955
2007
002

Human Movement Classification

Jelle Wiersma

stud. nr. 1211463

June 2007

Supervised by:

Dr. Bart de Boer
Drs. Gert Kootstra

Artificial Intelligence, University of Groningen
Email: [wiersma, bart, gert]@ai.rug.nl

Rijksuniversiteit Groningen
BIBLIOTHEEK
SOCIALE WETENSCHAPPEN

955

Table of Contents

1 Introduction 4

1.1 Introduction 5

1.2 Research Goals 7

2 Human Movement Classification 9

2.1 Human body structure analysis 9

2.1.1 Two-Dimensional non-model-based body structure analysis 9

2.1.2 Two-Dimensional model-based body structure analysis 11

2.1.3 Three-Dimensional model-based body structure analysis 12

2.1.4 Recapitulation 13

2.2 Tracking 14

2.2.1 Structural tracking 14

2.3 Activity Recognition and Action Classification 15

2.3.1 Template matching 16

2.3.2 State-space approaches 16

3 Face Detection 19

3.1 Background 19

3.2 Feature-based approaches 21

3.2.1 Low-level analysis 21

3.2.2 Feature analysis 21

3.2.3 Active shape models 22

3.3 Image-based approaches 22

3.3.1 Linear subspace methods 23

3.3.2 Neural Networks 24

3.4 A trainable feature-based approach 25

3.4.1 Integral Image 25

3.4.2 Haar-like features 26

3.4.3 AdaBoost 26

4 Tracking 29

4.1 MeanShift 29

4.1.1 Basic steps 29

4.1.2 Proof of convergence 30

4.2 CamShift 31

4.2.1 Basic steps 31

4.2.2 Computational complexity 33

4.3 Blob prediction 33

5 Classification 37
5.1 Movement statistics 37
5.2 Decision tree-based classification 39
5.3 Synthesis of the decision tree 40
6 The Complete Model 43
6.1 Programming environment 43
6.2 Design 43
6.2.1 Face detection 45
6.2.2 Histogram calculation 45
6.2.3 Histogram backprojection 46
6.2.4 Thresholding, dilation and erosion 47
6.2.4.1 Thresholding 47
6.2.4.2 Dilation 47
6.2.4.3 Erosion 48
6.2.5 Track window positioning 49
6.2.5 Decision tree-based classification 49
6.3 Discussion 50
6.3.1 Design decisions 50
6.3.2 Disadvantages 51
7 Experiments 53
7.1 Conditions 54
7.2 Results 54
8 Discussion & Conclusion 57

Appendix A: Group Results 59

Appendix B: All Results 65

Bibliography 73

Chapter 1

Introduction

1.1 Background

In a world where computers take a more and more essential place, the interaction between humans and computers is an important and interesting issue to be studied. How do we interact with the machines that influence our daily lives for a great part, and in what way could this interaction be improved? In the current situation, computers can be regarded as relatively blind and deaf. The information that they need, has to be spoon-fed to them by keyboard or mouse input. This is a very low-bandwidth and tedious form of communication, compared to the way in which humans interact. It also leads to a *reactive* mode of operation, where typed-in commands are simply executed. There is not much intelligence involved in the process, at least not on the machines behalf.

My aim however, and presumably that of the entire field of Artificial Intelligence, is to attempt to provide a machine with a little bit of our own intelligence. We are, after all, the most intelligent beings on this planet, so why not implement some of this intelligence on our computers? If computers were more aware of their environment, a more *pro-active* mode of operation could be established. And in making computers more aware of their environment, vision and speech are the modalities of greatest importance. If a computer can see that there is a person in front of it, it can act accordingly. Even better it would be, if the computer could recognize this person and thereby deciding whether to grant access to this person or not. Subsequently that person might give instructions to the computer just by using his or her voice. Though speech recognition goes beyond the scope of this thesis, it certainly is a discipline that could and surely will be combined with visual human-computer interaction.

The focus of my attention in this graduation project will be on the recognition and classification of human movements. The main reason for this choice is that I am fascinated by the idea that computers can be made smarter, eventually maybe smarter than humans. The most important of the five senses, I think is the sense of vision. So the idea of making computers smarter on the visual part came

as natural to me. I am also motivated by the fact that there is such a broad variety in promising applications for the movement recognition area.

For example, one can think of the domain of smart surveillance. Here "smart" describes a system that does more than motion detection, which is a straightforward task prone to false alarms (e.g. animals wandering around or objects moving due to strong wind). A first capability would be to sense if a human is indeed present. This might be followed by face recognition for the purpose of access control. In other applications, one needs to determine what a person in a scene is doing, rather than simply signaling human presence. In a parking lot setting, one might want to signal suspicious behavior such as wandering around and repeatedly bending over to cars. In a supermarket setting, valuable market information could be obtained by observing how consumers interact with the merchandise. This would be useful in designing a store layout which encourages sales. Other surveillance settings would include street cameras, monitoring a notorious street corner in front of bars where groups of people gather in the middle of the night. A smart camera that can automatically recognize aggressive behavior would account for a useful preprocessing step, reducing the work load for surveillance employees.

Another application domain is virtual reality. In order to create a presence in a virtual space, the body pose in the physical space needs to be recovered first. Application areas lie in interactive virtual worlds, usually with the internet as a medium. Other applications in this domain are games (e.g. making real music while playing air guitar), virtual studios and teleconferencing.

In the user interface application domain, vision is useful to complement speech recognition and natural language understanding for a natural and intelligent dialogue between man and machine. The contribution of vision to a speech-guided dialogue can be very useful. As mentioned before, a system can determine if a user is present to decide whether to initiate a dialogue or not. More detailed cues can be obtained by recognizing who the user is, observing facial expressions and gestures as the dialogue progresses, maybe remembering some of the past interactions, and in case of multiple participants, determining who is talking to who.

Another important application area in the user interface domain is in social interfaces. In this area, computer-generated characters with human-like behaviors are involved. The aim is to interact with humans in a more personal approach.

General Domain	Specific Area
Surveillance Systems	- access control - parking lots - street surveillance - traffic
Virtual Reality	- interactive virtual worlds - games - virtual studios - teleconferencing
Advanced User Interfaces	- gesture driven control - social interfaces - sign language translation - signaling in noisy environments
Motion Analysis	- content-based video indexing

Table 1.1: Overview of application areas of human movement recognition

Finally, there is the application area of motion analysis. With the rising popularity of digital libraries, the ability to automatically interpret video sequences will save tremendous effort in sorting and retrieving video files using content-based queries.

1.2 Research Goals

In the above section, various application areas of visual human-computer interaction have been discussed. In this section, I will describe the project that I have undertaken, with each of the subgoals that are the building blocks of this project. First I will state my research question, which is the following:

- In what way can the movements of human subjects best be observed, tracked and recognized automatically?

I have split this problem up into three major subproblems. The first one would be how to detect a human in front of a camera. There are many ways to find and model a human being in a camera frame. Chapter 2 will go into detail about this problem, and discuss previous contributions in this area. The choice that I have made for modeling the most important parts of the human being is beyond the scope of chapter 2. The choice that I have made is to detect a human face as a first step. Whenever a face can be detected in the camera frame, the assumption is

made that there is in fact a human standing in front of the camera. For an elaborate description of face detection techniques, I refer to chapter 3. In that chapter, I will also describe the face detection technique that I have used for the human movement classification task.
Tracking human movements in front of a camera is the second important step in the process. In chapter 4, a description is given about the tracking technique that I have chosen to use. It is called Continuously Adaptive Mean Shift (CamShift), an algorithm that has been proposed by Yizong Cheng (1995).
The final step in the human movement classification task is classification. In the previous two steps the subject has been detected, his or her movements have been tracked, and it is time to decide which of a prebuilt set of classes the movement can best be classified in. More details about this in chapter 5.
Chapter 6 will give a brief overview of the complete model, and will give a description of the separate parts. The design decisions will be discuss, and the (dis)advantages that the complete model has.
Chapter 7 contains the experimental setup and the results.

Chapter 2

Human Movement Classification

The purpose of Human Movement Classification (HMC) is to detect a human in a video sequence, track his or her movements and recognize the movement that this human makes. In the past a lot of research has been conducted on the subject. There are a number of ways to tackle the problem, and in this chapter the most successful methods that have been developed over the last couple of years will be discussed.

2.1 Human body structure analysis

This section will be about the analysis of the human body structure. To successfully track and recognize human movements, it is necessary to choose a good method of representation.

The body can be analyzed with or without using a pre-built model, and the representation can be in two or three dimensions. In section 2.1.1 two-dimensional body structure analysis without using a pre-built model will be discussed. Section 2.1.2 will be about two-dimensional approaches with explicit shape models. Finally, three-dimensional approaches will be discussed in section 2.1.3. See figure 2.1 for an overview of the different body structure analysis methods, and their corresponding researchers.

2.1.1 Two-Dimensional non-model-based body structure analysis

One general approach to the analysis of human movement has been to avoid a pose recovery step and to describe human movement in terms of simple low level, 2-dimensional features from a region of interest. Polana & Nelson (1994) call it "getting your man without finding his body parts". They describe models

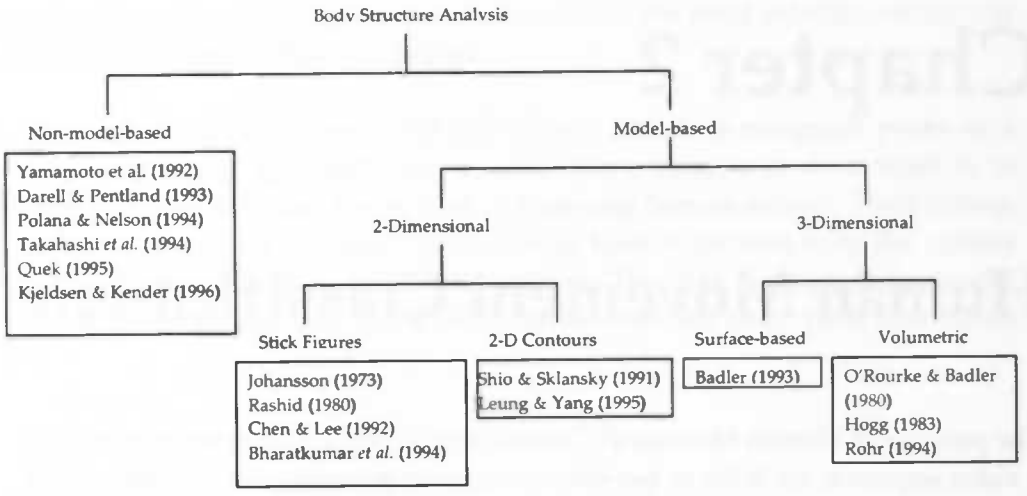

Figure 2.1: Overview of body structure analysis techniques

for human action in statistical terms derived from these low-level features, or by simple heuristics. The approach without explicit shape models has been especially popular for applications of hand pose estimation in sign language recognition and gesture-based dialogue management. For applications involving the human hand, the region of interest (ROI) is typically obtained by background image subtraction or skin color detection. This is followed by morphological operations to remove noise. The extracted features are based on hand shape, movement and location of the ROI.

Quek (1995) has proposed using shape and motion features alternatively for the interpretation of hand gestures. According to Quek, when the hand is in gross motion, the movements of the individual fingers are not important for gesture interpretation. On the other hand, gestures in which fingers move with respect to each other will be performed with little hand motion.

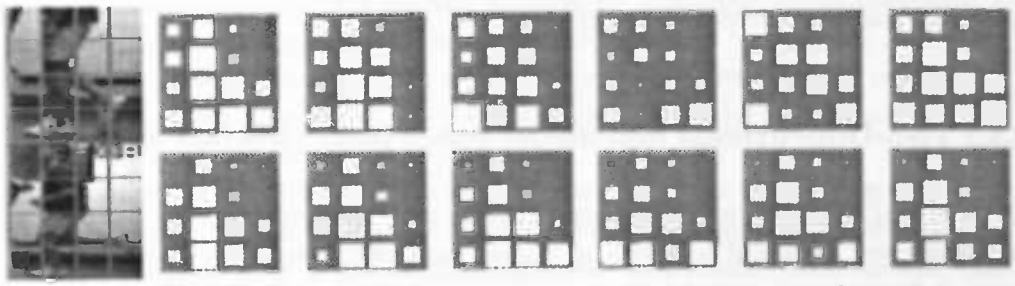

Figure 2.2: Detection of periodic activity using low level motion features (Polana & Nelson 1994)

A similar technique is to derive low level features is to superimpose a grid on the interest region, after a possible normalization of its extent. In each tile of the grid

a simple feature is computed, and these features are combined to form a $K \times K$ feature vector to describe the state of movement at time t. Polana & Nelson (1994) use the sum of the normal flow (see figure 2.2), Yamamoto et al. (1992) use the number of foreground pixels and Takahashi et al. (1994) define an average edge vector for each tile. Darell and Pentland (1993) and Kjeldsen and Kender (1996) use the image pixels directly as input.

