, it knows that the gesture is a pose and subsequently the pose classifier is used for classification. The feature b_{avg} (person's centroid height) is used by the pose network and is only updated when *walking* is performed. Since two separate classifiers are used, the potentially unstable feedback condition discussed earlier is avoided.

Both neural networks are RBF neural networks, each with 10 Gaussian basis functions. The number of basis functions was determined by visual inspection using the same rationale as for the 2D system (see Section 4.4.4). The networks were trained using various gesture training sequences as discussed in the next section.

5.5 Results

The system was trained on six people and tested on eleven people. Physical profiles of these people are given in Table 5.1. Training data was generated for person A to F and separate testing data of all eleven people. The distribution of the training data is shown in Figures 5.8(a) and 5.8(b) for the dynamic gestures and poses respectively. It was generated by using the six people who participated in training by capturing eight video sequences of each participant. Each video sequence contains a single gesture. A

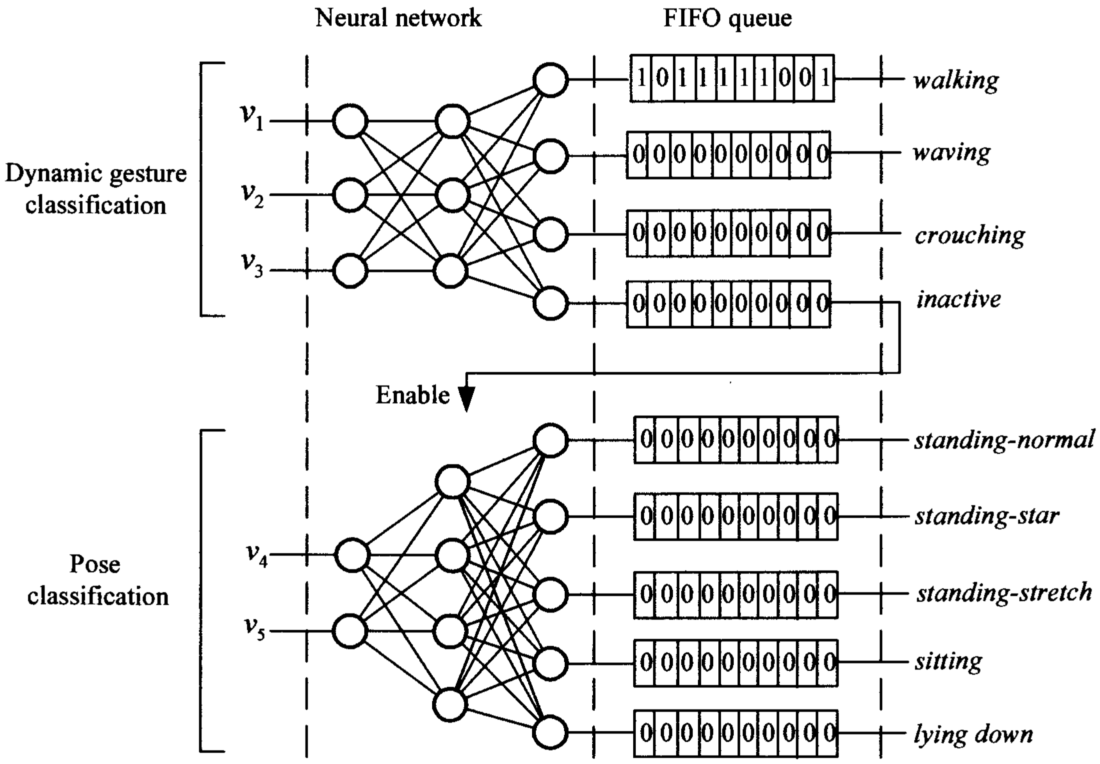
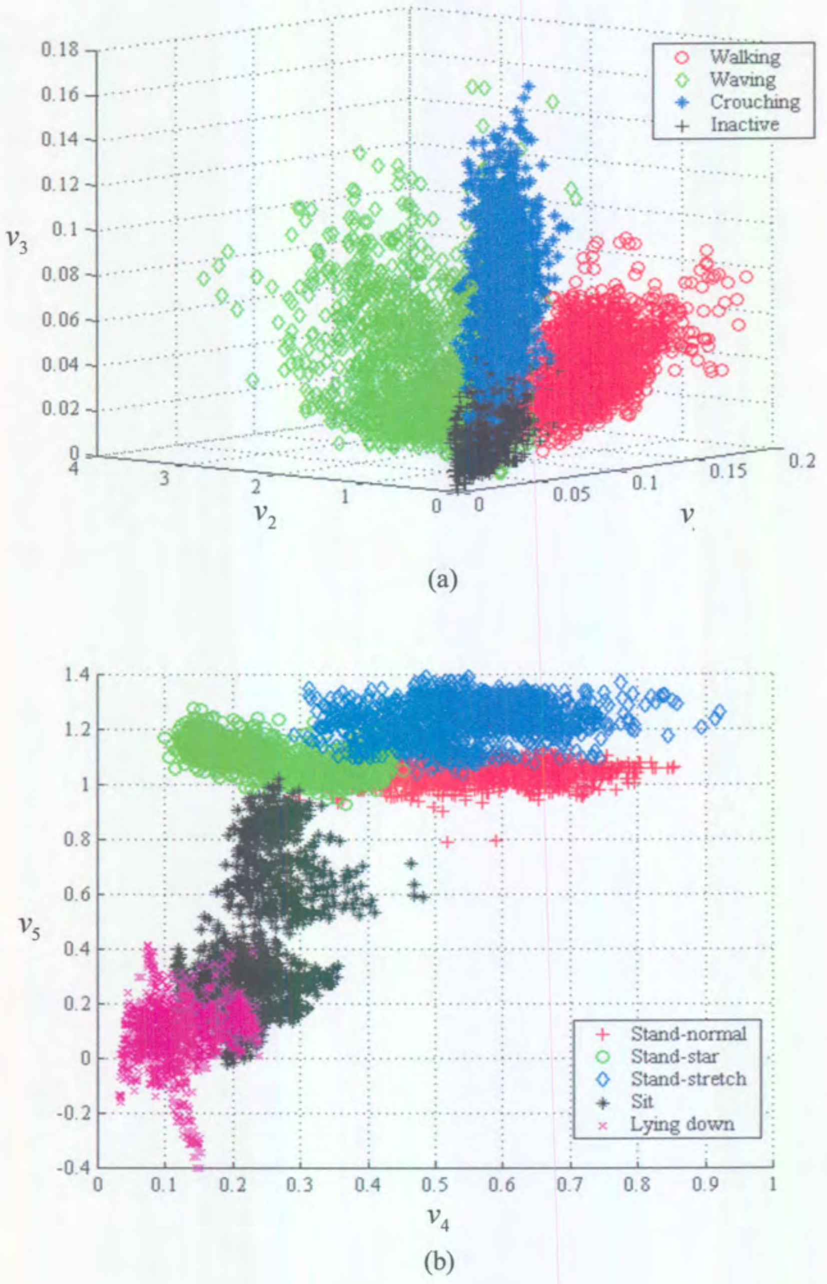


Figure 5.7: Classification consists of a RBF neural network and voting system. The class with the most votes in its queue is the selected class. Dynamic gestures and poses are classified by two separate classifiers.



(a)

(b)

Figure 5.8: Feature distribution of training gesture sequences for (a) dynamic gestures and (b) poses.

Person	A	B	C	D	E	F	G	H	I	J	K
Height(m)	1.22	1.40	1.89	1.50	1.70	1.91	1.85	1.77	1.70	1.80	1.95
Weight(kg)	21	35	94	52	73	95	80	68	80	82	120

Table 5.1: Physical profiles of the people used as training and testing subjects

video sequence is approximately 30 seconds and captured at a rate of 10 frames per second. Therefore, in total 180 seconds of video was used to generate the training data for each class. This corresponds to 1800 data points per gesture. During acquisition of training and testing data, participants were asked to execute gestures and poses while facing various directions and in the case of the dynamic gestures they were requested to execute the gestures at different rates.

The test set was generated by capturing two video sequences of each of the eleven participants - one sequence contained the gestures and the second sequence contained the poses[2]. Each sequence is approximately 50 seconds and captured at a frame rate of 10 frames per second. The gestures and poses were executed for approximately 10 to 15 seconds by participants in the respective video sequences.

To test the system a ground truth for each test sequence was first obtained by classifying them by hand. The recognition system was then used to classify the same set of sequences. The ground truth and machine classified verdicts were then compared. A frame is correctly classified if the system and ground truth have the same verdict. The results are summarised in Table 5.2 with an average recognition rate of 84.7%.

Two gesture recognition systems were implemented, namely an online and offline system. The offline system performs recognition on recorded video data and is primarily used for training and testing of the system. The online system performs recognition of live video data in real-time at approximately 9 frames per second on a Pentium III-600 PC. Example videos demonstrating the 3D system are located in *chap5\recognition\* on the CD.

5.6 Discussion

The system has a fairly large performance range over gesture class, ranging from 67.8% for *sitting* to 95.1% for *walking*. There is a clear distinction between the performance associated with dynamic gestures and poses. Dynamic gestures have an average recog-

[2] It is easier to manage smaller video sequences and for this reason they were split into two sequences per person.

Person	**Gesture recognition rate (%)**								**Avg(%)**
	Wk	**Wv**	**Cr**	**Sl**	**Sr**	**Sh**	**Sit**	**Ly**	
A	94.4	*44.8*	98.7	82.5	88.6	89.8	92.9	89.6	**85.2**
B	81.9	80.4	89.8	92.3	89.4	75.9	*0.0*	75.6	**73.2**
C	99.0	100.0	92.4	93.7	76.7	82.0	79.2	81.8	**88.1**
D	99.4	88.6	91.8	89.5	92.0	100.0	80.0	84.8	**90.8**
E	95.1	71.6	92.5	83.4	76.3	89.6	85.1	76.1	**83.7**
F	98.4	99.4	92.5	89.2	85.2	98.8	94.6	83.7	**92.7**
G	86.8	94.7	86.5	85.0	84.7	71.2	86.3	80.6	**84.5**
H	98.0	73.9	94.1	79.4	77.6	83.9	76.4	72.0	**82.0**
I	99.3	98.3	89.3	84.4	84.2	76.9	*64.9*	78.2	**84.4**
J	100.0	96.4	94.7	94.9	100.0	83.1	*0.0*	78.0	**80.9**
K	93.3	87.7	84.0	86.5	100.0	68.8	86.8	N/A	**75.9**
Avg(%)	**95.1**	**85.1**	**91.5**	**87.7**	**86.8**	**83.6**	**67.8**	**80.0**	**84.7**

Table 5.2: Recognition rate results of the 3D gesture classifier. Cases that performed poor (below 70%) are printed in italics (Abreviations are: Wk — walking, Wv — waving, Cr — crouching, Sl — standing-normal, Sr — standing-star, Sh — standing-stretched, Sit — sitting, Ly — lying down)

nition rate of 90.5%, while that of the poses is 81.2%. Reasons for the lower performance figures are discussed in the remainder of this section.

As in the case of the 2D system, the delayed response of the classifier is one of the factors that impacts on the performance of the 3D system. This is a result of the averaging operations required for feature processing (Equations 5.1 to 5.3) and the classifier's voting system. It causes a gesture to be classified as the previous class during the initial period of the new gesture. During acquisition of testing data, gestures were executed in the sequence *walking*, *waving* and *crouching*. The delayed response phenomenon causes the first 1 to 2 seconds of each of these respective gestures to be classified as *unknown*, *walking* and *waving* as is evident in the classification confusion matrix of Table 5.3. From this table it is also clear that *waving* is at times confused with *crouching*. This occurs if the person does not face any camera and waves in an arms-over-head manner.

Poses are always preceded by one of the dynamic gestures during testing data acquisition. For example, *standing-normal* is preceded by *walking* and *standing-star* by *waving* (to get from *standing-normal* to *standing-star* a wave is performed). As a result of the classifier's delayed response poses are being classified as one of the dynamic

System	True Class (%)							
Classified	Wk	Wv	Cr	Sl	Sr	Sh	Sit	Ly
Wk	**95.1**	1.6	0.9	10.8	1.7	1.3	3.9	3.1
Wv	0.0	**85.1**	5.3	0.0	9.8	9.3	4.1	9.1
Cr	2.4	9.2	**91.5**	0.0	0.7	1.5	7.3	5.9
Sl	1.1	0.5	1.0	**87.7**	1.0	4.1	0.0	0.0
Sr	0.4	2.2	0.5	0.0	**86.8**	0.2	0.0	0.0
Sh	0.4	1.4	0.1	0.6	0.0	**83.6**	0.0	0.0
Sit	0.0	0.0	0.0	0.0	0.0	0.0	**67.8**	1.0
Ly	0.0	0.0	0.0	0.0	0.0	0.0	15.7	**80.0**
Un	0.8	0.1	0.9	0.9	0.0	0.1	1.3	0.9

Table 5.3: Classification confusion matrix of the 3D gesture classifier.

gestures (see confusion matrix). The upright poses (*standing-normal*, *standing-star* and *standing-stretch*) are seldom confused with each other and make up less than 1% of the false positive classifications.

Sitting and *lying down* have the worst performance of all classes. Table 5.2 shows that for person B and J the system did not recognise *sitting* at all. For the particular case *sitting* was classified as *lying down* 15.7% of the time as shown in the confusion matrix. This is caused by the overlap in class distributions (see Figure 5.8(b)). Execution of *crouching* precedes *sitting* and contributed to 7.3% of the false positive classifications of *sitting*. The major cause of false positive classification for *lying down* is the dynamic gesture that precedes it. During test data capturing, *sitting* is the pose that precedes *lying down*. The dynamic gesture that precedes lying down - the activity of lying down - is not defined here and the system should classify it as *unknown*. Most of the false positive classifications in this situation are, however, *waving*, *crouching* and *walking* (in order of contribution).