2.1.2 Two-Dimensional model-based body structure analysis

This section discusses previous work which uses explicit *a priori* knowledge of how the human body appears in 2-D, taking essentially a model-based approach to segment and label body parts. Since self-occlusion makes the problem hard for movement recognition, many systems assume *a priori* knowledge of the type of movement or the viewpoint under which it is observed. The human shape is typically segmented by subtracting the background pixels, assuming a relatively stationary background and a fixed camera.

The models used are usually stick figures or 2-D contours. The advantage of using stick figures, is that it has some implicit assumptions that a leg or arm can only move within a certain degree of freedom. Therefore, some top-down information about the world is taken into the body structure analysis. This concept was initially considered by Johansson (1973), who marked joints as moving light displays (MLD's). Just by observing the movement of the MLD's, one could give a good indication of the activity that was being performed.
With this idea in mind, Rashid (1980) attempted to recover a connected human structure with projected MLD by assuming that points belonging to the same object have higher correlations in projected positions and velocities.

Chen and Lee (1992) recovered the 3-D configuration of a moving subject according to its projected 2-D image. Their model used 17 line segments and 14 joints to represent the features of the head, torso, hips, arms and legs. Various constraints were imposed for the basic analysis of the gait.
Bharatkumar *et al.* (1994) also used stick figures to model the lower limbs of the human body, where joints such as the hip, knee and ankle were considered. They aimed at a general model for gait analysis in human walking.

When a 2D contour model is used, it is necessary to distinguish on a rough scale between the different body parts (head, torso, arms and legs). These body parts can all be represented by a group of pixels (a *blob*). Shio and Sklansky (1991) focused their work on 2D translational motion of human blobs. The blobs were

grouped based on the magnitude and direction of the pixel velocity, which was obtained using optical flow techniques.

Leung and Yang (1995) applied a 2-D ribbon model to recognize poses of a human performing gymnastic movements. The emphasis of their work is to estimate motion just from the outline of a moving human subject. Their system consists of two major processes: extraction of human outlines from the camera image and interpretation of human motion. The 2-D contour body model outlines the structural and shape relations between the body parts (see figure 2.3).

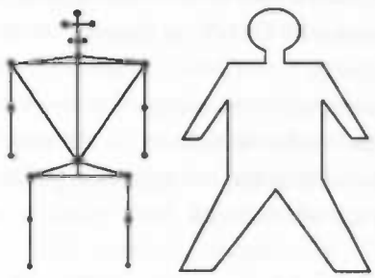

Figure 2.3: Human body representations: stick figure and 2-D contours

2.1.3 Three-Dimensional model-based body structure analysis

In this section the 3-D analysis of the human body will be discussed. When it comes to 3-D modeling of human forms, elliptical cylinders are one of the most commonly used volumetric models. Hogg (1983) and Rohr (1994) used a cylinder model in which the human body is represented by 14 elliptical cylinders. Each cylinder is described by three parameters: the length of the axis and the major and minor axes of the ellipse cross section. Both Hogg and Rohr attempted to generate 3-D descriptions of a human walking by modeling. Hogg presented a computer program called WALKER, which attempted to recover the 3-D body structure of a walking person. Rohr applied eigenvector line fitting to outline the human image, and then fitted the 2-D projections into the 3-D human model using a distance measure similar to Hogg.
O'Rourke and Badler (1980) conducted 3-D human motion analysis by mapping the input images to an elaborate volumetric model. That model is a well-defined structure, consisting of a set of line segments and joints. Their model also includes the constraints of human motion, such as restrictions on joint angles, and a method to detect collisions.

Many highly accurate surface models have been used in the field graphics to model the human body. Badler (1993) used a surface-based model containing thousands of polygons obtained from actual body scans. In vision, where the inverse problem of recovering the 3-D model from the images is much harder and less accurate, the use of volumetric primitives is preferred, because of the lower number of model parameters involved.

2.1.4 Recapitulation

All of the above approaches must match each real image frame to the corresponding model, which represents the human body structure as well as possible. This procedure is non-trivial. The complexity of the matching process depends on the number of parameters that the model uses and the efficiency of human body segmentation. When fewer parameters are used, it is easier to match the feature to the model, but more difficult to extract the feature. For example: the stick figure is the simplest way to represent a human body, and thus it is relatively easier to fit the extracted line segments to the corresponding body segments. However, extracting a stick figure from real images needs more care than searching for 2-D blobs or 3-D volumes.
All of the above discussed techniques have been studied to the purpose of the human movement classification model. The actual model contains a derivative of the two-dimensional non-model based method, which is not based on the modeling of the entire human body. Instead, it focuses on the three parts which carry the most information about the movement, namely the head and the hands.

2.2 Tracking

The second step in the process of human movement classification is tracking. Once a good representation has been found, whether it is model-based or non-model-based, the tracking of the various distinguished body parts is the next task to be performed. The objective of tracking is to establish correspondence of the image structure between consecutive frames based on features related to position, velocity, shape, texture and color. Typically, the tracking process involves matching between images using pixels, lines, and blobs, based on their motion, shape and other visual information (Aggarwal *et al.*, 1981). There are two general classes of correspondence models, namely *iconic models* and *structural models*. Iconic models use correlation templates, and are generally suitable for any objects, but only when the motion between two consecutive frames is small enough so that the object images in these frames are highly correlated. Because the sub-goal in this project is tracking human bodies, which retain a certain degree of non-rigidity, the focus will be on the use of structural models in the next section.

2.2.1 Structural tracking

Structural tracking, in contrast to iconic tracking, uses image features. Feature-based tracking typically starts with feature extraction, followed by feature matching over a sequence of images. The criteria for selecting a good feature are its robustness to noise, brightness, contrast, and size. To establish feature correspondence between successive frames, well-defined constraints are usually imposed to eliminate invalid matches and distinguish a unique correspondence. There is a trade-off between feature complexity and tracking efficiency. Low-level features, such as points, are easier to extract but relatively more difficult to track than higher-level features such as lines, blobs and polygons.

Cai et al. (1996) focused on tracking the movements of the whole human body using a viewing system with 2-D translational movement. They focused on dynamic recovery of still or changing background images. The image motion of the viewing camera was estimated by matching the line segments of the background image. After that, motion-compensated frames were constructed to adjust three consecutive frames into the same spatial reference. In the final stage, subjects were tracked using the center of the bounding boxes and estimated motion information.

Segen and Pingali's (1996) people tracking system utilized the corner points of moving contours as the features for correspondence. These feature points were matched in forward and backward orders between two successive frames using a distance measure related to position and curvature values of the points. The matching process implies that a certain degree of rigidity of the moving human body and small motion between consecutive frames was assumed. Finally, short-lived or partially overlapped trajectories were merged into long-lived paths.

Rossi and Bozzoli (1994) use moving blobs to track and count people crossing the field of view of a vertically mounted camera. Occlusion of multiple subjects was avoided due to the viewing angle of the camera, and tracking was performed using position estimation during the period when the subject enters the top and the bottom of the image.

Pentland *et al.* (1995) explored the blob feature thoroughly. In their work, blobs were not restricted to regions due to motion, and could be any homogeneous areas, such as color, texture, brightness, motion, shading or a combination of these. Statistics such as mean and covariance were used to model the blob features in both 2-D and 3-D. The feature vector of a blob is formulated as (x,y,Y,U,V), consisting of spatial (x,y) and color (Y,U,V) information. A human

body is constructed by blobs representing various body parts, such as head, torso, hands and feet. Meanwhile, the surrounding scene is modeled as a texture surface. Gaussian distributions were assumed for both models of human body and background scene. Finally, pixels belonging to the human body were assigned to different body part blobs using a log-likelihood measure.

2.3 Activity Recognition and Action Classification

In this section, past contributions to recognizing human activities from image sequences will be discussed. This is the third and final step in the Human Movement Classification task, after body structure analysis and tracking. Usually, human activity recognition is based on successfully tracking the human subject through image sequences, and therefore it is considered to be a higher level task. In the past, most efforts have been concentrated on two methods, namely *template matching* and *state-space approaches*. The basis of template matching is comparing the extracted features from the given image sequence to pre-stored patterns during the recognition process. The advantage of using this technique is its inexpensive computational cost. However, it is relatively sensitive to the variance of the movement duration. Approaches using state-space models, on the other hand, define each static posture as a state. These states are connected to each other by certain probabilities. Any motion sequence as a composition of these static poses is represented as a tour going through various states. Joint probabilities are computed through these tours, and the maximum value is selected as the criterion for classification of the activity. In such a scenario, duration of the movement is no longer an issue because each state can repeatedly visit itself.
However, approaches using these methods usually need intrinsic nonlinear models and do not have closed-form solutions. Nonlinear modeling requires searching for a global optimum in the training process, which requires complex computing iterations. Furthermore, a problem that this area still struggles with, is selecting the proper number of states and dimension of the feature vector, to avoid *underfitting* or *overfitting*. In the next sections, the work that has been done in both the *template matching* and *state-space approaches* areas will be discussed.

2.3.1 Template matching

Polana and Nelson (1994) detect periodic activity such as persons walking lateral to the viewing direction using spatio-temporal templates. They argue that a template matching technique is effective here because a sufficiently strong normalization can be carried out on the region of interest with respect to spatial and time scale variations. For the case of a stationary camera and a single object

of interest, background subtraction and size normalization of the foreground region is sufficient to obtain spatial invariance, if perspective effects are small. Polana and Nelson also describe a technique to deal with the more complex case of a moving camera and multiple overlapping objects, based on detecting and tracking independently moving objects. Size changes of the object are handled by estimating the spatial scale parameters and compensating for them, assuming the objects have a fixed height throughout the sequence. Temporal scale variations are factored out by detecting the frequency of an activity. After these normalizations, spatio-temporal templates are constructed to denote one generic cycle of activity. A cycle is divided into a fixed number of subintervals for which motion features are computed. The features of a generic cycle are obtained by averaging corresponding motion features over multiple cycles.

2.3.2 State-space approaches

State-space models have been widely used to predict, estimate, and detect signals over a large variety of applications. One representative model is the Hidden Markov Model (HMM), which is a probabilistic technique for the study of discrete time series. HMM has been very popular in speech recognition, but only recently has it been adopted for recognition of human motion sequences in computer vision (Yamamoto *et al.*, 1992). Its model structure could be summarized as a hidden Markov chain and a finite set of output probability distributions (Poritz, 1988). The basic structure of an HMM is shown in figure 2.4.

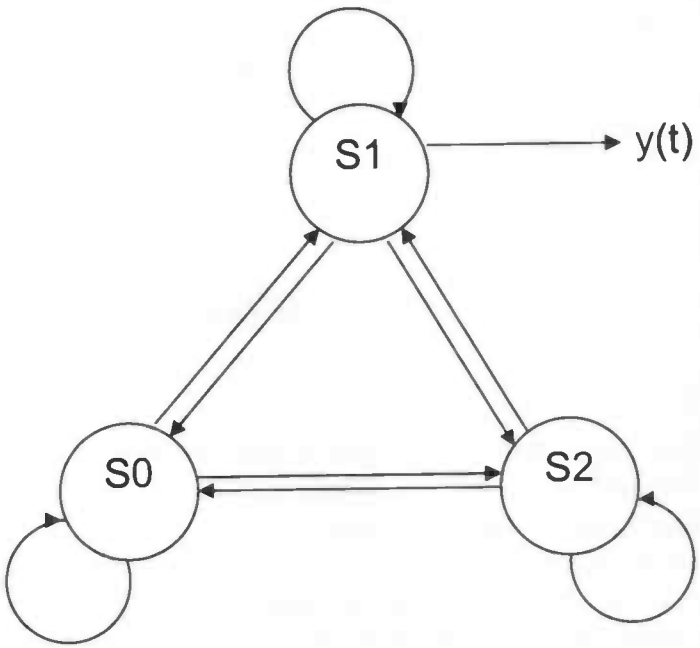

Figure 2.4: The basic structure of a Hidden Markov Model

Every state is represented by a circle S_i, which is connected to other states by probabilities. The parameter $y(t)$ represents the observation derived from each state. The main tool in HMM is the Baum-Welch (forward-backward) algorithm for maximum likelihood estimation of the model parameters. Features to be recognized in each state vary from points and lines to 2-D blobs.