5.7 Conclusions

This chapter investigated a 3D gesture recognition system that extends the previous 2D system in spatial representation. The results showed that by doing this, the plane motion constraint of the previous system has been removed. The average recognition rate of *walking*, *waving* and *crouching* is 95.6% for the 2D system and 90.5% for the 3D system. If one considers that the extra spatial dimension adds additional complexity to the problem in the case of the 3D system, then 5.1% loss in performance can be tolerated. The pose classes that extend the 2D system's gesture vocabulary also have an acceptable average recognition rate at 81.2%. Facing direction invariance was achieved, making the system truly spatially unconstrained within the limitation of the system itself.

Although the 3D system is more advanced than the 2D system, recognition of dynamic gestures suffers from the same problem experienced by the 2D system, namely delayed response. This is an attribute of the recognition scheme and can cause gestures executed at a high rate to be ignored (see also Section 4.7). This problem is addressed in the next chapter.

Chapter 6

Alternative temporal model

Up to now recognition of gestures has been based on the scheme described in Section 4.4.3. In essence, this technique uses the last ten bounding box derived features and approximates the first principle component of a gesture from its feature distribution. Each gesture has a prominent first principle component that is used to uniquely identify that gesture. This scheme is simple, allows for real-time implementation and has an acceptable recognition rate for the chosen gestures. Unfortunately, it also discards information as a result of averaging operations during feature processing (see Equations 4.1 to 4.3 for the 2D system and Equations 5.1 to 5.3 for the 3D system). This information might be useful for classification purposes, especially if a larger gesture vocabulary is considered. In addition, the recognition scheme also suffers from the delayed classification response pointed out in Sections 4.7 and 5.7.

This chapter investigates an alternative temporal model with the aim of improving the recognition capabilities of the previous system. It starts by stating the chapter objectives in Section 6.1. In Section 6.2 the problem is analysed and alternative approach motivated. The major part of this chapter focuses on the application of the alternative temporal model to the gesture recognition problem, which is discussed in Section 6.3. Section 6.4 looks at the results achieved by the alternative model and the chapter is then concluded with Section 6.5.

6.1 Objectives

The objective here is to *improve* on the temporal model used previously for recognition. The word *improve* refers to an extended and more practical gesture vocabulary compared to the original system and achieving an acceptable recognition rate. The gesture vocabulary is to be extended from the three dynamic gestures of the original system (*walking*, *waving* and *crouching*) to ten. These are *crouching, standing up, waving one arm up, waving one arm down, waving two arms up, waving two arms down, extending single arm, retracting single arm, extending two arms* and *retracting two arms*. Figure 6.1 shows selected frames of each gesture. In this chapter, pose recognition (static gestures) is not considered.

The recognition scheme used previously required that gestures should be executed for 1 to 2 seconds before being recognised (see Section 4.6). Therefore, gestures executed over a period less than that stand a chance of not being detected at all. A more practical gesture set is therefore chosen for testing the alternative recognition scheme. The set listed above can be executed in a period as short as 0.5 seconds and is therefore not constrained in a temporal sense.

It was decided to test the alternative temporal model using the 3D bounding box representation, since it has more practical value than the 2D system. As in the case of the previous 3D system, the aim is also to achieve gesture recognition invariant to facing direction.

6.2 Problem analysis and approach

Section 4.4.3 briefly introduced the problems of recognising the signatures obtained from the bounding box models. To understand the nature of these signatures, an example is used: Figure 6.2(b) shows s, the 3D bounding box perimeter, as a function of time for four example sequences of the single upwards waving gesture shown in Figure 6.2(a). The four signatures are representative of this gesture being executed at various rates. Inspection of the signatures reveals the following:

- **Variable length**: The length of each signature depends on the gesture's execution rate. A high execution rate results in the gesture being completed sooner as compared to a lower execution rate. For example, signature A in Figure 6.2(b) represents waving being executed at a higher rate than the representation of signature B.

- **Warping**: Execution rate can also vary as it is being executed. Signature C of Figure 6.2(b) represents waving initially being executed at a low rate, while the last part of the gesture is executed at a high rate. Signature D represents the opposite case. Due to this inter-execution rate variability a signature can therefore also be warped.

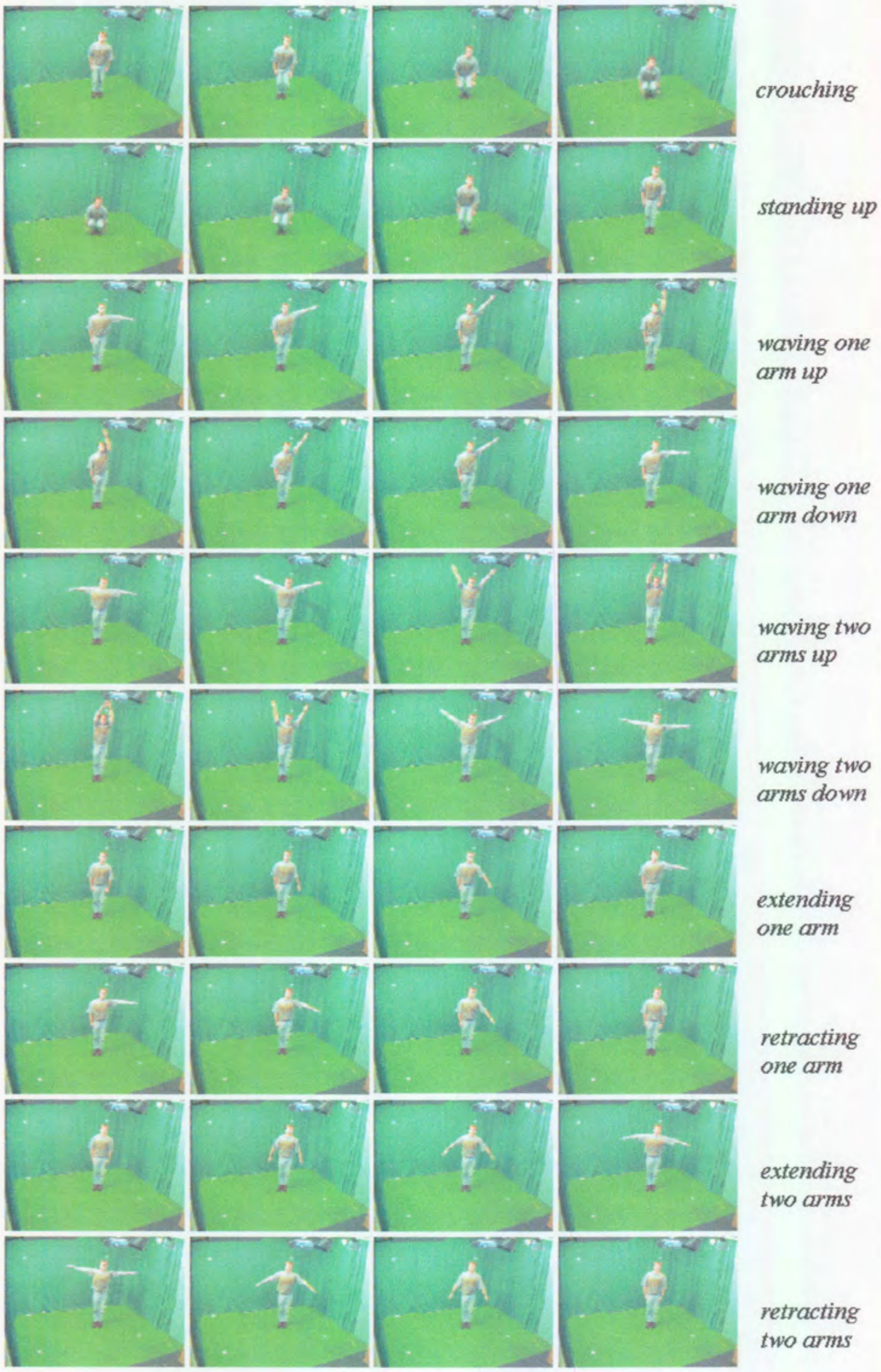


Figure 6.1: Four selected frames of each of ten gestures are shown. Only one camera view is shown here and the person is facing the camera for the purpose of illustration.

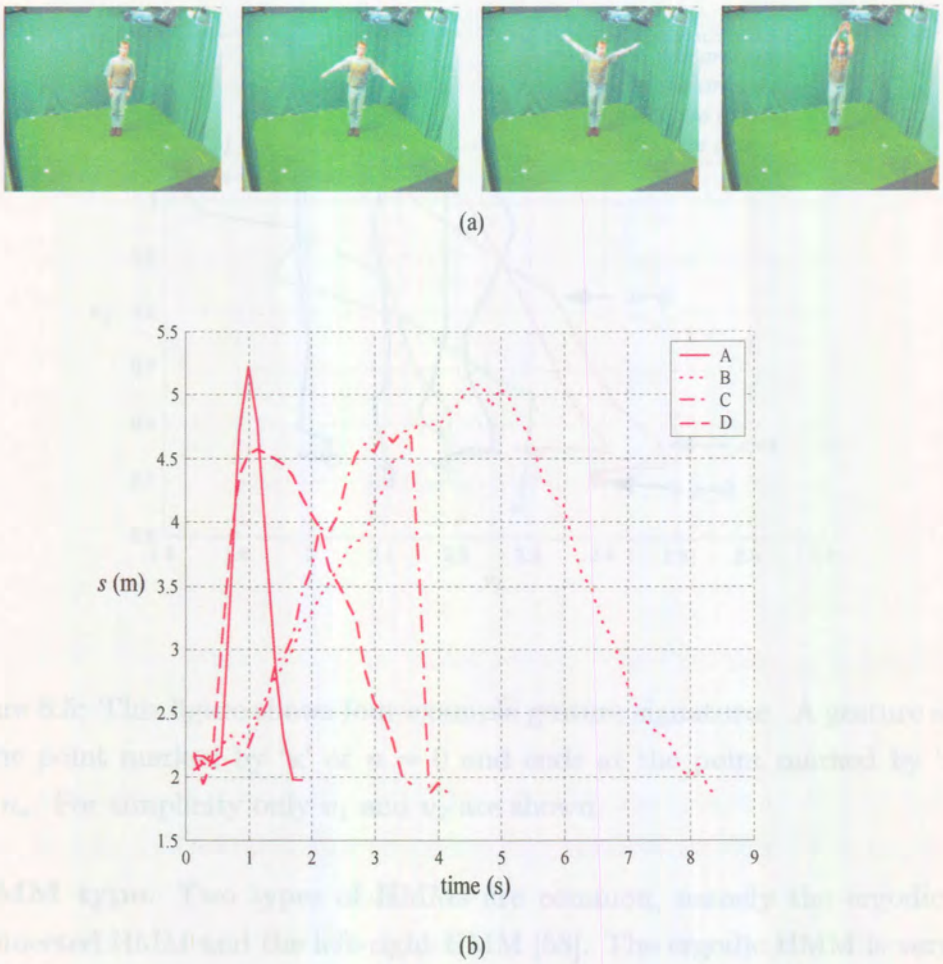


Figure 6.2: (a) shows a person executing an upwards waving gesture and (b) the resulting signatures of a 3D bounding box's perimeter, s. Four signatures are shown in (b) illustrating the variable length and warping properties typical of gestures. 'A' represents the gesture executed at a high rate. 'B' represents a slowly executed gesture. 'C' represents a slowly, then rapidly executed gesture. 'D' represents a rapidly, then slowly executed gesture.

Despite the above properties, all signatures in Figure 6.2(b) have a consistent characteristic: s starts at a low value, increases to an upper value and then decreases again to a lower value. The question is: How can one recognise this characteristic, without being affected by variable length and warping? There are two common techniques to handle this sort of problem, namely hidden Markov models (HMM) and dynamic time warping (DTW). Both were briefly reviewed in Section 2.3.3. Currently HMMs is the preferred way to solve this type of problem - it has a probabilistic framework, can more easily handle continuous data streams and can learn from training data [11]. For these reasons it was decided to pursue HMMs here rather than DTW.

Although HMMs lend themselves to continuous recognition of gestures, only the simpler isolated gesture recognition is considered here. This requires a pause of approximately one second between successive gestures. The pause acts as a cue for the temporal segmentation of gesture data.

6.3 Gesture recognition using HMMs

From a high level processing point of view, the system proposed in this chapter has the same flow diagram as the original 3D system (see Figure 5.2). The differences lie in feature selection and classification. Person tracking and 3D bounding box construction remain the same as discussed in Sections 5.4.1 and 5.4.2 respectively. Feature selection and classification for the alternative temporal model is discussed in the remainder of this section.