Goddard's (1994) human movement recognition focused on the low limb segments of the human stick figure. 2-Dimensional projections of the joints were used as inputs, and features for recognition were encoded by coarse orientation and coarse angular speed of the line segments in the image plane. Although Goddard did not directly apply Hidden Markov Models in his work, he did consider a movement as a composition of events linked by time intervals.

The above discussed methods have been studied for the purpose of the human movement classification model. As a consequence of a paradigm shift, none of these methods have been used in the implementation, although they have been of great value in the process of developing the program. Chapter 5 and 6 will go into detail about the chosen classification method.

Chapter 3

Face Detection

Computer vision in general aims to duplicate human vision. Detecting faces is one of the visual tasks which humans can do effortlessly. However, in computer vision terms, this task is far from easy. Computers cannot make use of the amount of top-down knowledge that is required for a high-level task like face detection. Humans, however, seem to have no trouble at all to recognize a human face, even if the face is only partly shown to the observer. Computer vision techniques have improved greatly over the last decades, but still there are a lot of problems to be solved. In this chapter, an overview will be given of the approaches with respect to the face detection task that have been used in the past. After that the approach that has been chosen to use for this implementation will be discussed.

3.1 Background

Face detection is a computer vision technology that determines the locations and sizes of human faces in digital images. The face detection task can be regarded as a special case of object-class detection, with the purpose of classifying any face candidates into one of two possible classes: namely face or non-face. This task should not be confused with face recognition, a more complex task which aims to not only detect a human face in an image, but also identify that person.
In the past, a lot of research has been done on the problem of recognizing a human face in a picture or video sequence. Various techniques have been developed over the years, ranging from simple edge-based algorithms (Sakai *et al.*, 1972; Craw *et al.*, 1987) to composite high-level approaches utilizing advanced pattern recognition methods (Sirovich & Kirby, 1987; Turk & Pentland, 1991; Rowley *et al.*, 1998).

Because face detection techniques require *a priori* information of the face, they can be effectively organized into two broad categories distinguished by their different approach to utilizing face knowledge. The techniques in the first category make explicit use of face knowledge and follow the classic detection

methodology in which low level features are derived prior to knowledge-based analysis. The apparent properties of the face, such as skin color and face geometry are exploited at different system levels. Typically, in these techniques face detection tasks are accomplished by manipulating distance, angles, and area measurements of the visual features derived from the scene. Since features are the main ingredients, these techniques are called the *feature-based approach*.

On the other hand, the *image-based approach* uses training algorithms that focus on the image itself, without using feature derivation and analysis. This approach is more robust with respect to the unpredictability of face appearance and environmental conditions. By formulating the problem as a pattern recognition problem, learning the face pattern from examples, the specific application of face knowledge is avoided. This eliminates the potential of modeling error due to incomplete or inaccurate face knowledge.

Most of the image-based approaches apply a window scanning technique for detecting faces. The window scanning technique can be seen as an exhaustive search for possible face locations at all scales. The image-based approaches can roughly be divided in three groups: linear subspace methods, neural networks and statistical approaches. In section 3.1.2, these three methods will be discussed. In section 3.1.1 feature-based approaches will be discussed.

In figure 3.1 the most recent and most successful face detection techniques are listed by group.

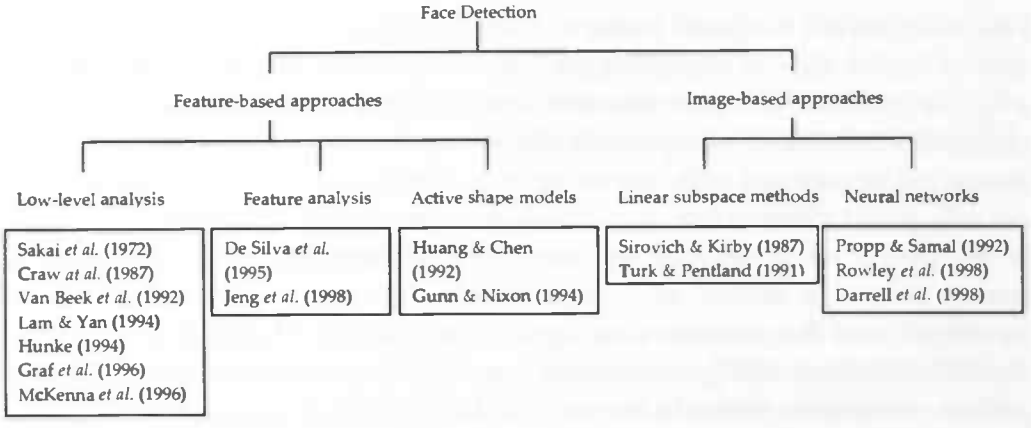

Figure 3.1: Overview of face detection techniques

3.2 Feature-based approaches

Feature-based approaches to face recognition can be further divided into *low-level analysis*, *feature analysis* and *active shape models*.

3.2.1 Low-level analysis

The two most primitive ways to analyze a human face are based on edges, or color information (grayscale or RGB).
Edge representation was applied in the earliest face detection work by Sakai *et al.* (1972). The work was based on analyzing line drawings of the faces from photographs, aiming to locate facial features. Craw *et al.* (1987) later proposed a hierarchical framework based on Sakai *et al.*'s work to trace a human head outline. The work includes a line-follower implemented with curvature constraint to prevent it from being distracted by noisy edges. Edge features within the head outline are then subjected to feature analysis using shape and position information of the face.

Besides edge details, the gray information within a face can also be used as features. Facial features such as eyebrows, pupils, and lips appear generally darker than their surrounding facial regions. This property can be exploited to discriminate between various facial parts. Several facial feature extraction algorithms (Van Beek *et al.* (1992), Graf *et al.* (1996) and Lam & Yan (1994)) search for local gray minima within segmented facial regions. In these algorithms, the input images are first enhanced by contrast-stretching and gray-scale morphological routines to improve the quality of local dark patches and thereby make detection easier.

While gray information provides the basic representation for image features, color is a more powerful means of discerning object appearance. Due to the extra dimensions that color has, two shapes of similar gray information might appear very differently in color space. It was found that different human skin color gives rise to a tight cluster in color spaces even when faces of different races are considered (Hunke (1994), McKenna *et al* (1996)., Yang & Waibel (1996)). This means color composition of human skin differs little across individuals.

3.2.2 Feature analysis

In feature analysis, visual features are organized into a more global concept of face and facial features using information of face geometry. Through feature analysis, feature ambiguities are reduced and locations of the face and facial

features are determined. Feature searching techniques begin with the determination of prominent facial features. The detection of the prominent features then allows for the existence of other less prominent features to be hypothesized using anthropomorphic measurements of face geometry (De Silva *et al.*, 1995). For instance, a small area on top of a larger area in a head and shoulder sequence implies a "face on top of shoulder" scenario, and a pair of dark regions found in the face area increases the confidence of a face existence.
Jeng *et al.* (1998) propose a system for face and facial feature detection which is also based on anthropomorphic measures. In their system, they initially try to establish possible locations of the eyes in binarized pre-processed images. For each possible eye pair the algorithm searches for a nose, a mouth, and eyebrows. Each facial feature has an associated evaluation function, which is used to determine the final most likely face candidate.

3.2.3 Active shape models

Unlike the face models described in the previous two sections, active shape models depict the actual physical and therefore higher-level appearance of features. Once released within a close proximity to a feature, an active shape model will interact with local image features (like edges and brightness) and gradually deform to take the shape of the feature. An example of active shape models is snakes. Snakes are commonly used to locate a head boundary (Huang & Chen, 1992; Gunn & Nixon, 1994; Lam & Yan, 1996). In order to achieve the task, a snake is first initialized at the proximity around a head boundary. It then locks onto nearby edges and subsequently assumes the shape of the head. The evolution of a snake is achieved by minimizing an energy function. Due to a high computational requirement in this minimization process, fast iteration methods, based on greedy search algorithms, have been employed (Huang & Chen, 1992; Lam & Yan, 1996).

3.3 Image-based approaches

Unlike feature-based approaches, image-based approaches use training algorithms that focus on the image itself, without using feature derivation and analysis. This approach is more robust with respect to the unpredictability of face appearance and environmental conditions. Again, these approaches can be further divided into two subclasses: *linear subspace methods* and *neural networks*.

3.3.1 Linear subspace methods

Images of human faces lie in a subspace of the overall image space. One method to represent these subspaces is Principal Component Analysis (PCA). Sirovich and Kirby (1987) developed a technique using PCA to efficiently represent human faces. Given a set of different face images, the technique first finds the principal components of the distribution of faces, expressed in terms of *eigenvectors*. Each individual face in the face set can then be approximated by a linear combination of the largest *eigenvectors*, more commonly referred to as *eigenfaces*, using appropriate weights.

Turk & Pentland (1991) later developed this technique for face recognition. Their method exploits the distinct nature of the weights of eigenfaces in individual face representation. Since the face reconstruction by its principal components is an approximation, a residual error is defined in the algorithm as a preliminary measure of how well the approximation fits the face. This residual error which they called "distance from face-space" gives a good indication of face existence through the observation of global minima in the distance map.

3.3.2 Neural networks

Neural networks have become a popular technique for pattern recognition problems, including face detection. Neural networks today are much more than just the simple multi-layer perceptron (MLP). Modular architectures, committee-ensemble classification, complex learning algorithms, autoassociative and compression networks, and networks evolved or pruned with genetic algorithms are all examples of the widespread use of neural networks in pattern recognition.

The first neural approaches to face detection were based on MLPs (Propp & Samal, 1992). The first advanced neural approach which reported promising results on a large, difficult dataset was by Rowley *et al.* (1998). Their system incorporates face knowledge in a retinally connected neural network, as shown in figure 3.2.

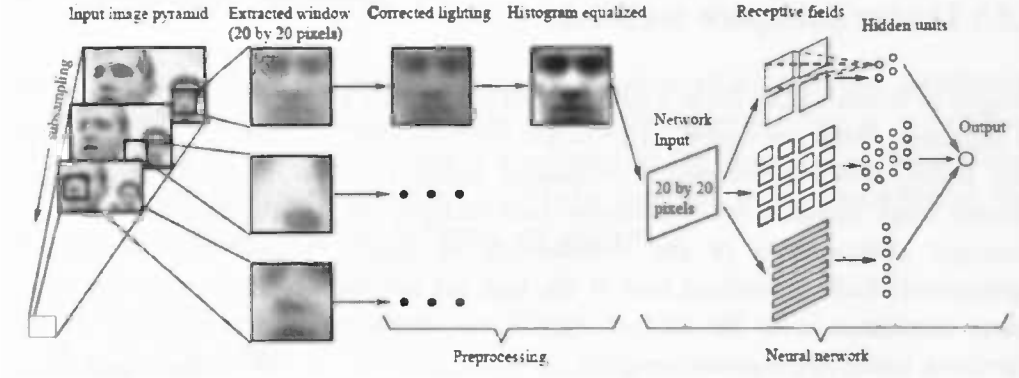

Figure 3.2: Face detection neural network by Rowley et al. (1998)

The network is designed to look at windows of 20 x 20 pixels. There is one hidden layer with 26 units, where 4 units look at 10 x 10 pixel subregions, 16 units look at 5 x 5 pixel subregions, and 6 units look at 20 x 5 pixel overlapping horizontal stripes. The input is preprocessed through lighting correction and histogram equalization. The network has an 86.2% recognition rate with 23 false positives on a testset containing 130 images. To further improve performance, Rowley *et al.* trained multiple networks and combined the output by a voting system. Their algorithm has been applied in a person tracking system by Darrell *et al.* (1998).

3.4 A trainable feature-based approach

In this section, a feature-based approach will be discussed, introduced by Viola and Jones (2001). This is the approach that will be used for the human movement recognition program. Viola and Jones propose an algorithm which focuses on features rather than on pixels directly. The main reason is that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. The second reason is that a feature-based system operates much faster than a pixel-based system.

The algorithm uses Haar-like features (see section 3.2.2). Using these features, a machine learning algorithm can be trained to discriminate between faces and non-faces. Freund and Schapire (1995) have developed a method called AdaBoost (see section 3.2.3) which selects a small number of important features to form extremely efficient classifiers. A new image representation called the *integral image* has been proposed by Viola and Jones (2001) which allows for very fast feature computation (see section 3.2.1).