6.3.1 Feature selection and conditioning

As in the case of the previous 2D and 3D recognition systems, a feature vector is derived from the bounding box parameters. The parameters of the 3D system are: height (h), perimeter (s), the true centroid ($\mathbf{b}$) and lower centroid ($\mathbf{x}$) as discussed in Section 5.4.2. A gesture can be considered as a sequence of poses. The approach used here to derive a feature vector is similar to that of pose recognition in Section 5.4.3: A feature is based on the ratio of selected model parameters. By using ratios, a degree of invariance to the physical profiles of people are built into the feature. The following three features were chosen:

$$v_1(n) = \frac{h_n}{b_0} \qquad \text{for } n \geq 1 \tag{6.1}$$

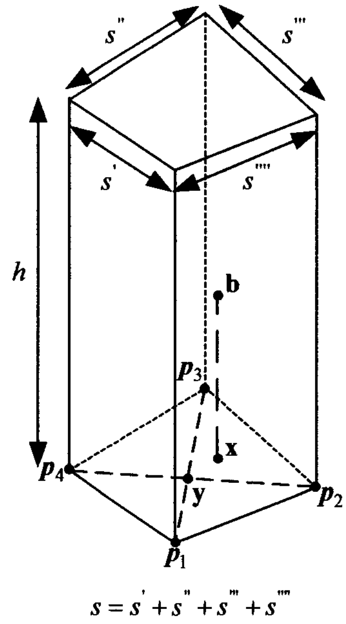


Figure 6.3: This figure shows the bounding box parameters used to construct a feature vector. The bounding box parameters are: height (h), perimeter (s), the true centroid $(\mathbf{b})$, lower centroid $(\mathbf{x})$ and lower vertices intersection $(\mathbf{y})$.

$$v_2(n) = \frac{r_n}{b_0} \qquad \text{for } n \geq 1 \tag{6.2}$$

$$v_3(n) = \frac{|\mathbf{y}_n - \mathbf{x}_n|}{r_0} \qquad \text{for } n \geq 1 \tag{6.3}$$

where b_n and h_n are the true centroid height and bounding box height respectively at frame n. $\mathbf{y}_n$ is the intersection of lines joining the four lower vertices of the 3D bounding box ($\mathbf{p}_1$ to $\mathbf{p}_4$) shown in Figure 6.3. r_n is given by:

$$r_n = \frac{1}{4}\sum_{i=1}^{4} |\mathbf{p}_i - \mathbf{x}| \tag{6.4}$$

r_n is the radius of a cylinder that approximates the bounding box. v_1 is a measure of the height of the pose, v_2 a measure of the width of the pose and v_3 a measure of how off-center the pose is. The purpose of v_3 is to discriminate between gestures that involve the use of a single arm and those that involve the use of two arms. Single arm

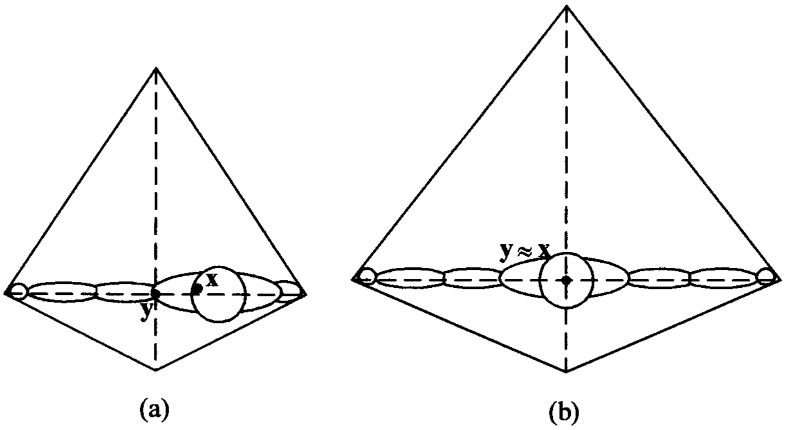


Figure 6.4: (a) and (b) show the plan view of the bounding box of a person extending a single arm and two arms respectively. In (a) the distance between **x** (lower centroid) and **y** (intersection of lower vertices) is significant, while in (b) it is approximately zero. This property is used to discriminate between gestures that make use of a single arm and those that use two arms.

gestures have larger values of v_3, since **x** is biased away from the center of the bounding box (**y**). This is illustrated in Figure 6.4(a) for a person extending a single arm and Figure 6.4(b) where two arms are extended.

Figure 6.5 shows example v_1 and v_2 curves, parameterised by the frame number n for four gestures (*wave single arm up, wave single arm down, extend single arm* and *retract single arm*). These gestures involve the use of a single arm and v_3 is therefore similar and not shown. Gestures start at $n = 0$ and end at $n = n_e$, where n_e is the last frame of the gesture. It is clear from the figure that a gesture starts in a particular region in the feature space and then moves along a path towards an end region. It is this property that an HMM can learn from training examples.

6.3.2 Classification

The next step is to classify unknown sequences. In Section 6.2 it was argued that an HMM might be suitable for this purpose and it was subsequently decided to use HMMs for classification. This is pursued in this section by setting up an HMM for each gesture class. The architecture of an HMM requires some thought, since it is critical for performance and also might have implementation implications. The choice of architecture is based on the following considerations:

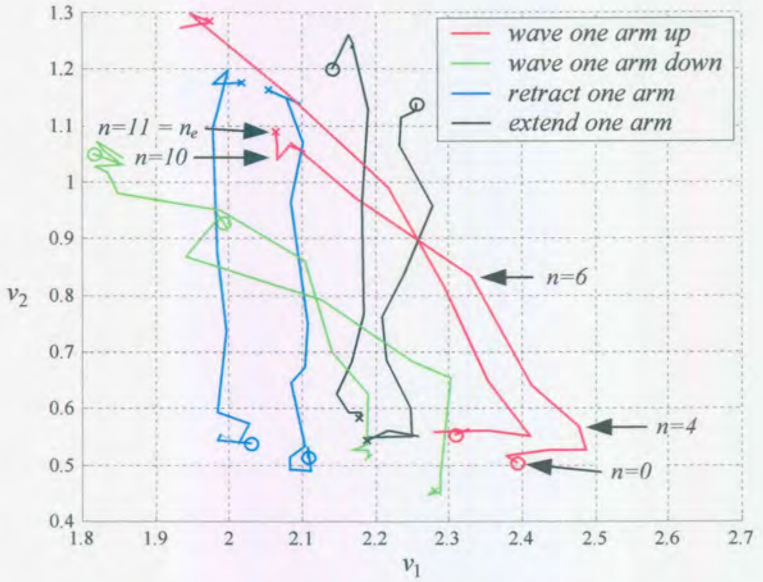


Figure 6.5: This figure shows four example gesture signatures. A gesture starts at the point marked by 'x' or $n = 0$ and ends at the point marked by 'o' or $n = n_e$. For simplicity only v_1 and v_2 are shown.

- **HMM type**: Two types of HMMs are common, namely the ergodic or fully connected HMM and the left-right HMM [58]. The ergodic HMM is very flexible - it allows transition from any state to any other state. A left-right HMM is constrained and only allows state transitions to the current state and previously unvisited states. This property corresponds to the behaviour of the problem considered here and is illustrated by means of an example at the end of this section. Another advantage of a left-right model is that it has a better chance to train optimally, since it has fewer parameters to learn than an equivalent ergodic HMM. A poorly trained model will cause the system to perform suboptimally. Given these arguments it was decided to use a left-right HMM model for the gesture recognition problem. In particular the left-right model chosen allows only the current state or the immediate next state to be visited.

- **State symbol representation**: The choice here is between a discrete and continuous representation. For this problem it was decided to use a continuous observation density, since it does not require a vector quantization procedure often required in the discrete case [41, pp. 111-133]. It also has similar recognition rates when compared to discrete representations in speech applications

[41, pp. 250-255][59]. In gesture recognition applications no direct comparisons could be found in the literature, although both discrete [24, 60] and continuous representations [45, 43, 46] are in use.

- **State density**: The state density type chosen is a Gaussian distribution function. So far there is no evidence that state distributions are Gaussian, but a Gaussian distribution is often assumed in the literature due to its simple mathematical framework to handle multivariate distributions [45, 43, 46]. An extension to the single Gaussian distribution is a mixture of Gaussians [56, pp. 59-73]. Mixture models allow for the modeling of an arbitrary distribution by linearly combining a number of unimodal distributions (for example by combining Gaussian distributions). This technique is often used in speech recognition [41, pp. 175-184]. However, for the current problem a single Gaussian distribution is deemed sufficient.

- **Number of states** (N): Another important model parameter is the number of states. An empirical approach is taken to determine this parameter: Recognition rate is calculated as a function of N for a training set of gestures (training data generation is described in the next section). The optimal N is then the case where the recognition rate is the highest. Figure 6.6 shows the average recognition rate as a function of N for the training data. The graph peaks at 94.3% with $N = 7$ and is therefore the N chosen for all models.

A basic framework for classification using HMMs is now established. The next step is to train the ten HMMs, one for each gesture class. The training procedure used is the segmental k-means algorithm [61]. Given the trained models, an unknown observation sequence is classified by calculating the likelihood that the sequence belongs to each model. The gesture type with the largest likelihood is then selected as the recognised gesture. Likelihood calculation is done by means of the forward procedure [58].

As in the case of the original 2D and 3D systems, *unknown* is not recognised explicitly, but as the absence of the known gestures. The RBF neural network used in the previous systems allows a threshold on its output to be set. *Unknown* is selected if all outputs of the neural network is less than the threshold (see Sections 4.4.4 and 5.4.4). An HMM has two outputs, namely a likelihood of an observation sequence belonging to the model and the state sequence of the observation sequence. Likelihood depends, amongst other factors, on the length of the observation sequence. A fixed likelihood threshold can therefore not be used to qualify *unknown* as in the case of the other classifiers, since observation sequences usually have variable lengths. On the

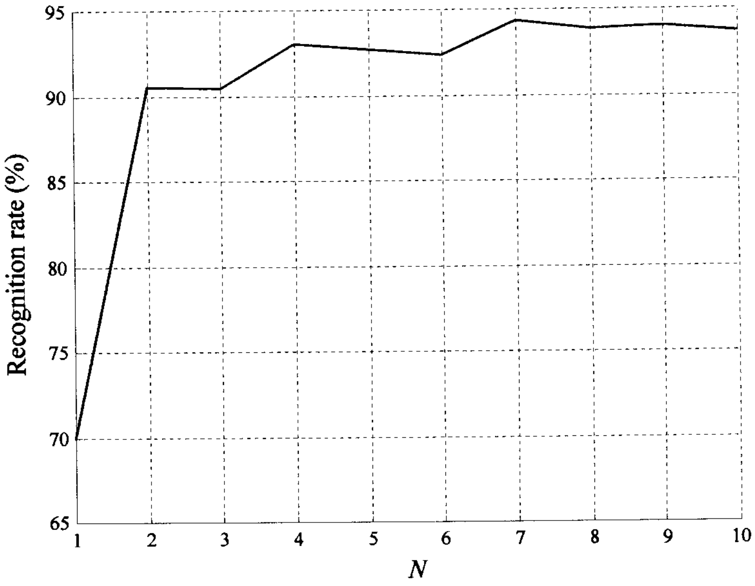


Figure 6.6: This graph shows the average recognition rate of the training data as a function of the number of HMM states (N).

other hand, the state sequence has a consistent behaviour for the purpose of detecting *unknown*. As discussed earlier in this section, the state sequence in a left-right model follows a characteristic sequence if the model and observation sequence is of the same class. This is illustrated in Figure 6.8 for *crouch*. In this figure each state is represented by a mean and a constant likelihood contour[1].

A gesture observation sequence steps from one state to the next as decoded by the Viterbi algorithm [62]. A left-right HMM state sequence always starts at the first state ($j = 0$ in Figure 6.8) and ends at $j = e$, where $e < N$. e depends on the state density parameters and the location of the end point of the observation sequence in the feature space. Therefore, for an observation sequence to belong to a model, it should conform to the following: It should have the highest likelihood value compared to other models and it should step through a minimum of M states as decoded by the Viterbi algorithm. If these conditions are not met, the sequence is classified as

[1] Since a Gaussian density is used to model each state, the mean corresponds to the mean of the Gaussian density. The constant likelihood contour corresponds to a constant Mahalanobis distance from the mean [56, p. 35]. In the case of a Gaussian density this is represented as an ellipse in 2D space. If all three features were to be shown in Figure 6.8, a constant likelihood contour would correspond to an ellipsoid in 3D space. The contour gives an indication of the distribution of the data modeled by the density.

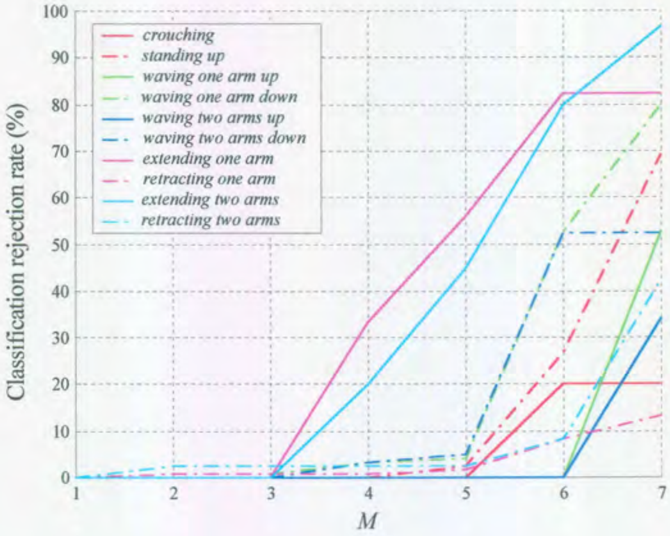


Figure 6.7: This graph plots the classification rejection rate as a function of the minimum number of states stepped through (M).

unknown. M is obtained by calculating the classification rejection rate of the training sequences as a function of M. Classification rejection rate is defined as the number of sequences that are normally positively classified, but are instead classified as *unknown* due to the "minimum step through states" criterion described above. The value of M is then determined independently for each class by means of inspection. Figure 6.7 shows the classification rejection rate as a function of M for the training data of the ten classes. All classes have an acceptable rejection rate for $M = 3$. For larger M values classification rejection rate is class dependent. M was determined by inspection and the chosen values are given in Table 6.1. The criterion used to choose the M values is that a class should not have a rejection rate of more than 5%.

Earlier in this section the left-right type HMM was motivated as a good model for this problem. The reason given was that its behaviour corresponds to the behaviour of the problem. This is illustrated in Figure 6.8 for *crouching*: Each state abstracts a set of poses that are closely related in appearance. The first state represents the standing upright pose up to a point where the knees are slightly bent. The next state represents the poses from where the previous state ended up to a point where the knees are more bent. This continues until all e states have been visited for the particular observation sequence. The execution of a full gesture therefore corresponds to a visit from one state to the next (or the same state if the pose is still part of that state). We as humans

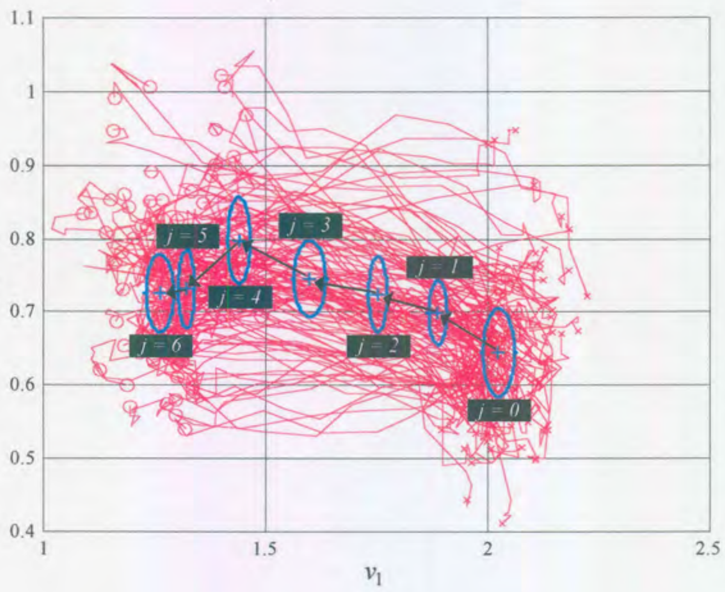


Figure 6.8: This figure shows the 100 training sequences for *crouching*. The start of a sequence is designated by an 'x' and the end by an 'o'. The sequences were used to train a seven state HMM. Each state is represented by a mean value ('+') and a constant likelihood contour (ellipse). A typical gesture observation sequence would step from the first state ($j = 0$) to the last state ($j = e$ where $e < N$).

are physically constrained to execute poses in a set order. In other words, we cannot stand upright and then instantaneously be in a fully crouched position. This constraint therefore matches the constraint of the left-right model and motivates the model type qualitatively.