3.4.1 Integral Image

The integral image has been introduced by Viola and Jones (2001). This image representation is a pre-processing step to the computation of Haar-like features, which will become clear in the next sections. The integral image contains the sum of the pixels above and to the left of *x,y*:

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x', y')$$

where $ii(x,y)$ is the integral image and $i(x,y)$ is the original image. Using the following pair of recurrences:

$$s(x,y) = s(x,y-1) + i(x,y)$$
$$ii(x,y) = ii(x-1,y) + s(x,y)$$

where $s(x,y)$ is the cumulative row sum, $s(x,-1) = 0$, and $ii(-1,y) = 0$, the integral image can be computed in one pass over the original image. This constant time complexity is a powerful feature of the integral image representation. In figure 3.3 an example of an integral image computation is given. The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is equal to A + B, at location 3 it is A + C, and at location 4 it is A + B + C + D. Thus the sum within rectangle D can be computed as 4 + 1 – (2 + 3).

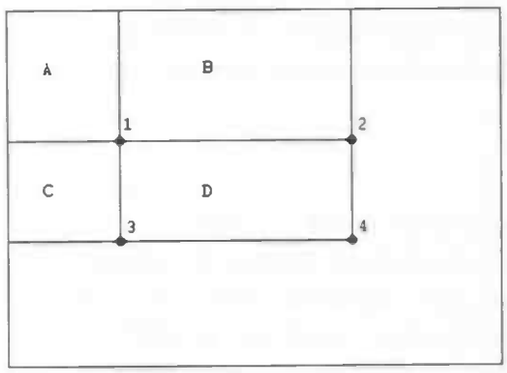

Figure 3.3: Integral image computation

3.4.2 Haar-like features

Isolated pixel values do not give any information other than the luminance and/or the color of the radiation received by the camera. Thus, a recognition process can be much more efficient if it is based on the detection of features that encode some information about the class to be detected. In the figure below, the 14 features used, divided into 3 classes, are shown:

1. Edge features:

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

2. Line features:

(a) (b) (c) (d) (e) (f) (g) (h)

3. Center-surround features:

(a) (b)

Haar-like features are the key features of the face recognition algorithm. Instead of focusing on raw pixel values, a set of Haar-like features is computed from an image. Any of these Haar-like features can be computed at any scale or location in constant time using the integral image representation for images.

3.4.3 AdaBoost

Given a feature set and a training set of positive and negative images, a machine learning approach called AdaBoost (Freund & Schapire, 1996) is used to learn a classification function. The algorithm both selects a small set of Haar-like features and trains a classifier. AdaBoost is used to improve (i.e. boost) the classification performance of a simple (weak) learning algorithm. The primary weak learning algorithm selects the single feature which best separates the positive and negative training examples. For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples are misclassified. A weak classifier $h_j(x)$ consists of a feature f_j, a threshold θ_j and a parity p_j indicating the direction of the parity sign:

$$h_j(x) = \begin{cases} 1 & if\ p_j f_j(x) < p_j \theta_j \\ 0 & otherwise \end{cases}$$

Here x is a 24x24 pixel sub-window of an image. Below, the boosting process is shown in pseudocode.

- Given example images $(x_1, y_1), \mathrm{K}\ ,(x_n, y_n)$ where $y_i = 0,1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0,1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \mathrm{K}\ , T$:
    1. Normalize the weights,

       $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$ so that w_t is a probability distribution.
    2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\varepsilon_j = \sum_i w_i \,|\, h_j(x_i) - y_i \,|$.
    3. Choose the classifier, h_t, with the lowest error ε_t.
    4. Update the weights:

       $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$

       where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$.

- The final strong classifier is:

$$h_j(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$.

No single feature can perform the classification task with low error. Features which are selected in early rounds of the boosting process have error rates

between 0.1 and 0.3. Features selected in later rounds, as the task becomes more difficult, have error rates between 0.4 and 0.5.
The first features selected by AdaBoost are meaningful and easily interpreted. The first feature seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose. See figure 3.3.

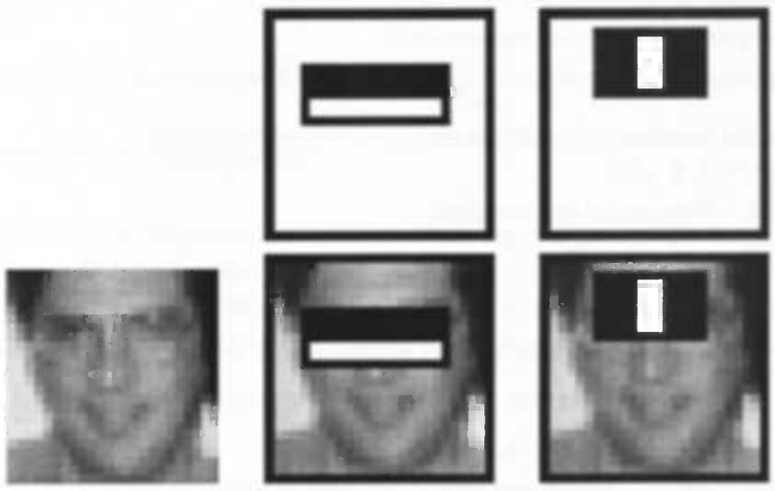

Figure 3.3: The first and second features selected by AdaBoost

The outcome of the AdaBoost learning process is a cascade of classifiers. The initial classifier eliminates a large number of negative examples with very little processing. Subsequent layers eliminate additional negatives but require more computation. After several stages of processing the number of subwindows has been reduced radically. See figure 3.4.

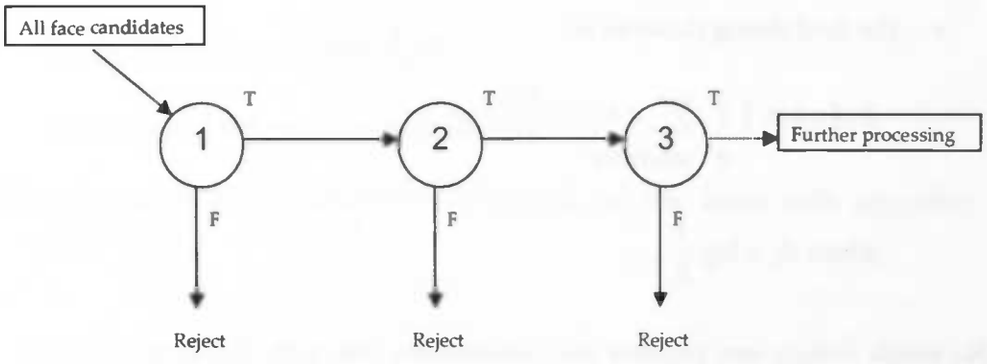

Figure 3.4: Schematic depiction of the detection cascade

Chapter 4

Tracking

The second step in the human movement recognition task is tracking. In this project, the most important human body parts to be tracked are the face and hands. The blobs that represent these parts carry the most information about the movement that is made. Furthermore, they are also the easiest to track, because of their distinctive color and shape. In this chapter, the theory of an algorithm that has been chosen to perform the tracking task with, called CamShift (Continuously Adaptive Mean Shift) will be discussed. It is an expansion of the Mean Shift algorithm, which will be described in section 4.1. Furthermore, two movement prediction techniques will be discussed in section 4.3.

4.1 Mean Shift

The Mean Shift algorithm has been presented by Yizong Cheng (1995). It is a robust, non-parametric technique that climbs the gradient of a probability distribution to find the mode (peak) of that distribution. Tracking is a task that can be generally regarded as nothing more than following a peak of a distribution of pixels. The basic steps of this technique will be discussed in section 4.1.1. Because it is a gradient ascent function, the convergence of the algorithm needs to be proved. Section 4.1.2 will go into detail about the proof of convergence.

4.1.1 Basic steps

The basic steps of the Mean Shift algorithm are the following:

1. Choose an initial search window size
2. Choose the initial location of the search window
3. Compute the mean location in the search window
4. Center the search window at the mean location computed in step 3
5. Repeat 3 and 4 until convergence i.e. until the mean location moves less than a preset threshold

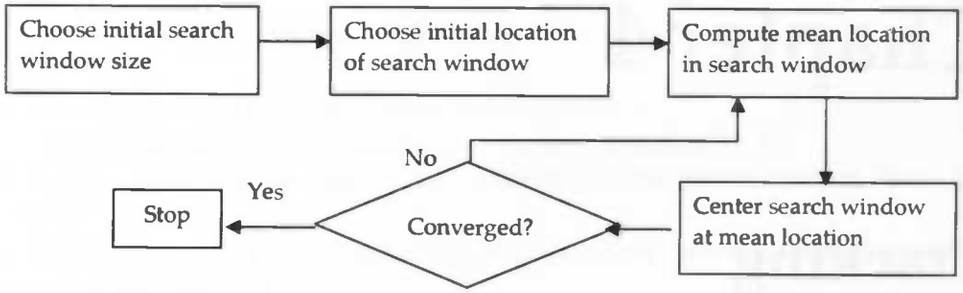

Figure 4.1: The Mean Shift algorithm

4.1.2 Proof of convergence

The proof of the convergence of the Mean Shift algorithm is taken from Cheng's article (Cheng, 1995).
Assuming a Euclidian distribution space containing distribution *f*, the proof is as follows reflecting the steps discussed in the above section:

1. A window *W* is chosen at size *s*
2. The initial search window is centered at data point p_k
3. Compute the mean position within the search window:

$$\hat{\bar{p}}_k(W) = \frac{1}{|W|}\sum_{j \in W} p_j$$

The mean shift climbs the gradient of *f(p)*

$$\hat{\bar{p}}_k(W) - p_k \approx \frac{f'(p_k)}{f(p_k)}$$

4. Center the window at point $\hat{\bar{p}}_k(W)$

5. Repeat steps 3 & 4 until convergence

Near the mode of the distribution, $f'(p) \cong 0$, so the mean shift algorithm converges there.

4.2 CamShift

The primary difference between CamShift and Mean Shift is that CamShift uses continuously adaptive probability functions. That means that distributions can be recomputed for every frame. The difference is that Mean Shift is based on static distributions. The distributions are not updated unless the target experiences significant changes in shape, size or color. Since CamShift does not maintain static distributions, spatial moments are used to iterate towards the mode of the distribution. This is in contrast to the implementation of the Mean Shift algorithm where target and candidate distributions are used to iterate towards the maximum increase in density using the ratio of the current distribution over the target.

CamShift is primarily intended to perform efficient head and face tracking in a perceptual user interface. Bradski (1998) used CamShift to build a user-friendly interface for controlling commercial computer games and for exploring 3-D graphic worlds. He implemented a face tracker that makes it possible to play Quake 2 hands free and to fly over a 3-D generated landscape, controlling the directions by head movements.

4.2.1 Basic steps

A preliminary step in the CamShift algorithm computation is the calculation of moments. Moments have to be calculated in order to find the center of mass i.e.: the mean location of the search window. The zeroth and first moments for x and y are calculated as follows:

Zeroth moment: $$M_{00} = \sum_x \sum_y I(x,y)$$

First moment for x: $$M_{10} = \sum_x \sum_y xI(x,y)$$

First moment for y: $$M_{01} = \sum_x \sum_y yI(x,y)$$

Then the mean search window location is calculated as follows:

$$x_c = \frac{M_{10}}{M_{00}}; y_c = \frac{M_{01}}{M_{00}};$$

where I(x,y) is the pixel probability value at position (x,y) in the image, and x and y range over the search window.

The basic steps of the CamShift algorithm are:

1. Choose the initial location of the search window
2. Execute the Mean Shift algorithm as described in section 4.1.1
3. Set the search window size equal to a function of the zeroth moment found in step 2
4. Repeat steps 2 and 3 until convergence i.e. until the mean location moves less than a preset threshold

In figure 4.2, CamShift is shown beginning the search process at the top left (step 1) step by step until convergence is reached (step 6). In this figure, the red graph is a 1-dimensional cross-section of an actual flesh color probability distribution of an image of a face (the biggest red part, from x = 8 to x = 22) and a nearby hand (the smaller red part, from x = 1 to x = 7). The CamShift search window is depicted as a yellow square. The purple triangle is representing the mean shift point, i.e. the center of mass of the search window. On the horizontal axis, the x-position within the image is given. The distribution value is given on the vertical axis. The window is initialized at size three and converges to cover the tracked face in 6 steps. The nearby hand is ignored as long as it doesn't overlap the face.

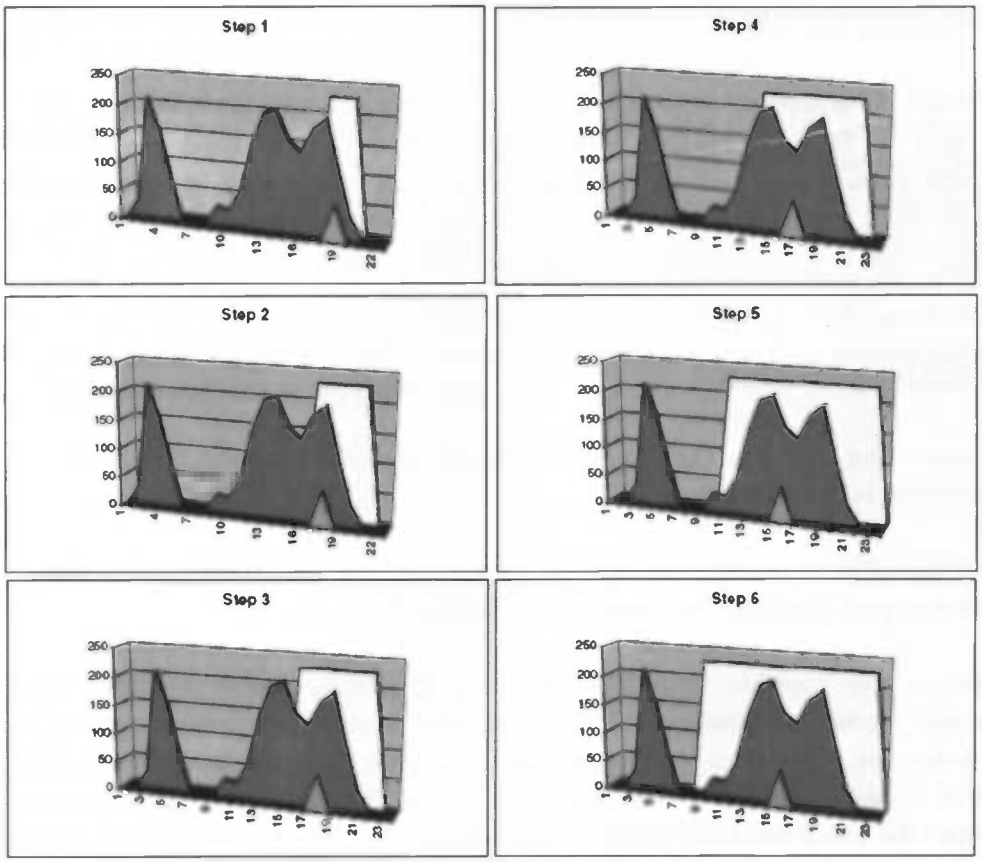

Figure 4.2: Example of CamShift tracking a human face

In figure 4.3, the next camera frame is shown as the head has moved slightly to the left. The initial search window location is equal to the last position in the previous step (figure 4.2, step 6). The CamShift algorithm now converges in just 2 steps.

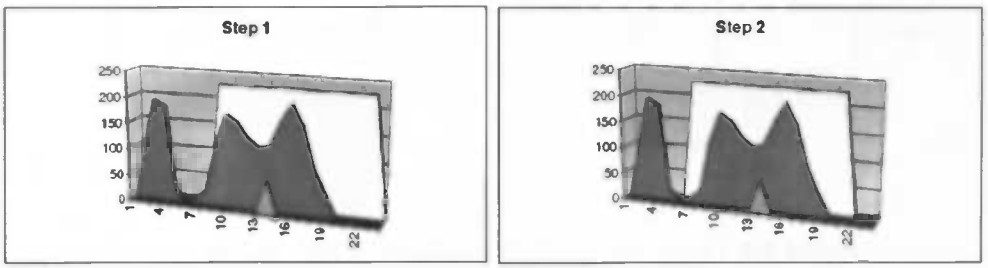

Figure 4.3: The next step in the CamShift tracking procedure

4.2.2 Computational complexity

The order of complexity of CamShift is $O(N^2)$, where the image is taken to be of size N x N. Furthermore, the computational time is most influenced by the moment calculations and the average number of mean shift iterations until convergence. The biggest computational savings come through scaling the region of calculation to an area around the search window size.

4.3 Blob prediction

An interesting problem that emerges from the way the tracking task has been represented is the following:

- Is it possible to predict the location of a (hand or head) blob, given that a number of past positions and velocities are known?

There are two possible ways to solve the problem: by linear prediction and parabolic prediction. Linear prediction takes into account two previous positions and velocities. It assumes a constant velocity in both the x and y directions. This method is best suited for relatively small blob movements, because error margin between the predicted blob position and velocity and the real blob position and velocity is trivial when the positional difference is small.

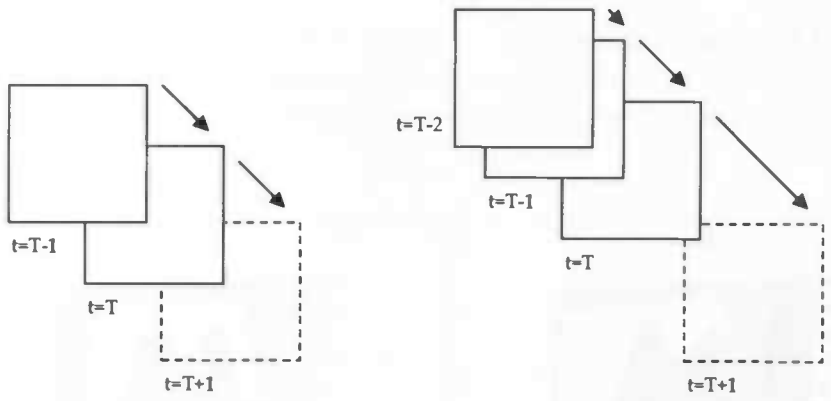

Figure 4.4: Linear and parabolic prediction of blob motion

On the other hand, when the blob velocity is higher, or the difference between the consecutive blob positions is greater, parabolic prediction should be applied. Parabolic prediction uses three (or more) previous data points. Velocity can be variable in this case, though acceleration is assumed to be constant in the case of

3 data points. Parabolic yields the best results if the variation in velocity is great or if there is a sudden change of direction.

The need for a blob prediction technique stems from the inherent inclination of CamShift to lose track of one or more blobs under some circumstances. These circumstances can be occlusion of the object by another object or a movement that is too quick, so that the difference between two consecutive blob locations is too big for CamShift to cover. By implementing a good prediction algorithm, these problems can be solved. Although experiments with both prediction techniques have been conducted, the final movement classification program does not contain an option of blob prediction.

Chapter 5

Classification

In the previous chapter the tracking of a human subject has been discussed. During this process a number of statistics have been monitored throughout the movement. On the basis of these statistics, the classification algorithm is making a classification into one of the prebuilt set of classes.

5.1 Movement Statistics

In order to perform a classification of a certain data point (an observation of a movement), some information about that data has to be known. Information that can distinguish the data point from other points. The information that is known at time t, is the position of track windows 1, 2 and 3 at time t and the positions of these three windows at all past times up until time t.
Therefore the statistics that are kept about the track windows are the following:

- x and y position of the track windows at time t (denoted as x_0, y_0, x_1, y_1, x_2 and y_2);
- The extreme values of these positions (minimum and maximum) up until time t and the difference between those, yielding the variation in x and y direction;
- The difference between x and y positions of different blobs;
- The mean and mode over y positions of the head blob up until time t;
- The center of mass of the movement, which is the running average of all positions up until time t;
- The angle of the position of the window with respect to the center of mass;
- A statistic monitoring the direction of a circular movement, based on a decrease or increase of the angle for a period of time greater than a preset threshold.

In figure 5.1, four movements (M1, M2, M5 and M11) are shown, with three snap shots of every movement plus the graph of their most important variable. In figure 5.1a and 5.1b this is the y-position of the head blob. Based on the difference in mean(y) and mode(y) a distinction is made. For jumping, the mean of the y-positions of the head blob is always greater than the mode of the y-positions.

Figure 5.1a: Movement 1, jumping

Figure 5.1b: Movement 2, knee bending

Figure 5.1c shows both y-positions of the hand blobs, the blue line representing the left hand and the black line representing the right. The variation in y-direction is greater than the variation in x-direction for both hands. This is a statistic that is taken into the classification process.

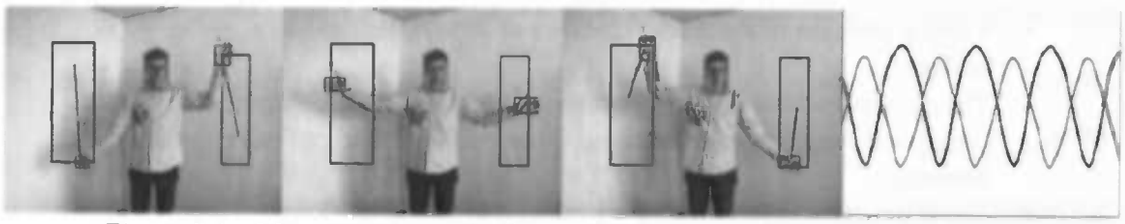

Figure 5.1c: Movement 5, vertically waving both arms in counterphase

In the first three graphs, the time variable is on the horizontal axis. In figure 5.1d, t is on the vertical axis, with the x-positions of the hand blobs showing on the horizontal axis. In this case, the difference between the two hand blobs is also taken into account, as well as the x- and y-variation of the hand blobs. The difference in x-position remains constant during the movement, which tells us that the hand movements are in phase.

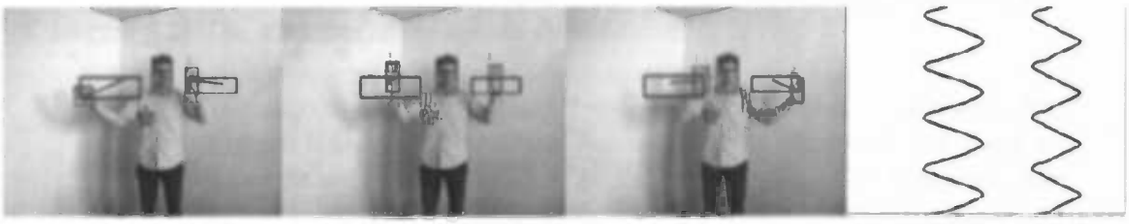

Figure 5.1d: Movement 11, waving both arms in phase

5.2 Decision Tree-based Classification

The basis of the classification algorithm is a decision tree. In figure 5.2 the decision tree that is used for the human movement classification program is shown. In table 5.1, the fifteen movements and their referral code are shown.

M1	Jumping
M2	Knee bending
M3	Clapping
M4	Vertically Waving Both Hands in Phase
M5	Vertically Waving Both Hands in Counterphase
M6	Clockwise Circular Movement with One Hand
M7	Counterclockwise Circular Movement with One Hand
M8	Waving Both Hands in Counterphase
M9	Drumming in Counterphase
M10	Waving Right Hand
M11	Waving Both Hands in Phase
M12	Drumming in Phase
M13	Vertically Waving Left Hand
M14	Vertically Waving Right Hand
M15	Waving Left Hand

Table 5.1

5.3 Synthesis of the Decision Tree

In figure 5.2, the decision tree is given.