6.4 Results

The system was trained on five participants and tested on nine participants. Physical profiles of the participants are given in Table 6.2. Training data was generated using participants A to E. Independent testing data was generated using all nine participants. Each class set used for training consisted of 100 observation sequences - 20 observation sequences were captured from each of the five participants who took part in training. During acquisition of training and testing data participants were asked to execute gestures while facing various directions and executing the gestures at different rates.

	Gesture class									
	Cr	St	WU1	WD1	WU2	WD2	Ex1	Rt1	Ex2	Rt2
Min. states (M)	4	4	6	4	6	4	3	4	3	5
Rejection rate (%)	0.0	0.0	0.0	3.3	0.0	3.2	0.0	0.8	0.0	2.5

Table 6.1: The minimum state steps and the resulting classification rejection rate chosen to qualify the *unknown* class. Abreviations are: Cr — crouching, St — standing up, WU1 — waving one arm up, WD1 — waving one arm down, WU2 — waving two arms up, WD2 — waving two arms down, Ex1 — extending one arm, Rt1 — retracting one arm, Ex2 — extending two arms, Rt2 — retracting two arms.

Participant	A	B	C	D	E	F	G	H	I
Height(m)	1.60	1.70	1.85	1.60	1.86	1.74	1.77	1.89	1.50
Weight(kg)	59	74	80	72	73	73	68	94	52

Table 6.2: Physical profiles of the participants used for training and testing the system.

Two gesture recognition systems were implemented: The first system performs offline recognition on manually segmented gestures and is primarily used for training and testing. Manual segmentation is done by an operator who identifies the start and stop frames of each gesture. The second system is an online system that does recognition of live video data in real-time. This system requires the person in the video to pause for about one second before starting with the next gesture. By detecting this pause, it can automatically segment gestures. The online system is demonstrated in videos located on the CD in *chap6\recognition\*.

Testing data consists of a total of 180 observation sequences per gesture class. This was compiled by acquiring 20 observation sequences per gesture for each of the nine participants. To test the system, observation sequences were classified by the system and the verdict compared to the ground truth. The classification results are summarised in Table 6.3 with an average recognition rate of 87.9%.

Another test conducted was to compare the performance of the original system presented in Chapter 5 to that of the alternative system presented in this chapter. As discussed earlier, the original system uses a principle component approximation based technique that discards information as a result of feature processing. By using an HMM-based classifier, better utilisation can be made of the available information in the feature signatures, which should improve performance. To determine if this is the

Person	Gesture recognition rate (%)										Avg.
	Cr	St	WU1	WD1	WU2	WD2	Ex1	Rt1	Ex2	Rt2	
A	100	95	95	100	85	95	90	90	95	90	93.5
B	100	100	90	80	100	75	90	95	100	90	92.0
C	100	100	95	90	100	95	90	100	100	85	95.5
D	100	100	90	75	85	85	70	75	95	85	86.0
E	100	95	75	60	100	90	90	65	100	75	85.0
F	75	100	70	80	90	90	75	70	100	100	85.0
G	100	95	60	85	80	100	85	95	95	100	89.5
H	100	80	70	80	75	100	65	85	60	100	81.5
I	100	95	80	90	90	70	65	65	100	75	83.0
Avg.	97.2	95.5	80.5	82.2	89.4	88.8	80.0	82.2	93.8	88.8	87.9

Table 6.3: Recognition rate (%) results of the alternative 3D gesture classifier. See the caption of Table 6.1 for gesture name abbreviations.

case the two classifiers were compared by testing the original system with the test data generated for the alternative system. The original system was designed to recognise *waving* where both arms were being used and therefore only the test gesture sequences of the alternative system that use two arms were tested on the original system. The gestures of the alternative system corresponds to the following gestures of the original system: *Crouching* and *standing up* of the alternative system corresponds to *crouching* of the original system. Also, *waving two arms up*, *waving two arms down*, *extending two arms* and *retracting two arms* corresponds to *waving* of the original system. The original system classifies gestures frame for frame. In order to compare the classification results of the two systems, the last classification verdict of the last frame in a gesture sequence is chosen as the selected class in the case of the original system. When a new gesture sequence is tested, the classifier is re-intialised, which clears the "memory" of the previous gesture. Running the gesture test sequences on the original 3D system revealed the following results: The recognition rate of *crouching* is 100% and that of *waving* 65.0%. The recognition rate for the alternative system is 96.4% for *crouching* (the average of *crouching* and *standing up*) and 86.3% for *waving* (the average of *waving two arms up*, *waving two arms down*, *extending two arms* and *retracting two arms*). A discussion of the results is given in the next section.

6.5 Discussion

The average recognition rate of 87.9% indicates that the use of HMMs as a recognition scheme has promise. Recognition rates for the individual gestures varies from 80.0% to 97.2%. Four gestures are at the lower end of this range, namely *waving one arm up*, *waving one arm down*, *extending one arm* and *retracting one arm*. All of these gestures involve the use of a single arm. To determine the reason for the lower classification rates, a confusion matrix was compiled and is shown in Table 6.4. According to the matrix, these four gestures are usually confused with one of two gestures: The first confusion is with the same gesture, but involves the use of two arms and the second is a mirror-like gesture (e.g. *extending one arm* has a top-down arm motion and its mirror gesture is *waving one arm down* which has a bottom-up motion). Both these observations account for most of the false classifications and vary from 5.0% to 11.1% per false classification class. The fact that the “one arm” and “two arm” gestures are confused suggests that the feature used to discriminate between them (v_3 of Equation 6.3) does not perfectly separate the classes. False classification as a result of mirror gestures is interpreted as the system not always being able to distinguish between arms moving up and arms moving down. By visually inspecting the 3D bounding box during such gestures it is also hard to recognise the motion direction of the arms. It can therefore not be expected of a machine-based recognition system to recognise this faultlessly, since the information has been lost during feature extraction.

Another observation concerns the recognition rates of the participants: Those that generated the training data have an average recognition rate of 94.3% on the training set. The average rate for the test data of the same group is 90.4% and the average rate of the participants that only took part in testing is 84.8%. This indicates that classification has a dependance on the manner in which gestures are performed and the physical profiles of the participants. This dependence can be reduced by training the system on many more people and perhaps choosing more features.

Invariance with respect to facing direction is one of the objectives of this system. The system performed well under such conditions, the exception being when the person did not face either of the cameras. This covers approximately 15% of all possible facing directions. Under these conditions the bounding box motion of gestures involving the arms is at a minimum and gestures are sometimes incorrectly classified. *Crouching* and *standing up* are not included in this phenomenon and perform well for all facing directions. Another desirable property of the system is that it can tolerate warping and variation in the observation sequence length.

System	**True Class (%)**									
Classified	Cr	St	WU1	WD1	WU2	WD2	Ex1	Rt1	Ex2	Rt2
Cr	**97.2**	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0
St	0.0	**95.5**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
WU1	0.0	0.0	**80.5**	0.0	3.8	0.0	0.0	5.0	0.0	0.0
WD1	0.0	0.0	0.0	**82.2**	0.0	*6.1*	5.0	0.0	0.0	0.0
WU2	0.0	0.0	*8.3*	0.0	**89.4**	0.0	0.0	0.5	0.0	*7.2*
WD2	0.0	0.0	0.0	*7.2*	0.0	**88.8**	0.0	0.0	2.2	0.0
Ex1	0.0	0.0	0.0	*7.2*	0.0	0.5	**80.0**	0.0	1.6	0.0
Rt1	0.0	0.0	*7.7*	0.5	0.0	0.0	0.0	**82.2**	0.0	2.7
Ex2	0.0	0.0	0.0	2.2	0.0	2.7	*11.1*	0.0	**93.8**	0.0
Rt2	1.1	0.0	2.2	0.0	*5.5*	0.0	0.0	*8.8*	0.0	**88.8**
Un	1.6	4.4	0.5	0.5	1.1	1.1	3.8	3.3	2.2	1.1

Table 6.4: Confusion matrix of the alternative 3D gesture classifier. The most confusing classes are in italics. See the caption of Table 6.1 for gesture abreviations.

The *unknown* class (abbreviated as Un in Table 6.4) accounts for an average of 2.0% of overall classifications. An interesting point to note is that the strategy used to recognise *unknown* (discussed in Section 6.3.2) rather classifies a false positive gesture as *unknown* than the incorrect gesture. In other words, observation sequences classified as *unknown* in Table 6.4 are sequences that would have been false positives if the *unknown* recognition strategy was not used. This is the case for all classes except *standing up*, where the 4.4% sequences classified as *unknown* would have belonged to *standing up* if the *unknown* recognition strategy was not used.

It is also interesting to compare the performance of a single state HMM to that of multi-state HMMs (e.g. the seven state model used here). A Markov process is embedded in an HMM, which means that a future observation has a probabilistic dependance on its immediately preceding observation. A single state HMM does not possess this property and is instead analogous to a minimum error classifier with all classes having equal *a priori* probabilities. By comparing the performance of a multi-state HMM to that of a single state, one can get an idea of whether the observation sequences are generated from Markov sources. Inspection of Figure 6.6 reveals that this is indeed the case. The single state HMM system has an average recognition rate of 70.0%, while all multi-state HMMs have rates above 90%.

To conclude, a few remarks about the performance of the original and alternative recognition schemes: The original system slightly outperforms the alternative system in the case of *crouching* (100% versus 96.4%). However, the original system performs poorly for *waving* with an average recognition rate of 65.0% versus the 86.3% of the alternative system. This is explained as follows: It was stated in Section 4.6 that the principle component estimation scheme requires a period of 1 to 2 seconds before it recognises a gesture. Each gesture sequence in the test set generated for the alternative system (also the test set used for the comparison test) is between 0.5 to 2.5 seconds in length. In the case of *crouching* and *standing up*, the average gesture period is 1.6 seconds, while for *waving two arms up*, *waving two arms down*, *extending two arms* and *retracting two arms*, it is about 1.1 seconds. The reason for the longer time periods for crouching related gestures is due to more mass being displaced when these types of gestures are executed. The longer time allows for the principle component estimation recognition scheme to correctly classify the crouching related gestures. However, the waving related gestures are too short and are often not recognised.

6.6 Conclusion

The primary focus of this chapter was to investigate an alternative recognition strategy to the simple technique used for the original 3D system. The scheme chosen for this purpose is the hidden Markov model, which exploits the temporal information in the observation sequences. By utilising the temporal information, the gesture vocabulary is extended to ten gestures, while maintaining an acceptable recognition rate of 87.9%. In addition, gestures of variable length are recognised in the case of the HMM-based classifier. As pointed out in the previous section, short gestures are often not recognised in the case of the original system.

In this chapter progress has been made in improving gesture recognition capabilities. However, the system can still only recognise coarse gestures - a result of the limitations of the bounding box. Also, the system here can only recognise isolated gestures. For a system to be of true practical use, both these issues need to be resolved. Suggestions to achieve this are made in Chapter 7.

Chapter 7

Conclusion

This chapter concludes the study undertaken in this dissertation. First, the work of this study is reviewed in Section 7.1. Research conclusions are made in Section 7.2 and Section 7.3 suggests future work that might enhance current performance.

7.1 Research review

The focus of this study is automatic human gesture recognition by utilising computer vision techniques. Two main paradigms exist in the literature to achieve this, namely configuration-based and motion-based methods. Configuration-based methods represent the human body with a model that resemble the appearance of the real body. In contrast, motion-based methods use motion directly for representation. In this dissertation a novel configuration-based model, the bounding box model, was used to model the human body for the purpose of gesture recognition. The following aspects of this model were investigated:

- **Conceptual behaviour**: The bounding box model has a simple structure and it is expected to have some limitations. These were investigated by means of visual perception experiments in Chapter 3. Human participants were asked to identify one of six gestures while being shown only the bounding box representation of the gesture. It was concluded from the experiments that the model has potential to represent coarse or large movement gestures.

- **2D machine-based operation**: Given a conceptual understanding of the model, it was applied to a computer vision system that automatically recognised gestures in Chapter 4. The simplest possible scenario was chosen: the vocabulary consisted of only four dynamic gestures and a 2D bounding box was used. The 2D bounding box required that motion be constrained to a particular plane and is therefore limited in application. A simple principle component approximation technique was used to recognise gesture signatures derived from the bounding box model. Within this simple framework the system achieved an average recognition rate of 96.3%. This suggests that a bounding box can indeed be used to recognise certain classes of coarse gestures.

- **3D machine-based operation**: We are living in a 3D world. A practical gesture recognition system should be capable of operating in a 3D environment. The bounding box concept lends itself to extension from 2D to 3D. This was the purpose of the 3D gesture recognition system discussed in Chapter 5. It allowed unconstrained body motion as opposed to the plane motion of the 2D system. A person could also execute gestures facing any direction. The gesture vocabulary was extended to eight gestures of which five were poses (static gestures). The system used the same principle component approximation recognition scheme used by the 2D system. The 3D system had an average recognition rate of 84.7% of which the dynamic gesture's recognition rate was 90.5%.

- **Alternative temporal model**: The principle component approximation technique used for recognition in the 2D and 3D systems discarded temporal information in its operation. Chapter 6 explored an alternative recognition scheme that utilised HMMs and explicitly modeled the temporal behaviour. This scheme was integrated with the 3D bounding box model and was able to recognise ten dynamic gestures at an average rate of 87.9%. Gestures could be of variable length and could be executed at a variable rate. In this respect it is better than the 3D system that uses the principle component approximation technique, since rate variability is implicit to the HMM. This is opposed to the neural network classifiers of the original 3D system that learnt the rate variability.

A by-product of the 3D recognition system was the ability to track people. The tracker determined the location of a single person in 3D space relative to a reference. A typical application of tracking is to determine if a person is in the proximity of dangerous equipment that can cause bodily harm.

7.2 Research conclusions

This study investigated various aspects of the bounding box model for the purpose of gesture abstraction and ultimately gesture recognition. Results of the various gesture recognition tests showed that the model has promise in this application. When compared to other models it has the following advantages:

- **Simplicity**: The parameters of the bounding box model are extremely easy to calculate. Given a segmented person, the box is drawn to encapsulate the extremities of the person. This is in contrast to high DOF models which are generally very complex to register and often require nonlinear optimisation techniques.

- **Dimensionality**: The model is easily extended from 2D to 3D. To construct a 3D model, the 2D model of each view is determined *independently*. The 3D model is then constructed by means of stereopsis. Other techniques often rely on a simultaneous solution by using information of all the camera views [23, 63, 17, 47].

- **Startup**: In some systems registration of high DOF models requires the initial pose of the person to be known [14, 17, 47]. Given a known pose, subsequent registration is easier and quicker to determine, since a previous pose is not very different to the succeeding pose. The bounding box model does not suffer from this problem, since it is easy and quick to recalculate the model parameters and this can be done independently of previous calculations.

The bounding box model is not a silver bullet for the gesture abstraction problem. It has the following disadvantages:

- **Coarse gestures**: The price paid for its simplicity is that it only abstracts coarse gestures. High DOF models can represent finer gestures (e.g. nodding of the head), although this capability depends on the complexity and detail of the model. The bounding box model has significantly fewer parameters than high DOF models and it is to be expected that it is limited in its representation capabilities. Depending on the application, this might not necessarily be a problem. For example, it is ideally suited for a system that aims to recognise the hand signals of a traffic pointsman. In such an application the hand movements are coarse, since motorists need to recognise the signals from a distance. Other applications might include recognition of the hand signals of officers who direct taxying aeroplanes or the recognition of sports gestures such as tennis strokes and coarse ballet movements.

- **Segmentation**: The model relies heavily on proper segmentation. In this study the environment was controlled to comply with this requirement through chromakeying. Informal experimentation yielded unreliable results if this requirement is not satisfied. This is not a problem faced only by the bounding box model, however, but by most configuration-based approaches.

- **Occlusion**: A person that is partially or fully occluded is also a cause of unreliable behaviour. Again, this is a general problem faced by all gesture recognition systems and is one that is seldom addressed in the literature. A system capable of handling occlusion on a representation level, but not on a recognition level, is the Pfinder system of MIT [20].

An important aspect of a gesture recognition system is system performance. Popular criteria used in the literature to qualify system performance are recognition rate, gesture vocabulary and execution rate. These criteria are also used to qualify the systems discussed in this dissertation:

- **Recognition rate**: The 2D system, original 3D system and alternative 3D system achieved average recognition rates of 96.3%, 90.5% and 87.9% respectively. The more complex the system and the larger the gesture vocabulary, the lower the recognition rate. The minimum acceptable recognition rate is typically governed by the application.

- **Gesture vocabulary and type**: The vocabulary of the 2D system is perhaps not that practical, but was chosen to test the bounding box model. The gesture vocabulary of the subsequent systems were extended to more meaningful gestures with the alternative 3D system having the largest and perhaps most practical vocabulary. As stated earlier gestures are limited to coarse types and suggestions to address this are given in the next section.

- **Execution rate**: All the systems were implemented as software applications capable of recognising gestures in real-time live video. Frame rate for the 2D system was 14 frames per second and for both the original and alternative 3D systems it was 9 frames per second. These frame rates were achieved on a dual Pentuim III-600 computer running Windows NT. In the case of the 3D systems both processors were utilised - one for each video stream - by employing multithreading techniques. The image size of the video stream was 192 by 144 pixels. Frame rates of all three systems are sufficient for real-time application, although the frame rates of the 3D systems are perhaps on the lower limit of the real-time definition.

The first choice to qualify system performance is to compare it to other systems in the literature based on the above criteria. This is, however, difficult since researchers usually generate test sequences applicable to their particular problem. To compare various systems reference gesture test sequence sets are required similar to what the Lena image is for the image processing community.

7.3 Future research

Future work includes improving on current system performance, extending the gesture vocabulary and working towards a system that can operate in a more practical environment. Suggestions are:

- **Scale space representation**: To improve the spatial representation capability of a bounding box, a scale space representation can be used [2, pp. 31-45]. This fits well with the bounding box paradigm. A 2D bounding box can be seen as the coarsest resolution (highest level) of a scale space representation. The next level of such a representation contains more information about the shape of the object to modeled, or in this case the human body. For example, the next level of a quad tree scale space is a 2 by 2 array. This is the equivalent of dividing the 2D bounding box into a 2 by 2 array, which should lead to a better representation of body structure compared to the bounding box alone. By considering the motion of each cell in the array relative to the level above (that is the bounding box), a better abstraction of gestures might be obtained. Lower levels of the scale space can also be explored, with the lowest level being the pixels of the image.
- **Continuous gesture recognition**: The HMM-based recognition system of Chapter 6 can only recognise isolated gestures. To recognise a continuous stream of gesture data, the system has to automatically segment the stream into separate gestures. A procedure to achieve this is known as level building [42].

The above suggestions are related to the improvement of gesture recognition capabilities. Other work can also be done to make the system more practical. This includes work on segmentation where the aim is to robustly segment people in a realistic environment. Most environments also contain objects such as furniture and often more than one person is present. A practical gesture recognition system has to cope with occlusions caused by these objects and at the same time track multiple people.

Appendix A

Moving bounding box experiment answers

This appendix gives the detailed answers of the eleven people that participated in the Moving Bounding Box experiment. The purpose of this experiment was to qualify the bounding box model as a shape descriptor of the human body and a representation of human gestures. The approach used was to conduct a visual perception experiment on a number of participants. Details of the experiment were given in Chapter 3.

This appendix is organised as follows: Section A.1 gives the answers of Experiment 1 to 3. In the case of Experiment 1 the answers are interpreted as discussed in the section. This is followed by Section A.2 that gives the uninterpreted answers of Experiment 1.

A.1 Answers of Experiments 1 to 3

Tables A.1 to A.6 show the answers of the participants for the 2D and 3D sets of Experiments 1 to 3. The answers of Experiment 1 (Tables A.1 and A.2) are interpreted, since the participant had no prior knowledge of the gesture set. The criteria used for the interpretation was: Is the answer given by a participant for a particular gesture a reasonable description of the actual gesture? If the answer is yes, then it was assumed that the perceived gesture was recognised as the actual gesture. The abbreviations applicable to the tables in this section are: Wk — *walking*, Wv — *arms waving*, Cr — *crouching*, Hd — *hand waving*, In — *inactive*, Nd — *nodding*, Un — *unknown*, Or — *other* (none of the above).

Person	Video sequence							
	Wk	**Nd**	**In**	**Hd**	**Cr**	**Hd**	**Nd**	**Wv**
A	Or	In	In	Un	Cr	In	In	Or
B	Wk	In	Or	Or	Cr	Or	Or	Or
C	Wk	Un	Hd	Un	Cr	Or	In	Or
D	Wk	Nd	Or	Or	Cr	Or	Or	Wv
E	Wk	In	Un	Un	Cr	Un	Un	Or
F	Wk	In	In	In	Cr	In	In	Wv
G	Wk	In	In	In	Cr	In	In	Wv
H	Wk	In	In	In	Cr	In	In	Or
I	Or	Or	Or	Or	Or	Or	Un	Or
J	Wk	Or	Or	Or	Cr	Or	Or	Or
K	Wk	In	In	In	Cr	In	In	Or

Table A.1: Participant answers for the 2D representation of Experiment 1.

Person	Video sequence							
	In	**Wk**	**Hd**	**Nd**	**Wv**	**In**	**Cr**	**Hd**
A	In	Wk	In	In	Or	In	Cr	In
B	In	Wk	In	In	Or	In	Cr	In
C	Un	Wk	Un	In	In	Or	Cr	In
D	In	Wk	In	In	Or	Or	Cr	Or
E	In	Wk	Un	Un	Or	Un	Cr	In
F	In	Wk	In	In	Wv	In	Cr	In
G	In	Wk	In	In	Wv	In	Cr	In
H	In	Wk	In	In	Or	Un	Cr	Un
I	Un	Wk	Or	Or	Or	Or	Cr	Or
J	Or	Wk	Or	Or	Or	Or	Cr	In
K	In	Wk	In	In	Or	In	Cr	In

Table A.2: Participant answers for the 3D representation of Experiment 1.

Person	**Video sequence**							
	Wk	**Nd**	**In**	**Hd**	**Cr**	**Hd**	**Nd**	**Wv**
A	Wk	In	Un	Nd	Cr	Hd	In	Wv
B	Wk	Hd	Hd	Hd	Cr	Nd	Hd	Wv
C	Wk	Hd	Hd	Un	Cr	In	Nd	Wv
D	Wk	Hd	Hd	Nd	Cr	In	Hd	Wv
E	Wk	In	In	Un	Cr	In	Nd	Wv
F	Wk	Hd	Hd	In	Cr	Nd	Hd	Wv
G	Wk	In	In	In	Cr	In	In	Wv
H	Wk	In	In	Hd	Cr	In	Hd	Wv
I	Wk	In	Hd	Hd	Cr	Nd	Hd	Wv
J	Wk	Un	Hd	Wv	Cr	Hd	Nd	Wv
K	Wk	In	Hd	Hd	Cr	Hd	Nd	Wv

Table A.3: Participant answers for the 2D representation of Experiment 2.

Person	**Video sequence**							
	In	**Wk**	**Hd**	**Nd**	**Wv**	**In**	**Cr**	**Hd**
A	In	Wk	Hd	Nd	Wv	Hd	Cr	Nd
B	In	Wk	Hd	In	Wv	Nd	Cr	In
C	In	Wk	Hd	Nd	Wv	Hd	Cr	In
D	In	Wk	Hd	In	Wv	Hd	Cr	Nd
E	In	Wk	Hd	Nd	Wv	Hd	Cr	Nd
F	Hd	Wk	Hd	Nd	Wv	In	Cr	Nd
G	In	Wk	In	In	Wv	In	Cr	In
H	Hd	Wk	Hd	In	Wv	Hd	Cr	Hd
I	Nd	Wk	Hd	Hd	Wv	Hd	Cr	Cr
J	Nd	Wk	Hd	Nd	Wv	Hd	Cr	Hd
K	Nd	Wk	Hd	Nd	Wv	Hd	Cr	In

Table A.4: Participant answers for the 3D representation of Experiment 2.

Person	Video sequence							
	Wk	**Nd**	**In**	**Hd**	**Cr**	**Hd**	**Nd**	**Wv**
A	Wk	Un	Hd	Un	Cr	Un	Un	Wv
B	Wk	Hd	Nd	In	Cr	Nd	In	Wv
C	Wk	Hd	In	Nd	Cr	Hd	Nd	Wv
D	Wk	Hd	In	Nd	Cr	In	Hd	Wv
E	Wk	Hd	Hd	Nd	Cr	Nd	Hd	Wv
F	Wk	In	Hd	Nd	Cr	Hd	Nd	Wv
G	Wk	In	Hd	In	Cr	Nd	In	Wv
H	Wk	In	Hd	Hd	Cr	In	In	Wv
I	Wk	Nd	In	Hd	Cr	Hd	Nd	Wv
J	Wk	In	Hd	Nd	Cr	Hd	In	Wv
K	Wk	In	Nd	Nd	Cr	In	In	Wv

Table A.5: Participant answers for the 2D representation of Experiment 3.

Person	Video sequence							
	In	**Wk**	**Hd**	**Nd**	**Wv**	**In**	**Cr**	**Hd**
A	In	Wk	Hd	Un	Wv	Hd	Cr	Nd
B	Nd	Wk	Hd	In	Wv	Nd	Cr	In
C	Nd	Wk	Hd	In	Wv	Nd	Cr	In
D	In	Wk	Hd	In	Wv	In	Cr	Nd
E	In	Wk	Hd	Nd	Wv	Nd	Cr	Nd
F	Hd	Wk	Nd	In	Wv	In	Cr	Nd
G	In	Wk	Hd	Nd	Wv	In	Cr	Nd
H	Hd	Wk	In	Hd	Wv	Hd	Cr	In
I	Nd	Wk	In	Hd	Wv	In	Cr	Hd
J	Nd	Wk	Hd	Nd	Cr	Hd	Cr	Nd
K	Hd	Wk	Hd	Nd	Wv	In	Cr	Nd

Table A.6: Participant answers for the 3D representation of Experiment 3.

A.2 Uninterpreted answers of Experiment 1

This section gives the uninterpreted answers of participants for Experiment 1:

- Answers of participant A for the 2D test set:

Actual Gesture	Participant Answer
Walking	Walking in circle
Nodding	Standing still
Inactive	Standing still
Hand waving	Unknown
Crouching	Crouching
Hand waving	Standing still
Nodding	Standing still
Arms waving	Turning around on the spot.