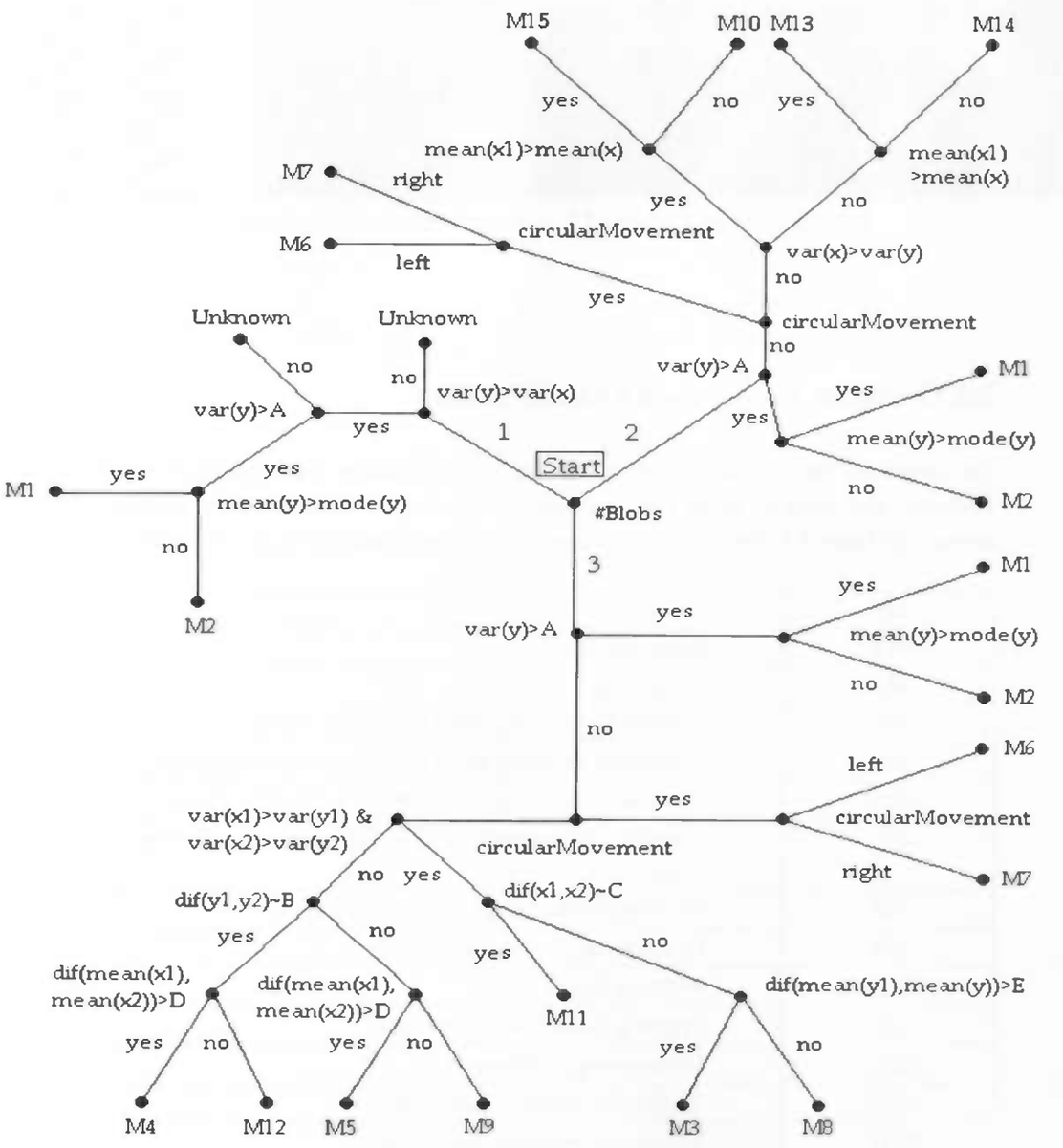

Figure 5.2

All distincting factors are taken into account in the tree, in order to decide what movement class the observation is classified into.
The tree has been created in a way that best separates the distincting features of the movements. A number of constants has to be set in order to achieve the best classification. These constants are represented by the letters A,B,C,D and E (see figure 5.2). The meaning of the constants are:

- A: The maximum amount of head movement that is allowed for movements M3-M15. These are the movements in which the head is supposed to stay relatively still. In movements 1 and 2, however, the head movement is significantly bigger.
- B: The difference in y-position between the hand blobs. This statistic is necessary to differ between in phase and in counterphase movements.
- C: Same as B, but for the x-position between hand blobs.
- D: A threshold for the maximum absolute horizontal distance between both hand blobs.
- E: A threshold for the maximum absolute vertical distance between both hand blobs.

The values of these constants have been entered after a period of testing. The performance of the decision tree is determined by these values, therefore it is important to have these parameters set as good as possible.
In the current form, the decision tree can only classify movements into one of the pre-set fifteen classes. There is no possibility to learn new movements. New movements can, however, be hard-coded into the tree. In order to achieve this, a branch has to be added to an existing branch. The movement must be well defined a priori in order to achieve a good recognition rate.

Chapter 6

The Complete Model

In this chapter, an overview of the implementation of the complete movement classification model will be given.

6.1 Programming environment

For the implementation, the programming language C has been chosen. The advantage of C is that it is a powerful, imperative programming language. Another reason for this programming language is that Intel's open source computer vision library OpenCV is based on C functions. OpenCV offers a great amount of useful visual processing functions. It is optimized and intended for real-time applications, which is exactly what is useful for an application like human movement classification.

6.2 Design

In this section, the main components of the movement classification program will be discussed. The program can be divided into the following components:

- Face detection
- Histogram calculation
- Histogram backprojection
- Thresholding, dilation & erosion
- Tracking windows using *CamShift*
- Decision tree-based classification

In figure 6.1, an overview of the movement classification program is given. In the next sections every component of the program will be discussed.

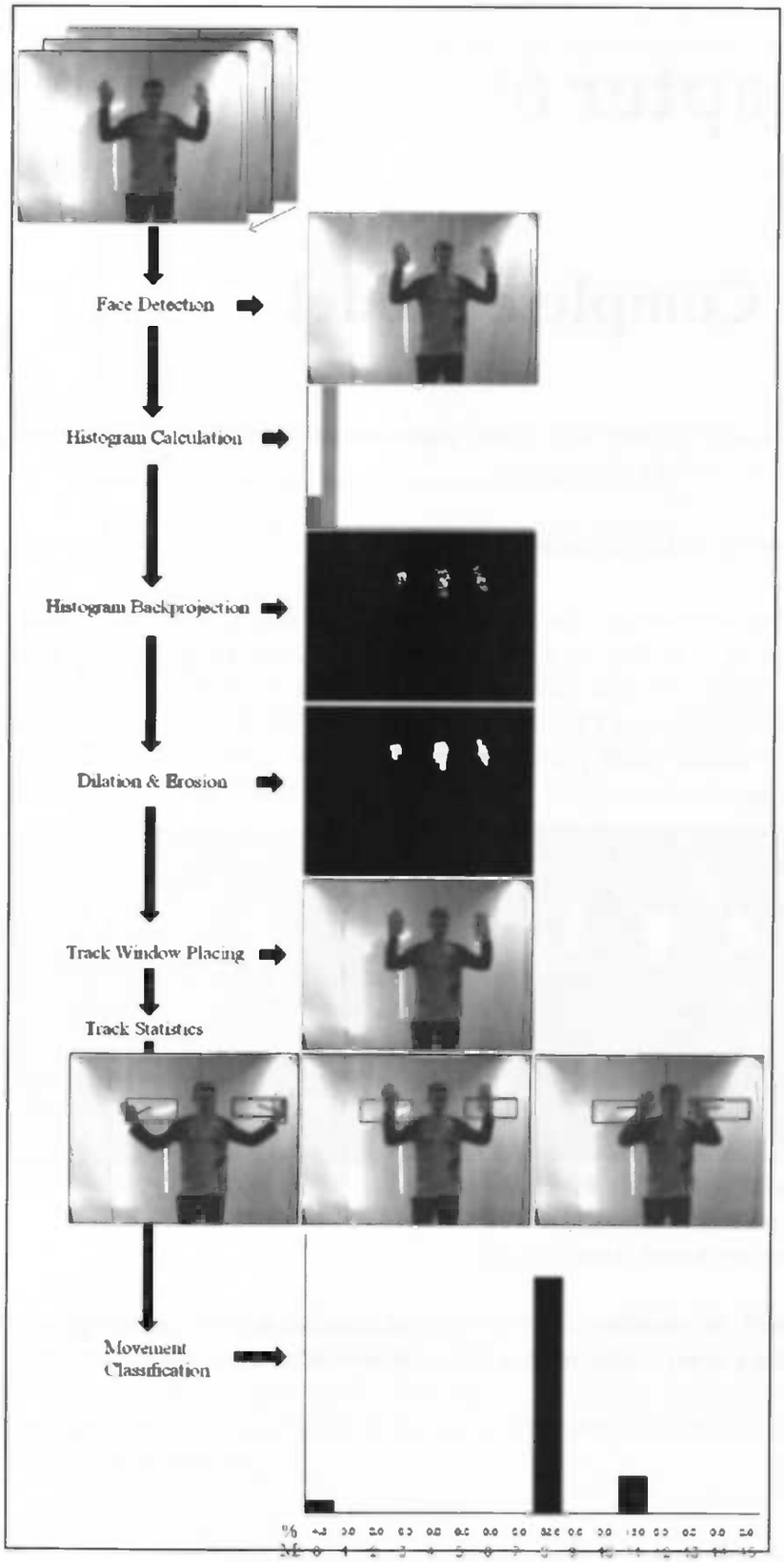

Figure 6.1: Overview of the movement classification model

6.2.1 Face detection

The first part of the program consists of a face detection algorithm. Using a trained classifier the program scans the first image of the input video sequence for faces. If no face can be found, or if there are multiple face candidates, the next camera frame is retrieved. Until a face is detected, the algorithm will keep searching for one. In chapter 3 the theory behind this face detection process has been discussed. Once a face has been found, the program moves on to the next step, which is to calculate a color histogram.
The program uses a prebuilt face classifier, which has been developed and tested over thousands of images by Intel. Experiments have been conducted with self-made face classifiers, but the bottleneck is the need of a vast amount of positive and negative examples (that is: over 1000 images each for an acceptable recognition rate). Therefore the choice to use an existing classifier was made.

6.2.2 Histogram calculation

A histogram of HSI-values, which stand for hue, saturation and intensity (sometimes V for value, or L for luminance are used), is calculated from the face subwindow. HSI is a different way to represent a color, other than RGB (red, green and blue). HSI is a preferred to RGB because for this problem, the dimensions better fit the color space that is observed mostly and therefore the most relevant.

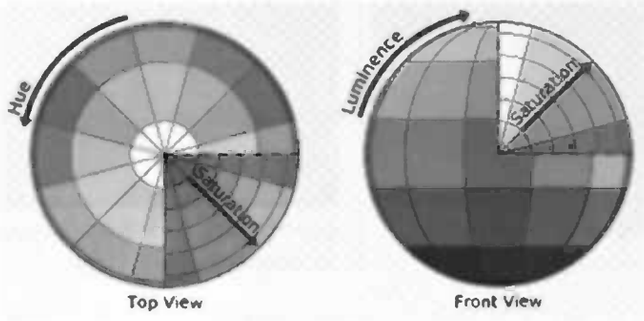

Figure 6.2: The HSI color representation

In the implementation, only the hue value is used. This abstraction is made because it will benefit the computational complexity of the algorithm. Furthermore, the color of human skin is represented in a good way merely by its

hue value. The average hue value for human skin lies between 0 and 45 degrees. A typical human skin color histogram will look similar to figure 6.3.

Figure 6.3: A typical human skin hue value distribution

6.2.3 Histogram backprojection

Now that the histogram is extracted from the face subwindow, the backprojection of this histogram on the original image is calculated. The resulting image consists of a 2-dimensional distribution of probabilities for every pixel to belong within the face color range. See figure 6.4a.

Figure 6.4a: A backprojection image

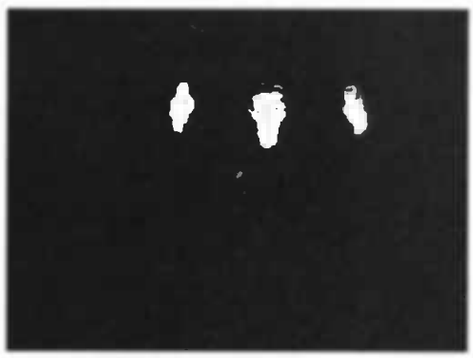

Figure 6.4b: A thresholded version of the backprojection image

The white pixels represent a high probability to belong within the face color range, whereas the black pixels represent a low probability. In this figure, the face and hands of the person can be distinguished. Using computer vision techniques like thresholding, dilation and erosion, the useful information can be extracted in a robust manner.

6.2.4 Thresholding, dilation and erosion

Three computer vision techniques that will be applied are thresholding, dilation and erosion. These techniques will be used to extract blob information from the backprojection image described above. Because the backprojection image itself is quite noisy and no obvious blobs can be abstracted from it, the image will be thresholded after which a number of dilation and erosion operations will be performed.

6.2.4.1 Thresholding

By thresholding the image, a binary classification is made for every pixel to belong to either class 0 (black) or 1 (white). The threshold can be set at any value, in this case it will be 0.5. In figure 6.4b the thresholded version of figure 6.4a is shown.

6.2.4.2 Dilation

Dilation is one of the two basic operators in the area of mathematical morphology (the other being erosion). It is typically applied to binary images. The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of white pixels. The dilation operator uses 2 pieces of data as inputs. The first is the thresholded image, and the second is a structuring element, also known as a kernel. In this case, a 3x3 square structuring element is used.