- Answers of participant A for the 3D test set:

Actual Gesture	Participant Answer
Inactive	Standing still
Walking	Walking
Hand waving	Standing still
Nodding	Standing still
Arms waving	Jumping
Inactive	Inactive
Crouching	Crouching
Hand waving	Inactive

- Answers of participant B for the 2D test set:

Actual Gesture	Participant Answer
Walking	Walking
Nodding	Standing Still
Inactive	Jumping
Hand waving	Jumping
Crouching	Crouching
Hand waving	Moving one step left and right
Nodding	Lifting leg up and down
Arms waving	Star jumps

- Answers of participant B for the 3D test set:

Actual Gesture	**Participant Answer**
Inactive	Standing still
Walking	Walking
Hand waving	Standing still
Nodding	Standing still
Arms waving	Star jumps
Inactive	Standing still with slight movement
Crouching	Crouching
Hand waving	Standing still

- Answers of participant C for the 2D test set:

Actual Gesture	**Participant Answer**
Walking	Walking
Nodding	Unknown
Inactive	Hands waving
Hand waving	Unknown
Crouching	Crouching
Hand waving	Looking around
Nodding	Standing still
Arms waving	Star jumps

- Answers of participant C for the 3D test set:

Actual Gesture	**Participant Answer**
Inactive	Unknown
Walking	Walking
Hand waving	Unknown
Nodding	Standing still with hand movement
Arms waving	Standing still with foot movement
Inactive	Standing still while lifting foot
Crouching	Crouching
Hand waving	Standing still with slight movement

- Answers of participant D for the 2D test set:

Actual Gesture	Participant Answer
Walking	Walking
Nodding	Nodding head
Inactive	Standing still while moving hand outwards
Hand waving	Turning on the spot
Crouching	Crouching
Hand waving	Tapping foot
Nodding	Slight kicking
Arms waving	Arms waving

- Answers of participant D for the 3D test set:

Actual Gesture	Participant Answer
Inactive	Inactive
Walking	Walking
Hand waving	Inactive
Nodding	Slight swaying body movement
Arms waving	Large kicking movement
Inactive	Moving arms slightly outwards
Crouching	Crouching
Hand waving	Standing on toes and then relaxing

- Answers of participant E for the 2D test set:

Actual Gesture	Participant Answer
Walking	Walking
Nodding	Standing still
Inactive	Unknown
Hand waving	Unknown
Crouching	Crouching
Hand waving	Unknown
Nodding	Unknown
Arms waving	Star Jumps

- Answers of participant E for the 3D test set:

Actual Gesture	Participant Answer
Inactive	Standing still
Walking	Walking
Hand waving	Unknown
Nodding	Unknown
Arms waving	Star Jumps
Inactive	Unknown
Crouching	Crouching
Hand waving	Standing Still

- Answers of participant F for the 2D test set:

Actual Gesture	Participant Answer
Walking	Walking
Nodding	Standing still
Inactive	Standing still
Hand waving	Standing still with slight body movement
Crouching	Crouching
Hand waving	Standing still with slight movement of arms
Nodding	Standing still with slight movement of arms
Arms waving	Arms waving

- Answers of participant F for the 3D test set:

Actual Gesture	Participant Answer
Inactive	Standing still
Walking	Walking
Hand waving	Standing still with slight arm movement
Nodding	Standing still with slight arm movement
Arms waving	Arms waving
Inactive	Standing still with slight body movement
Crouching	Crouching
Hand waving	Standing still with slight body movement

- Answers of participant G for the 2D test set:

Actual Gesture	Participant Answer
Walking	Walking
Nodding	Do nothing
Inactive	Do nothing
Hand waving	Do nothing
Crouching	Crouching
Hand waving	Do nothing
Nodding	Do nothing
Arms waving	Extending arms

- Answers of participant G for the 3D test set:

Actual Gesture	Participant Answer
Inactive	Do nothing
Walking	Walking
Hand waving	Do nothing
Nodding	Do nothing
Arms waving	Extending arms
Inactive	Do nothing
Crouching	Crouching
Hand waving	Do nothing

- Answers of participant H for the 2D test set:

Actual Gesture	Participant Answer
Walking	Walking
Nodding	Inactive
Inactive	Inactive
Hand waving	Inactive
Crouching	Crouchning
Hand waving	Inactive
Nodding	Inactive
Arms waving	Star jumps

- Answers of participant H for the 3D test set:

Actual Gesture	**Participant Answer**
Inactive	Inactive
Walking	Walking
Hand waving	Inactive
Nodding	Inactive
Arms waving	Star jumps
Inactive	Unknown
Crouching	Crouchnig
Hand waving	Unknown

- Answers of participant I for the 2D test set:

Actual Gesture	**Participant Answer**
Walking	Jumping to side while moving arms
Nodding	Jumping up and down
Inactive	Jumping up and down
Hand waving	Jumping up and down
Crouching	Bowing up and down
Hand waving	Standing on toes, then relaxing with minimal movement
Nodding	Unknown
Arms waving	Star jumps

- Answers of participant I for the 3D test set:

Actual Gesture	**Participant Answer**
Inactive	Unknown
Walking	Walking
Hand waving	Jogging on the spot
Nodding	Jogging on the spot
Arms waving	Star jumps
Inactive	Jogging on the spot
Crouching	Crouching
Hand waving	Jogging on the spot

- Answers of participant J for the 2D test set:

Actual Gesture	Participant Answer
Walking	Walking
Nodding	Standing on toes, then relax
Inactive	Moving arms outwards
Hand waving	Swaying from left to right
Crouching	Crouching
Hand waving	Moving arms outward
Nodding	Standing on toes, then relax
Arms waving	Star jumps

- Answers of participant J for the 3D test set:

Actual Gesture	Participant Answer
Inactive	Standing on toes with slight arms movement
Walking	Walking
Hand waving	Slight dancing
Nodding	Slight dancing
Arms waving	Slight jumping with arm movement
Inactive	Standing on toes, then relax
Crouching	Crouching
Hand waving	Slight torso movement

- Answers of participant K for the 2D test set:

Actual Gesture	Participant Answer
Walking	Walking
Nodding	Standing still
Inactive	Standing still
Hand waving	Standing still with slight body movement
Crouching	Crouching
Hand waving	Standing still with slight body movement
Nodding	Standing still with slight body movement
Arms waving	Star jumps

- Answers of participant K for the 3D test set:

Actual Gesture	Participant Answer
Inactive	Standing still with slight body movement
Walking	Walking
Hand waving	Standing still with slight body movement
Nodding	Standing still with slight body movement
Arms waving	Star jumping
Inactive	Standing still with slight body movement
Crouching	Crouching
Hand waving	Standing still with slight body movement

Appendix B

Camera calibration and 3D object tracking

This appendix discusses the implementation details and theory required to track a point object and ultimately a person in space. The technique identified in Chapter 5 to achieve this is stereopsis. Stereopsis is concerned with the 3D reconstruction of an object given two or more different perspective views of the object. It consists of two main problems, namely correspondence and reconstruction[1, p. 140]. The focus in this appendix is on reconstruction. As discussed in Chapter 3, correspondence is implied by registration of the model to the person in the image and is therefore not a requirement here.

Stereopsis employs a number of cameras[1] as a measuring or ranging device by expressing the location of a feature or object in a Cartesian reference frame. To realise this, the cameras need to be calibrated and this is the first item discussed here. Next, given that a distinguishable feature or point is visible on the object in the camera views, the location of the point can be determined. This is the building block required for 3D object reconstruction.

Also presented here, is an algorithm to correct for the apparent rotation phenomenon observed in images of arbitrary orientated cameras. This phenomenon impacts on the accuracy of the reconstructed 3D bounding box and ultimately causes a lower recognition rate if ignored.

[1] Only two cameras are considered here.

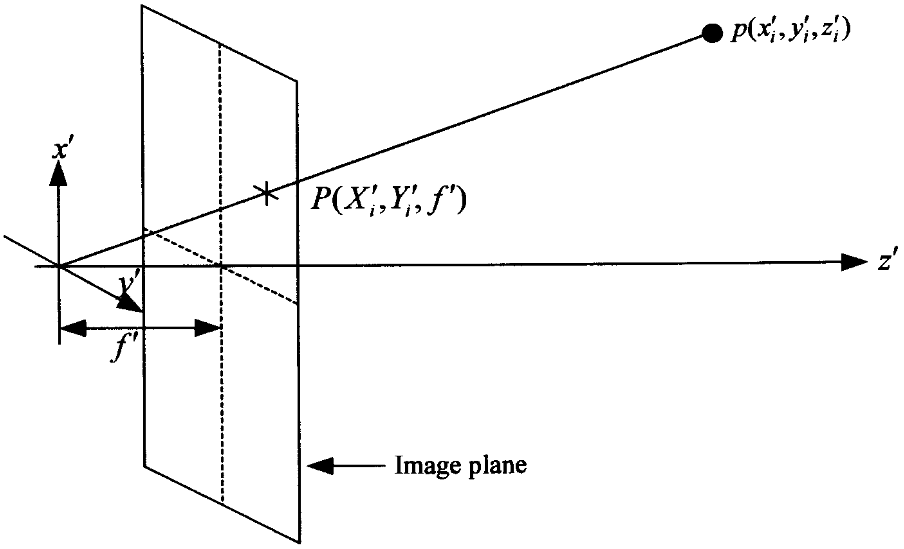


Figure B.1: Pinhole model of a camera. The object at $p(x_i', y_i', z_i')$ is projected onto the image plane of the camera located at a focal distance of $f'$from the pinhole.

B.1 Camera calibration

Camera calibration parameters are the rotation matrix ($\mathbf{R}$), translation matrix ($\mathbf{t}$) and focal length (f). $\mathbf{R}$ and $\mathbf{t}$ identify a camera's orientation and position in space respectively. f is a parameter that scales the object in the image. The following subsections describe how these parameters can be calculated.

B.1.1 Camera model

First the projection of an object in 3D space onto a 2D plane is described. Figure B.1 shows a representation of a pinhole model commonly encountered in the computer vision literature [1, pp. 26-27][57]. A ray can be traced from point i on the object $p(x_i', y_i', z_i')$ to the representation origin[2]. At a distance of f 'it intersects the image plane at $P(X_i', Y_i', f')$. This is the observed location of the point on the image plane and is expressed by:

$$\frac{X_i'}{f'} = \frac{x_i'}{z_i'} \tag{B.1}$$

[2] A particular view is identified with a $'$ symbol. A single $'$ represents the left camera and $''$ the right camera.

$$\frac{Y_i'}{f'} = \frac{y_i'}{z_i'} \tag{B.2}$$

A property of these equations is that information is reduced from 3D to 2D. All points along the line Pp have the same 2D plane projection. It is therefore not possible to recover the unique location of a point in 3D space given only its projection on a 2D camera plane. One way to solve this problem is by triangulating (see Figure B.2), which requires at least two cameras. Equations B.1 and B.2 are expressed relative to a particular camera's coordinate system. In order to successfully triangulate, the relation between the coordinate systems of the two cameras, (x_i', y_i', z_i') and (x_i'', y_i'', z_i''), need to be determined. This is achieved by expressing the coordinate systems of the cameras[3] relative to a reference point in space such as a corner of the room[4]. The next section describes how different reference axes are related.

B.1.2 Transforming between reference axes

A point i is related in two different reference axes by rotating and displacing the one axis relative to the other [1, pp. 35-36][57]. This is expressed by:

$$\begin{pmatrix} x_i' \\ y_i' \\ z_i' \end{pmatrix} = \mathbf{R}' \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \mathbf{t}' \tag{B.3}$$

where $\mathbf{R}'$ is a 3×3 matrix known as the rotation matrix and $\mathbf{t}'$ a 3-element vector known as the translation vector. Referring to Figure B.2, the relation between the world or room reference axis and that of the left camera view is expressed by Equation B.3 for point (x_i, y_i, z_i). The same relationship is expressed for the right camera view by using the appropriate notation discussed earlier.

$\mathbf{R}$ effectively rotates all points in the original reference coordinate system. This is achieved by projecting these points on the new reference axis, which is represented by three orthonormal vectors (rows of $\mathbf{R}$). An orthonormal vector adheres to the following properties:

$$r_{11}^2 + r_{12}^2 + r_{13}^2 = 1 \tag{B.4}$$

[3] The camera coodinate system is also known as the view coordinate system.

[4] This is known as the world coordinate system.

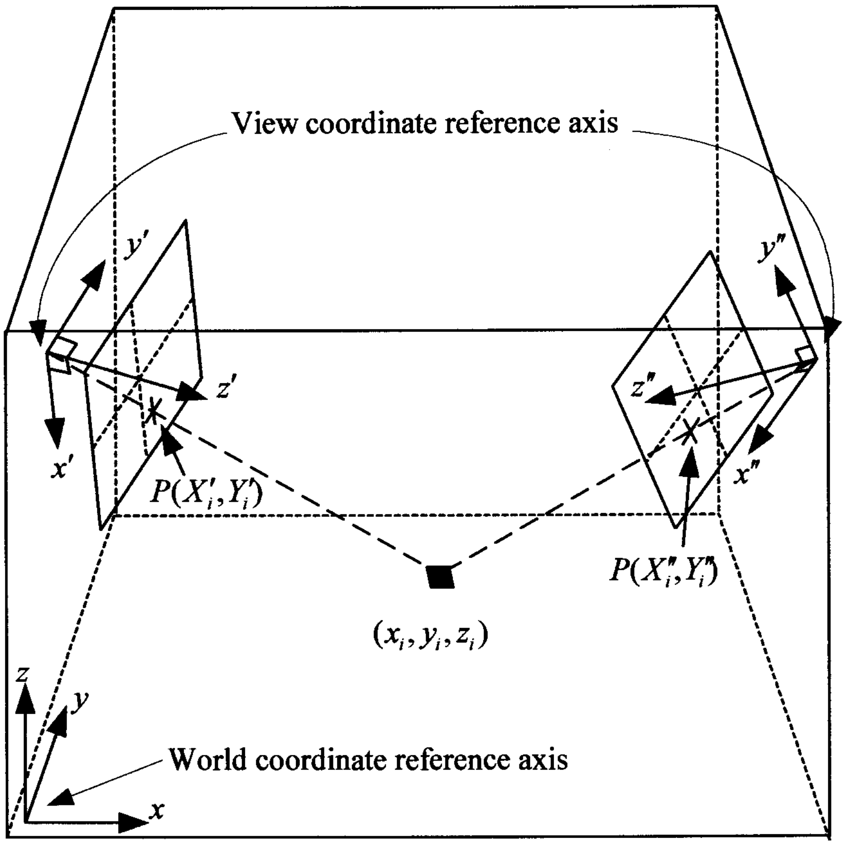


Figure B.2: Reference axis of the room and two views (cameras)

$$r_{21}^2 + r_{22}^2 + r_{23}^2 = 1 \tag{B.5}$$

$$r_{31}^2 + r_{32}^2 + r_{33}^2 = 1 \tag{B.6}$$

The above is required for the vectors to be unit size and the next set of equations forces the vectors to be normal:

$$r_{22}r_{33} - r_{23}r_{32} = r_{11} \tag{B.7}$$

$$r_{23}r_{31} - r_{21}r_{33} = r_{12} \tag{B.8}$$

$$r_{21}r_{32} - r_{22}r_{31} = r_{13} \tag{B.9}$$

R and **t** have twelve unknowns. This reduces to six if the above orthonormal property is taken into account.

B.1.3 Calculating the calibration parameters

To calculate the calibration parameters (**R**, **t**, f) Equations B.1, B.2 and B.3 are used. First, Equations B.1 and B.2 are substituted in Equation B.3:

$$z_i \frac{X_i}{f} = x_i = r_{11}x_i + r_{12}y_i + r_{13}z_i + t_1 \tag{B.10}$$

$$z_i \frac{Y_i}{f} = y_i = r_{21}x_i + r_{22}y_i + r_{23}z_i + t_2 \tag{B.11}$$

$$z_i = r_{31}x_i + r_{32}y_i + r_{33}z_i + t_3 \tag{B.12}$$

Substituting the third equation into the first two and making X_i and Y_i the objects of the respective equations, leaves:

$$X_i = \frac{(r_{11}x_i + r_{12}y_i + r_{13}z_i + t_1)f}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_3} \tag{B.13}$$

and

$$Y_i = \frac{(r_{21}x_i + r_{22}y_i + r_{23}z_i + t_1)f}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_3} \tag{B.14}$$

The above two equations relate the world coordinates (x_i, y_i, z_i) and view coordinates (X_i, Y_i) for point i as a function of the camera's calibration parameters (**R**, **t** and f). In total there are 7 calibration parameters to be solved given that use is made of **R**'s properties as depicted in equations B.4 and B.9. Given four known (x_i, y_i, z_i) calibration points and their corresponding (X_i, Y_i) projections it is possible to solve for **R**, **t** and f using Equations B.13 and B.14. Unfortunately these equations are nonlinear and require nonlinear techniques to solve. Such techniques is described by Wolf [64] and Ganapathy [65]. Ganapathy has also noted that these equations can be solved using linear techniques by increasing the number of calibration markers. Equations B.13 and B.14 are respectively rewritten as:

$$\begin{aligned} -X_i \; = \; & \left[\frac{r_{31}}{t_3}\right] X_i x_i + \left[\frac{r_{32}}{t_3}\right] X_i y_i + \left[\frac{r_{33}}{t_3}\right] X_i z_i - \left[\frac{r_{11}f}{t_3}\right] x_i - \left[\frac{r_{12}f}{t_3}\right] y_i - \\ & \left[\frac{r_{13}f}{t_3}\right] z_i - \left[\frac{t_1 f}{t_3}\right] \end{aligned} \tag{B.15}$$

and

$$\begin{aligned} -Y_i \; = \; & \left[\frac{r_{31}}{t_3}\right] Y_i x_i + \left[\frac{r_{32}}{t_3}\right] Y_i y_i + \left[\frac{r_{33}}{t_3}\right] Y_i z_i - \left[\frac{r_{21}f}{t_3}\right] x_i - \left[\frac{r_{22}f}{t_3}\right] y_i - \\ & \left[\frac{r_{23}f}{t_3}\right] z_i - \left[\frac{t_1 f}{t_3}\right] \end{aligned} \tag{B.16}$$

where the parameters to be solved are written in square brackets. In the equations there are 11 unknown parameters. To solve the unknowns, six calibration markers are needed. This results in twelve equations when substituted into the above equations and their unknowns are determined by linear techniques. Once this is done, the calibration parameters are determined by substituting the solved unknowns into equations B.4 to B.9.

B.2 Object tracking

Once both cameras are calibrated, it is possible to track an object. By tracking it is meant that the object's position is calculated relative to a given reference frame in space (e.g. world reference frame). In other words the aim is to solve for $(x_i,\ y_i,\ z_i)$ of Figure B.2 for each video frame pair. The first step is to substitute equation B.1 and B.2 into equation B.3 and make $(x_i,\ y_i,\ z_i)$ the object of the equation:

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \mathbf{R}^{'-1} \left(z_i^{'} \begin{pmatrix} \frac{X_i^{'}}{f^{'}} \\ \frac{Y_i^{'}}{f^{'}} \\ 1 \end{pmatrix} - \mathbf{t}^{'} \right) \tag{B.17}$$

The above equation is for the left camera. The equivalent equation for the right camera is:

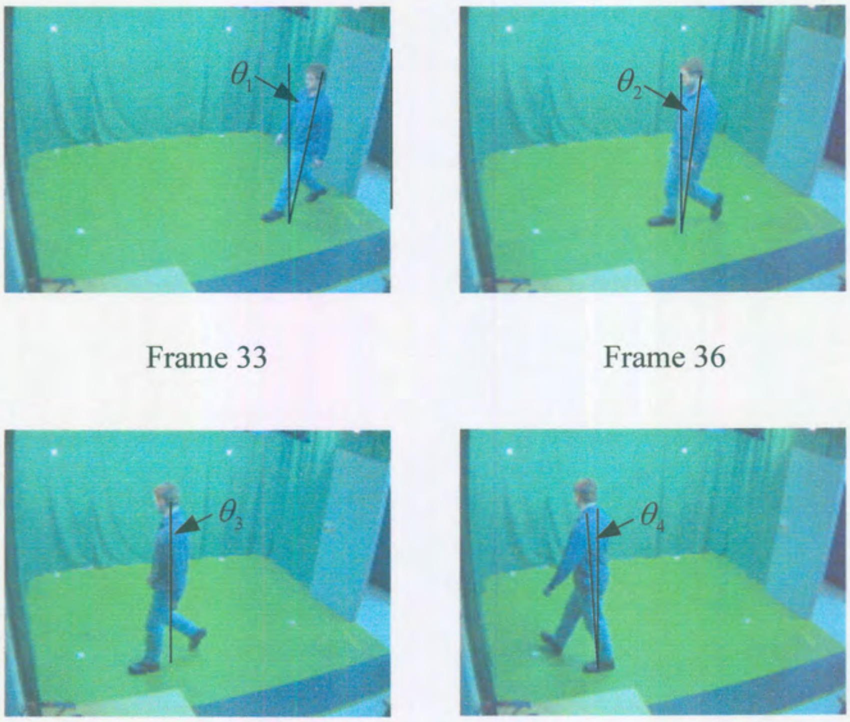
$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \mathbf{R}^{''-1} \left(z_i^{''} \begin{pmatrix} \frac{X_i^{''}}{f^{''}} \\ \frac{Y_i^{''}}{f^{''}} \\ 1 \end{pmatrix} - \mathbf{t}^{''} \right) \tag{B.18}$$

The unknowns in the equations are $z_i^{'}$, $z_i^{''}$ and $(x_i,\ y_i,\ z_i)$. To solve $z_i^{'}$ and $z_i^{''}$ the right sides of Equations B.17 and B.18 are set equal and $z_i^{'}$ and $z_i^{''}$ can then be determined by solving for them simultaneously. $(x_i,\ y_i,\ z_i)$ can then be calculated by substituting $z_i^{'}$ or $z_i^{''}$ back into Equation B.17 or B.18 respectively.

The above tracking algorithm is applicable to a point object. In order to track an object, one has to select a point on the object of each view ($(X_i^{'},\ Y_i^{'})$ and $(X_i^{''},\ Y_i^{''})$) that is used as a tracking reference point. In 5.4.1 this point was chosen to be the lower centroid where the objective was to track people. This can, however be any point as long as it is consistently the same point from frame to frame.

B.3 Orientation correction

Figure B.3 illustrates the apparent rotation phenomenon described in Section 5.4.2. As the person moves from right to left, he rotates relative to the view of the camera. The original bounding box model proposed in Chapter 3 assumes that the person is always in an upright position in the camera images. For practical reasons the cameras of the 3D system (Chapter 5) were mounted nonlevel and near the ceiling, which is the cause



Frame 33

Frame 36

Frame 40

Frame 43

Figure B.3: Illustration of the apparent rotation phenomenon of selected frames in a video sequence. As the person moves from right to left his orientation relative to the upright orientation changes. This is shown by the fact that $\theta_1 > \theta_2 > \theta_3 > \theta_4$.

of the apparent rotation phenomenon. If not corrected, it causes error in calculating the person's world position and construction of the 3D bounding box.

It is possible to correct for this by means of the following algorithm:

1. Determine the lower centroid projections of the person in each camera view ($(X_L^{'}, Y_L^{'})$ and $(X_L^{''}, Y_L^{''})$).

2. Calculate the world coordinates of the lower centroid by using $(X_L^{'}, Y_L^{'})$ and $(X_L^{''}, Y_L^{''})$ as input to the tracking algorithm of the previous section. The world coordinates of the lower centroid is designated by (x_L, y_L, z_L). So far the standard tracking procedure has been used.

3. Define a point (x_U, y_U, z_U) where $x_U = x_L$, $y_U = y_L$ and $z_U = z_L + h$. h is any value larger than 0 and is chosen to be 1 meter. (x_U, y_U, z_U) is therefore a point directly above (x_L, y_L, z_L) as shown in Figure B.4.

4. The points $(X_U^{'}, Y_U^{'})$ and $(X_U^{''}, Y_U^{''})$ are calculated next by projecting (x_U, y_U, z_U) onto each camera's image plain. This is achieved by using equations B.13 and B.14. The relative orientations are expressed as the two-element vectors $\mathbf{p}^{'}$ and $\mathbf{p}^{''}$ for the left and right camera view respectively. $\mathbf{p}^{'}$ (and similarly $\mathbf{p}^{''}$) can be calculated by: $\mathbf{p}^{'} = (X_U^{'}, Y_U^{'}) - (X_L^{'}, Y_L^{'})$.

5. Given the vectors $\mathbf{p}^{'}$ and $\mathbf{p}^{''}$, the angles of relative rotation ($\phi^{'}$ and $\phi^{''}$) for the person in each view are calculated by:

$$\phi^{'} = \arctan\left(\frac{p_1^{'}}{p_2^{'}}\right) \tag{B.19}$$

and similarly for $\phi^{''}$.

This equation gives the angle of rotation relative to the y-axis for the particular view and is used to counter rotate the bounding box. The bounding box properties (e.g. width, height, centroid) are then recalculated for the rotated box.

Step 1 uses an uncorrected bounding box to calculate $(X_L^{'}, Y_L^{'})$ and $(X_L^{''}, Y_L^{''})$. This causes an error when calculating $\phi^{'}$ and $\phi^{''}$. To reduce the error, steps 1 to 5 are repeated iteratively with the latest rotated bounding box used as input to step 1 for the new iteration. It has been found that the algorithm converges quickly and that one iteration is sufficient.

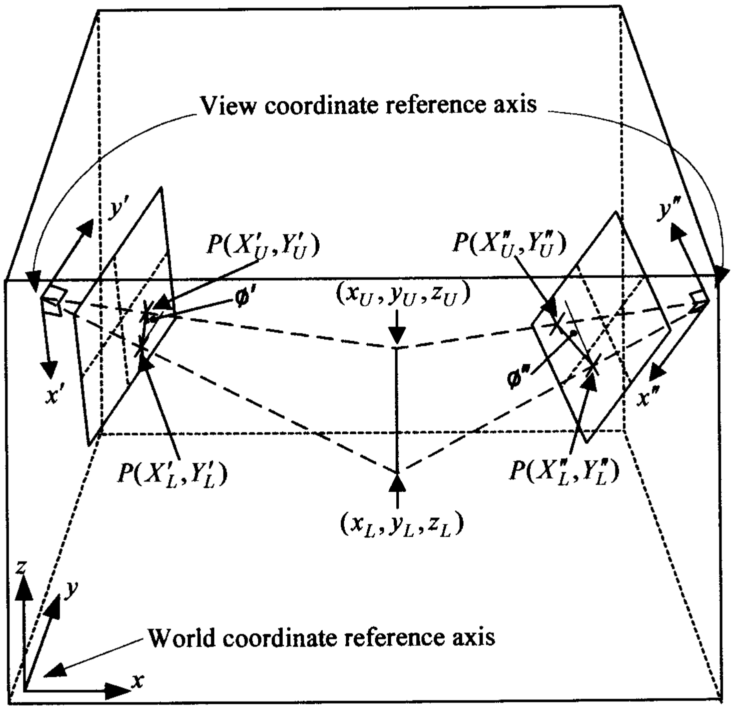


Figure B.4: This figure illustrates the projection of a vertical line onto the two camera views. The line is vertical in the real world, but appears slanted in camera views. The angles between the projection and camera view's vertical position (ϕ' and ϕ'') are used to correct for orientation of the bounding box.

B.4 Results

The calibration and tracking algorithms are used in the 3D gesture recognition and tracking system of Chapter 5. To ensure that the implementation is fault free, two experiments were designed to test it. The following sections describe the experiments and their results.