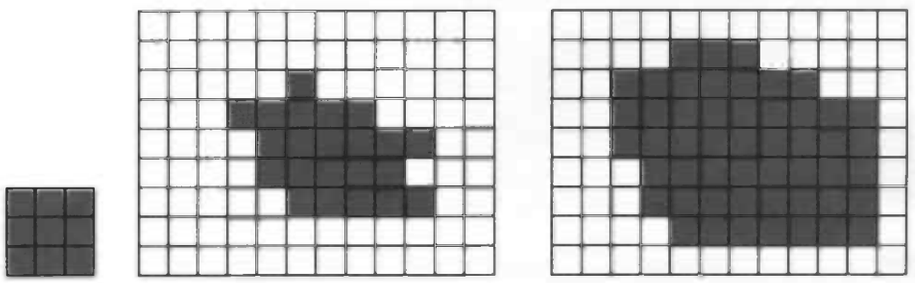

Figure 6.5: The 3x3 structuring element, the original binary image, and the resulting dilated image.

If A is the original image and B is the structuring element (both in Z^2), the definition of dilation is:

$$A \oplus B = \{x : B \text{ I } A \neq \emptyset\}$$

where Ø is the empty set.

In figure 6.6b, the thresholded image is shown after a number of dilation steps:

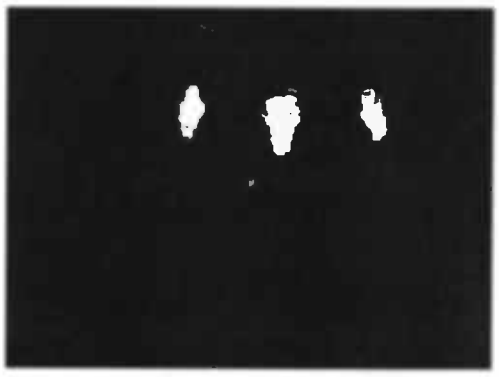

Figure 6.6a: The thresholded image

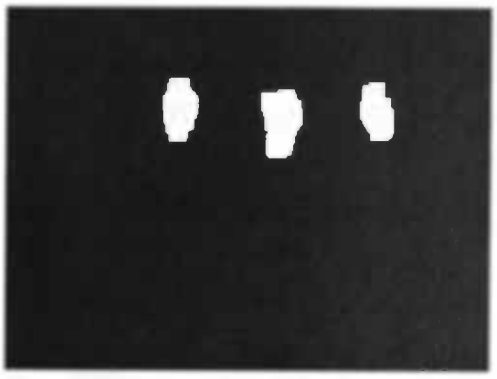

Figure 6.6b: The thresholded image, after a number of dilation steps

6.2.4.3 Erosion

Erosion is the action that does the opposite to dilation. See figure 6.6.
For sets A and B in Z^2, the erosion of A and B is defined as:

$$A \ominus B = \{x : B \subseteq A\}$$

The following example shows the result of an erosion step.

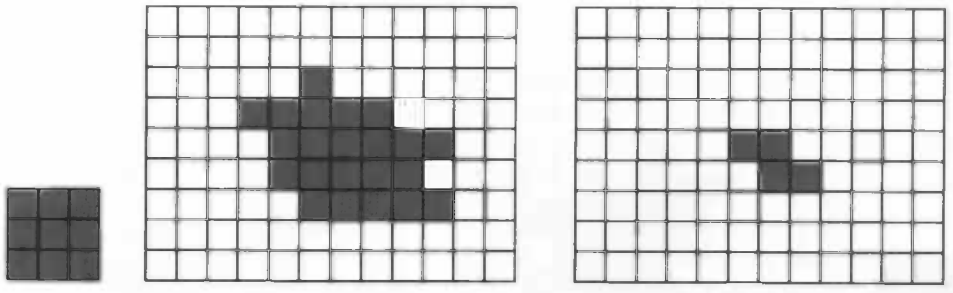

Figure 6.7: The 3x3 structuring element, the original binary image, and the resulting eroded image.

Erosion is used in this model to reduce the blob to its original size, after the dilation steps have removed noise and filled gaps if there were any.

6.2.5 Track window positioning

Once the head and hand blobs have been found, the locations are passed on to the next procedure, which is the positioning of track windows on the blobs. In chapter 4, the theory of blob tracking has been discussed. CamShift takes over from here, tracking one, two or three blobs as the movement is being performed. During this, various statistics such as the center of movement, minimum and maximum x and y values, are kept about the movement. These statistics will be the input for the classification algorithm.

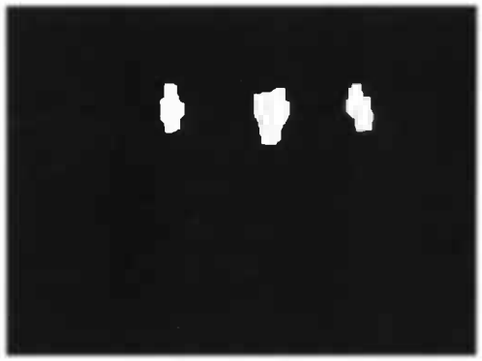

Figure 6.8a: The resulting three blobs

Figure 6.8b: The positioning of three track windows on the head and hand blobs

6.2.5 Decision tree-based classification

The input data to the decision tree (see chapter 5) are obtained by the CamShift window statistics. A classification is made on every frame of the image sequence, resulting in a distribution of classifications (in percentages). When no classification can be made, the algorithm outputs a 0. The results are displayed in the movement classification histogram, as shown in figure 6.9.

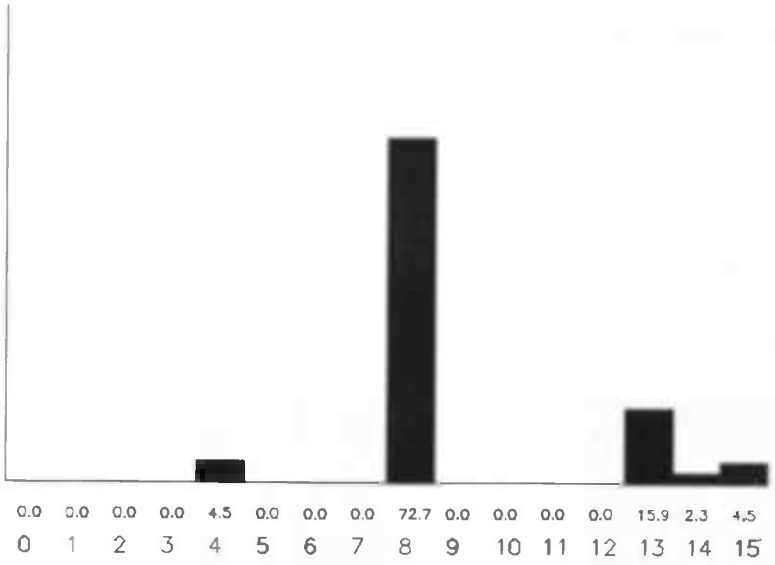

Figure 6.9: The result of the classification process is a histogram with percentages. In this case M8 is recognized correctly for 72.7% while resembling movements M4, M13, M14 and M15 are also recognized partially in this case.

6.3 Discussion

During the preparation, design, formalization and implementation of this project, a number of decisions had to be made about what methods would be the best to use for the problem at hand. Human movement classification is a complex task, with multiple sub-problems that all have their separate collection of techniques and methods. In the next sections the decisions will be discussed, as well as the inevitable disadvantages of the chosen methods.

6.3.1 Design decisions

The choice for face detection as a method of human recognition was made because this is a very robust method of deciding whether there is a human in front of the camera or not, and directly have access to the position of an important part: the face. The system can not be fooled easily. An added advantage of this choice is that the skin color information of the subject is available very fast, which improves the hand detection speed in a great way. Other human body representations that were discussed in chapter 2 were considered to be less favorable.

For tracking, CamShift has been chosen. A reason for this choice is that it works very well in real-time applications. Its calculations can be performed fast, so that the focus of the project can be put on other important sub-problems. Another reason is that the literature about CamShift shows very promising results.

The choice for a decision tree-based classification method is because of the practical consequences that CamShift has. The tracking process yields locations of the track windows on different points in time. The most natural way of processing these datapoints, is keeping statistics about them and using them as input for a decision tree. This is a rather straight-forward manner, which has proved to do the classification task well.

6.3.2 Disadvantages

Every design decision comes with disadvantages. In this section the biggest disadvantages of the design decisions will be discussed.
Firstly, the face detection task is rather computationally expensive and cannot be performed in the time that normally lies between two consecutive frames of a movie. Because it does not have to be executed on every frame (just the initial frame(s) until a face is found) this is not a very big problem.
The trained face classifier that was used in the human movement classification program, was provided by Intel's OpenCV library.

CamShift has the disadvantage that it sometimes loses track of a blob, when it is occluded or the velocity is too high. This has been a serious problem, and it caused the design of the experiments to restrict to non-occluded movements (that is: the hand does never disappear behind another body part). Secondly, the velocity of the movements had to be restricted. A proposed solution to this problem has been the prediction of blob movements, which has been discussed in section 4.3.
An implementation of CamShift was available in OpenCV, of which a modified version has been used for the human movement classification program.

The decision tree-based classification method has the disadvantage that it can only differ between a fixed number of movements. For instance, there is no way that the tree can learn a new movement. If a new movement were to be added, its properties would have to be hard-coded into the decision tree.
Another disadvantage is the generalization about movements that the decision tree makes. A movement which has been tagged to be mainly horizontal, can of course performed in a number of different ways. The temporal variations and subtle spatial differences can not be taken into account by the decision tree-based classification.

Chapter 7

Experiments

In order to test the Human Movement Classification program, the following experimental setup has been prepared. Three different persons will perform a fixed set of fifteen movements in front of the camera. Every subject will perform every movement five times. The set of fifteen movements that have been chosen is shown in the table below.

M1	Jumping
M2	Knee bending
M3	Clapping
M4	Vertically Waving Both Hands in Phase
M5	Vertically Waving Both Hands in Counterphase
M6	Clockwise Circular Movement with One Hand
M7	Counterclockwise Circular Movement with One Hand
M8	Waving Both Hands in Counterphase
M9	Drumming in Counterphase
M10	Waving Right Hand
M11	Waving Both Hands in Phase
M12	Drumming in Phase
M13	Vertically Waving Left Hand
M14	Vertically Waving Right Hand
M15	Waving Left Hand

Table 7.1: The fifteen experimental movements

This set of movements is chosen in such a way that the movements could carry some kind of information or command to a machine, e.g. waving the hand clockwise would imply an order for a robot to increase speed whereas a counterclockwise movement would imply the opposite. Other movements have been added in an attempt to create a set of movements that show enough variation in all possible directions. Furthermore, an important aspect is that the movements have to be substantially different, for the application to recognize the main differences between them.

7.1 Conditions

A number of conditions have to be met, in order to provide a good environment for the recognition program. Some of these conditions apply to the camera and surroundings, and others apply to the positioning of the person performing the movement.

Environmental conditions:

- The camera remains static while filming the movement.
- The lighting of the space in which the movement is performed remains constant.
- The background is static.

Personal conditions:

- There is always one person visible to the camera.
- The subject faces the camera while performing the movement.
- The subject always stays within the space that is visible to the camera.
- The face and hands of the subject are visible to the camera during the movement.

7.2 Results

Every movement in the test set has been performed 5 times by three different subjects. Of every movement, one periodic cycle has been taken to input into the movement recognition algorithm. The output is a resulting recognition distribution, showing in which of the fifteen classes the movement fits best, according to the classification algorithm. In appendix A, the recognition percentages of all movements are given. In this section, the results per subject and the total results will be given. The fifteen possible outcomes are shown on the top row of every table, plus a default zero output if the movement is not recognized at all.
To prove that the movements performed by the three subjects are statistically invariable, an F-statistic has been calculated for every movement. This statistic shows the ratio between the variance of the means of all groups and the mean of the variances of all groups. If that statistic has a lower value than

$F_{.05}([\#\text{ groups} - 1],[\text{total }\#\text{ subjects} - \#\text{ groups}]) = F_{.05}(2,12) = 3{,}89$

it can be safely assumed that the movements of all subjects are statistically invariable. See table 7.2.

M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F	0.08	0.29	1.94	1.39	1.99	0.91	0.34	0.24	0.28	1.56	0.16	1.2	2.42	1.16	0.91

Table 7.2

In figure 7.1, the confusion matrix of the classes is shown.