B.4.1 Verifying R and t

A very simple test can be conducted to verify that **R**, **t** and f are correct. The calibration markers used to calibrate the cameras (see Section B.1.3) have known coordinates in the world reference frame. By using equation B.3, the markers' locations in the left and right camera views can be verified. Figure B.5(a) and (b) show the markers as seen in the two camera images as well as their calculated locations using equation B.3. Table B.1 lists numerical values of the true and calculated coordinates and the error. Error is defined as the difference between calculated and true values and is given in

Marker	True coordinate	Calculate coordinate	Error (pixels)
1a	(16,31)	(16,32)	1
2a	(27,92)	(27,92)	0
3a	(46,131)	(47,92)	1
4a	(149,96)	(150,96)	1
5a	(108,70)	(108,70)	0
6a	(103,15)	(103,17)	2
1b	(72,40)	(73,40)	1
2b	(78,90)	(78,90)	0
3b	(43,120)	(43,120)	0
4b	(141.135)	(141,135)	0
5b	(154,96)	(154,96)	0
6b	(154,36)	(154,36)	0

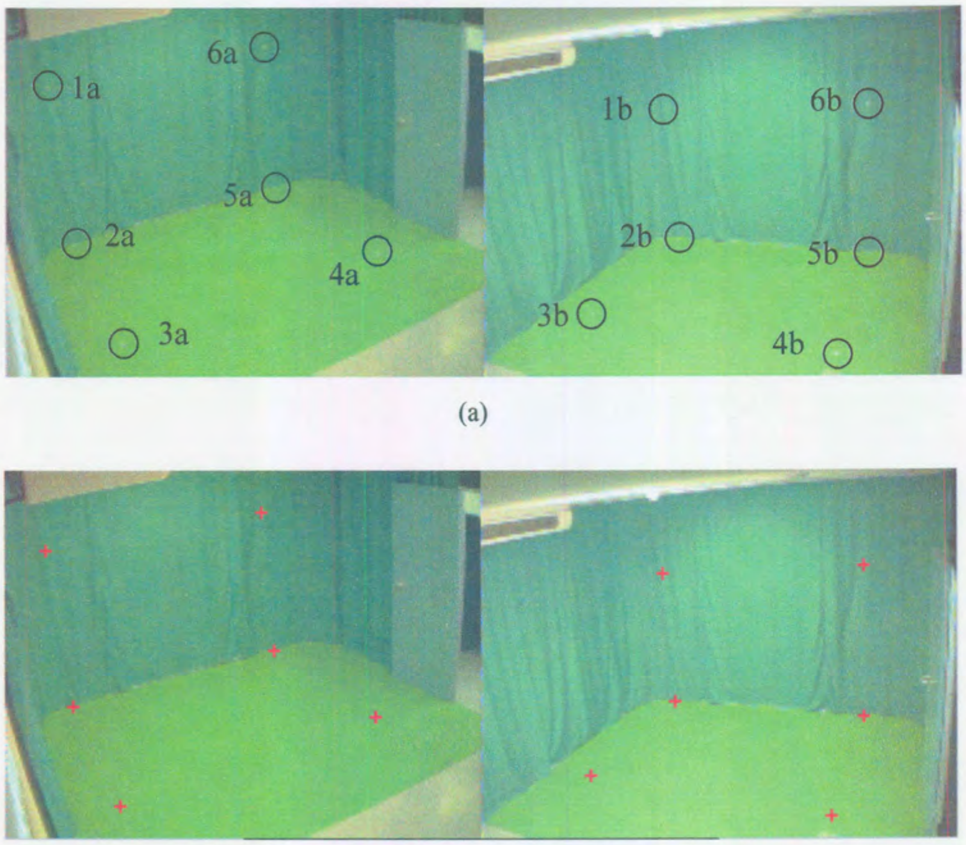
Table B.1: Observed and calculated values for calibration markers

number of pixels. It is clear from the low error values in the table that the calibration parameters are accurate.

B.4.2 Tracking a thin pole

In the previous section it was verified that the cameras are properly calibrated. The next step is to verify that an object can be tracked using the method described earlier. For this experiment a thin pole is chosen since its location can be determined accurately in practice. It is also easier to segment as compared to point-like objects such as a ping-pong ball. The bottom most point of the pole was chosen to be the point to be tracked. The pole was placed at seven known locations and its position calculated. Table B.2 summarises the results of this experiment.

Errors in the z-direction are the most profound and contribute mostly to the overall error. This is to be expected since the cameras are elevated at about 2.5 meters and looking downwards, resulting in a reduced measurement resolution in the z-direction. It can safely be assumed that a higher image resolution will result in better tracking resolution.



(a)

(b)

Figure B.5: (a) Location of markers as viewed from the cameras. (b) Location of markers after being calculated by using Equation B.3.

Location #	Calculated Position	Real Position	Error (m)
1	(2.00,3.10,-0.05)	(2.00,3.11,0.00)	0.05
2	(2.01,4.17,-0.09)	(2.00,4.09,0.00)	0.12
3	(2.01,5.17,-0.09)	(2.00,5.07,0.00)	0.13
4	(0.50,4.62,-0.07)	(0.52,4.59,0.00)	0.08
5	(1.01,3.61,-0.06)	(1.01,3.60,0.00)	0.06
6	(3.40,4.53,0.02)	(3.46,4.59,0.00)	0.09
7	(2.97,3.59,-0.04)	(2.97,3.60,0.00)	0.04

Table B.2: Results of the pole tracking experiment

Bibliography

[1] E. Trucco and A. Verri, *Introductory techniques for 3D computer vision.* New Jersey, NY, USA: Prenctice Hall, 1998.

[2] H. Wechsler, *Computational vision.* San Diego CA, USA: Academic Press, 1990.

[3] R. Schalkoff, *Digital image processing and computer vision.* New York, USA: John Wiley & Sons, 1989.

[4] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images - a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 76, no. 8, pp. 917–935, 1988.

[5] L. Wiskott and C. von der Malsburg, "A neural system for the recognition of partially occluded objects in cluttered scenes: A pilot study," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 4, pp. 935–948, 1993.

[6] R. Würtz, "Neural networks as a model for visual perception: What is lacking?," *Cognitive Systems*, vol. 5, pp. 103–112, June 1999.

[7] C. Cedras and M. Shah, "Motion-based recognition," *Image and Vision Computing*, vol. 13, pp. 129–155, March 1995.

[8] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception and Psychophysics*, vol. 14, no. 2, pp. 210–211, 1973.

[9] G. Johansson, "Visual motion perception," *Scientific American*, pp. 76–88, June 1976.

[10] S. Sumi, "Upside-down presentation of the Johansson moving light-spot pattern," *Perception*, vol. 13, pp. 283–286, 1984.

[11] D. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[12] T. Calvert and J. Chapman, "Aspects of the kinematic simulation of human movement," *IEEE Computer Graphics and Applications*, vol. 2, no. 9, pp. 41–49, 1982.

[13] Z. Chen and H.-J. Lee, "Knowledge-guided visual perception of 3D human gait from a single image sequence," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 2, pp. 336–342, 1992.

[14] J. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: An application to human hand tracking," in *European Conference on Computer Vision*, (Stockholm, Sweden), pp. 35–46, 1994.

[15] R. Conzalez and R. Woods, *Digital image processing.* Reading MA, USA: Addison-Wesley, 1992.

[16] V. Filova, F. Solina, and J. Lenarcic, "Automatic reconstruction of 3D human arm motion from a monocular image sequence," *Machine Vision and Applications*, vol. 10, no. 5, pp. 223–231, 1998.

[17] S. Iwasawa, J. Ohya, K. Takahashi, T. Sakaguchi, S. Kawato, K. Ebihara, and S. Morishima, "Real-time, 3D estimation of human body postures from trinocular images," in *IEEE International Workshop on Modelling People*, (Corfu, Greece), pp. 3–10, 1999.

[18] J. Segen and S. Kumar, "Fast and accurate 3D gesture recognition interface," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 86–91, 1998.

[19] L. Campbell and A. Bobick, "Recognition of human body motion using space constraints," in *Fifth International Conference on Computer Vision*, (Cambridge MA, USA), pp. 624–630, 1995.

[20] C. R. Wren, A. Azarbayejani, T. Darrel, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[21] J. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, 1999.

[22] K. Rohr, "Towards model-based recognition of human movements in image sequences," *Computer Vision Graphics and Image Processing: Image Understanding*, vol. 59, no. 1, pp. 94–115, 1994.

[23] D. Gavrila and L. Davis, "3-D model-based tracking of humans in action: A multi-view approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (San Francisco, CA, USA), pp. 73–78, 1996.

[24] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential image using hidden Markov models," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, (Champaign, IL, USA), pp. 379–385, 1992.

[25] T. J. Darrell and A. P. Pentland, "Recognition of space-time gestures using a distributed representation," Tech. Rep. TR#197, MIT Media Lab Vision and Modeling Group, 1992.

[26] T. Darrell, I. Essa, and A. Pentland, "Task-specific gesture analysis in real-time using interpolated views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1236–1242, 1996.

[27] H. Murase and S. Nayar, "Learning and recognition of 3D objects from appearance," in *Proceedings of the IEEE Qualitative Vision Workshop*, (New York, NY, USA), pp. 39–49, 1993.

[28] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (San Juan, USA), pp. 193–199, 1997.

[29] A. Baumberg and D. Hogg, "An efficient method for contour tracking using active shape models," in *Proceedings of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, (Austin, TX, USA), pp. 194–199, 1994.

[30] J. Davis, "Apperance-based motion recognition of human activities," Tech. Rep. TR# 387, MIT Media Lab Perceptual Computing Group, 1996.

[31] R. Polana, "Temporal texture and activity recognition," Tech. Rep. TR#525, University of Rochester, Department of Computer Science, Oct. 1994.

[32] R. Polana and R. C. Nelson, "Low level recognition of human motion (or how to get your man without finding his body parts)," in *IEEE Computer Society Workshop on Motion of Nonrigid and Articulate Object*, (Austin, TX, USA), Oct. 1994.

[33] F. Liu and R. W. Picard, "Detecting and segmenting periodic motion," Tech. Rep. TR#400, MIT Media Lab Perceptual Computing Group, 1996.

[34] S. A. Niyogi and E. H. Adelson, "Analyzing and recognising walking figures in XYT," Tech. Rep. TR#223, MIT Media Lab Vision and Modeling Group, 1993.

[35] J. Little and J. Boyd, "Describing motion for recognition," in *International Symposium on Conputer Vision*, (Miami, FL, USA), pp. 245–240, 1995.

[36] M. Shah and R. Jain, *Motion-based Recognition.* Dordrecht, The Netherlands: Kluwer Academic Publishers, 1997.

[37] C. Myers, L. Rabiner, and A. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 6, pp. 623–635, 1980.

[38] R. Bellman, *Dynamic Programming.* Princeton, NJ, USA: Princeton University Press, 1957.

[39] K. Takahashi, S. Seki, H. Kojima, and R. Oka, "Recognition of dexterous manipulations from time-varying images," in *Proceedings of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, (Autin, TX, USA), pp. 23–28, 1994.

[40] A. Bobick and A. Wilson, "A state-based technique for the summarization and recognition of gestures," in *Proceedings of the International Conference on Computer Vision*, (Cambridge, MA, USA), pp. 328–388, 1995.

[41] X. Huang, Y. Ariki, and M. Jack, *Hidden Markov Models for Speech Recognition.* Edinburgh, UK: Edinburgh University Press, 1990.

[42] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–285, 1989.

[43] A. Pentland and A. Liu, "Modeling and prediction of human behavior," Tech. Rep. TR#433, MIT Media Lab Perceptual Computing Section, 1997.

[44] T. Starner and A. Pentland, "Visual recognition of american sign language using hidden Markov models," in *International Workshop on Automatic Face and Gesture Recognition*, (Zurich, Switzerland), pp. 189–194, 1995.

[45] A. Wilson and A. Bobick, "Realtime online adaptive gesture recognition," in *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, (Corfu, Greece), pp. 111–115, 1999.

[46] C. Vogler and D. Metaxas, "Asl recognition based on a coupling between HMMs and 3D motion analysis," in *International Conference on Computer Vision*, (Bombay, India), pp. 363–369, 1998.

[47] M. Yamamoto, A. Sato, and S. Kawada, "Incremental tracking of human actions from multiple views," in *Computer Society Conference on Computer Vision and Pattern Recognition*, (Santa Barbara, CA, USA), pp. 2–7, 1998.

[48] Y. Kameda, M. Minoh, and K. Ikeda, "Three dimensional pose estimation of an articulated object from its silhouette image," in *Proc. of the Asian Conference on Computer Vision*, (Osaka, Japan), pp. 612–615, 1993.

[49] M. Matlin, *Cognition.* Orlando FL, USA: Holt, Rinehart and Winston Inc., 1989.

[50] M. K. Leung and Y.-H. Yang, "First sight: A human body outline labeling system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 359–377, 1995.

[51] A. Shio and J. Sklansky, "Segmentation of people in motion," in *Proceedings of the IEEE Workshop on Visual Motion*, pp. 325–332, 1991.

[52] N. Pal and S. Pal, "A review of image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[53] W. Long and Y.-H. Yang, "Stationary background generation: An alternative to the difference of two images," *Pattern Recognition*, vol. 23, no. 12, pp. 1351–1359, 1990.

[54] B. Wohlberg and G. Cox, "Multiple person tracking using computer vision techniques," Tech. Rep. TN# 1998/11, DebTech, 1998.

[55] D. Ballard and C. Brown, *Computer Vision.* Englewood Cliffs: Prentice-Hall, 1982.

[56] C. Bishop, *Neural Networks for Pattern Recognition.* Oxford, UK: Oxford Press, 1997.

[57] T. Huang and A. Netravali, "Motion and structure from feature correspondences: A review," in *Proceedings of the IEEE*, vol. 82, pp. 251–268, Feb 1994.

[58] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[59] A. Poritz and A. Richter, “On hidden Markov models in isolated word recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (Tokyo, Japan), pp. 705–708, 1999.

[60] Y. Iwai, H. Shimizu, and M. Yachida, “Real-time context-based gesture recognition using HMM and automaton,” in *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, (Corfu, Greece), pp. 127–134, 1999.

[61] B. Juang and L. Rabiner, “The segmental k-means algorithm for estimating parameters of hidden Markov models,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 9, pp. 1639–1641, 1990.

[62] G. Forney, “The Viterbi algorithm,” *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–277, 1973.

[63] J. Rehg and T. Kanade, “Model-based tracking of self-occluding articulated objects,” in *Proceedings of the International Conference on Computer Vision*, (Cambridge, UK), pp. 612–617, 1995.

[64] P. Wolf, *Elements of Photogrammetry.* McGraw-Hill, 1974.

[65] S. Ganapathy, “Decomposition of transformation matrices for robot vision,” *Pattern Recognition Letters*, vol. 2, pp. 401–412, 1984.