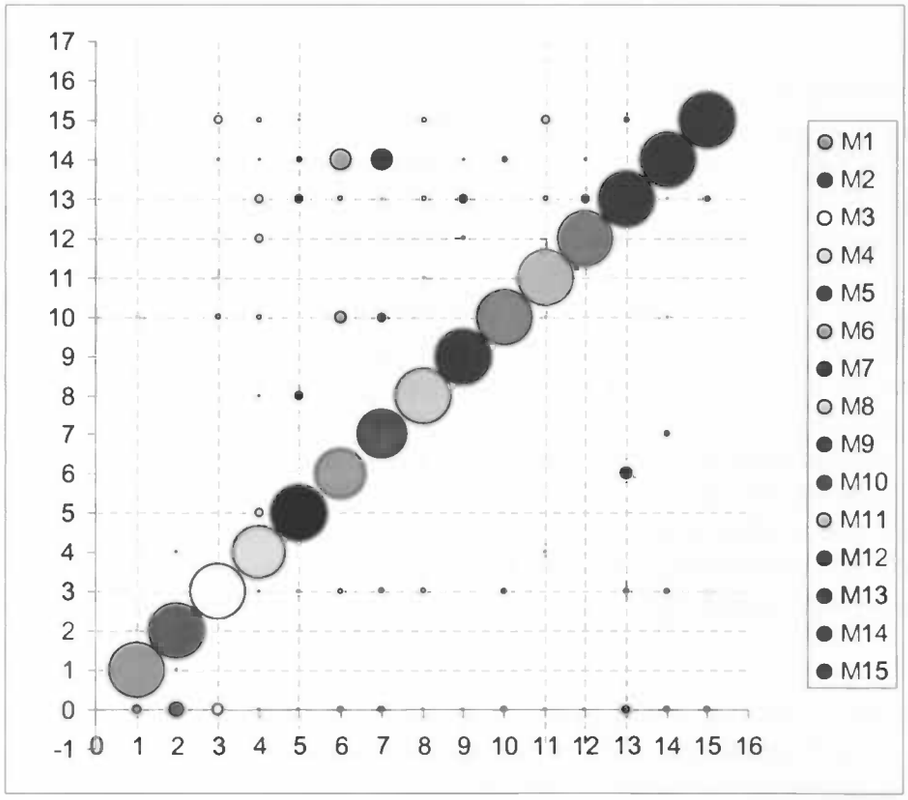

Figure 7.1: Confusion matrix

This matrix shows what movements are incorrectly classified in other classes than they should be. The horizontal axis shows the fifteen movements and the bubbles show the class in which they have been classified, with the bubble sizes representing the percentages. The exact percentages can be looked up in Appendix A.

It is noticeable that movements 6 and 7 (clockwise and counterclockwise circular movements) have the lowest recognition rate (76% and 76.1%). They are confused mostly (14.2% and 15.7%) with movement 14, which is vertically waving with one hand. The reason for this is that a circular movement can only be perceived when a certain part of the circle has been made. Up until one point the movement looks like a non-circular movement, until a large enough part of a circle can be distinguished. After that, the correct classification can be made.

Chapter 8

Discussion & Conclusion

In this thesis I have attempted to present an application that can perform the classification of human movements. The application is based on a model which was developed after the study of the literature of other researchers in this field. The choice has been made to detect a human subject by the presence of his/her face. This choice and other design decisions have been supported in section 6.3.1.

A learning technique has been discribed, which is based on the detection of features. Based on the color of the detected face, the algorithm searches for the hands of the subject. The assumption is made that the biggest two blobs resembling the color of the face, are in fact the hands. In an experimental environment, this assumption can be safely made because of the conditions that have been stated in section 7.1. If there would be more people in the scene, the application would have major trouble recognizing which blobs belong to the person whose movement is being observed. If the background were to change drastically during the movement, or the lighting would change, the program would also experience problems.

An experimental setup has been developed for this project, and the results show a good recognition rate with mean percentages ranging from 76% to 95,4%.
Reasons for the relatively lower recognition rates for some movements have been motivated in chapter 7. Over all, the recognition rate has been great.
Any suggestions for future work would be any improvement on one or more of the three subtasks face detection, tracking and classification.

For face detection one can think of an improvement in face detection speed, which I think will increase over the years as the computational power of computers in general increases. The quality of the algorithm itself might be improved by focusing on even more features, thereby giving the program more top-down knowledge about human faces. The tracking process has some problems still to be overcome, for example the problem of losing track of a blob when occlusion occurs. The idea of a blob prediction model has been suggested in this thesis, and could be a successful solution if it were to be examined and worked out in detail.

The quality of the classification output is very dependent on the quality of the tracking process. The parameters that have been mentioned in chapter 5 are also very important for the overall performance. The lack of adaptability of the classification algorithm is a weakness.

The decision tree-based classification method has the disadvantage that it can only differ between a fixed number of movements. For instance, there is no way that the tree can learn a new movement. If a new movement were to be added, its properties would have to be hard-coded into the decision tree.
Another disadvantage is the generalization about movements that the decision tree makes. A movement which has been tagged to be mainly horizontal, can of course performed in a number of different ways. The temporal variations and subtle spatial differences can not be taken into account by the decision tree-based classification.

Over all I think the chosen methods have proven to provide a good way to perform human movement classification. The results have been promising, and with a broad variety of areas in which human movement classification can be applied, this study could be a step closer to a better human-machine interaction in the future.

Appendix A

Group Results

Movement 1: Jumping

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	4.9	94.4	0.6	0	0.1	0	0	0	0	0	0	0	0	0	0	0
Mean P2	3.1	95.3	1.6	0	0	0	0	0	0	0	0	0	0	0	0	0
Mean P3	2.4	95.2	0.6	0	0.4	0.3	0	0	0	0	0.9	0	0	0	0.3	0
VAR P1	16	18.4	0.4	0	0.1	0	0	0	0	0	0	0	0	0	0	0
VAR P2	3.3	5.86	3.2	0	0	0	0	0	0	0	0	0	0	0	0	0
VAR P3	2.5	1.69	0.6	0	0.9	0.5	0	0	0	0	2.4	0	0	0	0.5	0
Mean	3.5	95	0.9	0	0.2	0.1	0	0	0	0	0.3	0	0	0	0.1	0
VAR	7.2	7.57	1.4	0	0.3	0.2	0	0	0	0	0.9	0	0	0	0.2	0
F		0.08														

Movement 2: Kneebending

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	8.2	0.7	91.1	0	0	0	0	0	0	0	0	0	0	0	0	0
Mean P2	6.3	0.1	93.6	0	0	0	0	0	0	0	0	0	0	0	0	0
Mean P3	6.6	0.1	91.4	0	1	0	0	0	0	0	0	0	0	0.6	0.3	0
VAR P1	37	1.6	39.6	0	0	0	0	0	0	0	0	0	0	0	0	0
VAR P2	13	0.1	13.8	0	0	0	0	0	0	0	0	0	0	0	0	0
VAR P3	11	0.1	1.8	0	5	0	0	0	0	0	0	0	0	0.3	0.4	0
Mean	7	0.3	92	0	0.33	0	0	0	0	0	0	0	0	0.2	0.1	0
VAR	18	0.6	17.1	0	1.67	0	0	0	0	0	0	0	0	0.2	0.1	0
F			0.29													

Movement 3: Clapping

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	12.4	0	0	76.4	0	0	0	0	0	0	1	1.4	0	0.1	1	7.6
Mean P2	2.88	0	0	94.3	0	0	0	0.7	0	0	2.14	0	0	0	0	0
Mean P3	4	0	0	95.6	0	0	0	0	0	0	0.14	0.1	0	0.1	0	0
VAR P1	347	0	0	507	0	0	0	0	0	0	2.24	2.9	0	0.1	2.4	108
VAR P2	1.82	0	0	7.07	0	0	0	2.5	0	0	16.1	0	0	0	0	0
VAR P3	14.7	0	0	15.7	0	0	0	0	0	0	0.1	0.1	0	0.1	0	0
Mean	6.44	0	0	88.8	0	0	0	0.2	0	0	1.09	0.5	0	0.1	0.3	2.5
VAR	123	0	0	233	0	0	0	0.8	0	0	6	1.3	0	0.1	0.9	44
F				1.94												

Movement 4: Vertically Waving Both Hands in Phase

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	0	0	0	0.1	96	0.1	0	0	0	0	0	0	0	2.4	0.1	1.1
Mean P2	2.3	0	0	0	73.3	4	0	0	0.9	0	6	0	7.7	1.7	1	3.1
Mean P3	0	0	0	1.6	81.5	6	0	0	0.3	0	0	0	3.4	5.1	0	2.1
VAR P1	0	0	0	0.1	0.72	0.1	0	0	0	0	0	0	0	1.5	0.1	2
VAR P2	26	0	0	0	636	30	0	0	3.7	0	76.8	0	22	1.2	5	4.5
VAR P3	0	0	0	12	221	68	0	0	0.2	0	0	0	58	38	0	4
Mean	0.8	0	0	0.6	83.6	3.4	0	0	0.4	0	2	0	3.7	3.1	0.4	2.1
VAR	8.7	0	0	4.1	340	34	0	0	1.2	0	30.5	0	34	14	1.7	3.7
F					1.39											

Movement 5: Vertically Waving Both Hands in Counterphase

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	0.3	0	0	0.1	0.1	95.6	0	0	0	0	0.3	0	0	1.7	1.4	0.4
Mean P2	0.7	0	0	0	0	89.9	0	0	5.3	0	0	0	0	1.8	1.9	0.4
Mean P3	0.3	0	0	2	0	83.7	0	0	5	0	0.3	0	0	7	1.7	0
VAR P1	0.2	0	0	0.1	0.1	1.14	0	0	0	0	0.4	0	0	1.2	1.8	0.2
VAR P2	0.5	0	0	0	0	64.4	0	0	27	0	0	0	0	5.5	4.9	0.9
VAR P3	0.4	0	0	0.9	0	94.5	0	0	18	0	0.4	0	0	42	1.2	0
Mean	0.4	0	0	0.7	0.1	89.7	0	0	3.4	0	0.2	0	0	3.5	1.7	0.3
VAR	0.3	0	0	1.2	0	71	0	0	19	0	0.2	0	0	21	2.3	0.3
F						1.99										

Movement 6: Clockwise Circular Movement with One Hand

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	4.3	0	0	1.7	0	0	76.4	0	0	0	11	0	0	0	6.86	0
Mean P2	0.4	0	0	3.1	0	0	75	0	0	0	1.3	0	0	3.3	16.9	0
Mean P3	0.1	0	0	1.1	0	0	76.4	0	0	0	1.3	0	0	1.9	19.2	0
VAR P1	1	0	0	9.5	0	0	1.51	0	0	0	59	0	0	0	17.4	0
VAR P2	0.4	0	0	4.3	0	0	4.52	0	0	0	3.7	0	0	5.3	12	0
VAR P3	0.1	0	0	0.4	0	0	0.49	0	0	0	1.3	0	0	0.7	5.22	0
Mean	1.6	0	0	2	0	0	76	0	0	0	4.4	0	0	1.7	14.3	0
VAR	4.3	0	0	4.8	0	0	2.34	0	0	0	40	0	0	3.6	40.5	0
F							0.91									

Movement 7: Counterclockwise Circular Movement with One Hand

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	2.44	0	0	0	0	0	0	76.6	0	0	9	0	0	0.7	11.3	0
Mean P2	1.28	0	0	3.7	0	0	0	76	0	0	1.1	0	0	1.3	16.6	0
Mean P3	2	0	0	1	0	0	0	75.7	0	0	1.7	0	0	0.3	19.3	0
VAR P1	1.23	0	0	0	0	0	0	0.62	0	0	5.4	0	0	2.6	5.92	0
VAR P2	0.91	0	0	11	0	0	0	3.45	0	0	6.5	0	0	4.5	23.5	0
VAR P3	2.22	0	0	2	0	0	0	0.7	0	0	7.2	0	0	0.5	10.3	0
Mean	1.91	0	0	1.6	0	0	0	76.1	0	0	4	0	0	0.8	15.7	0
VAR	1.49	0	0	6.5	0	0	0	1.49	0	0	19	0	0	2.3	23.2	0
F								0.34								

Movement 8: Waving Both Hands in Counterphase

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	1.4	0	0	1.58	0	0	0	0	93.4	0	0.14	0.7	0	0.1	0	2.6
Mean P2	0.7	0	0	1.72	0	0	0	0	94	0	0	0.6	0	0.6	0	2.5
Mean P3	0.7	0	0	2.58	0.28	0	0	0	91.9	0	0	0.7	0	2.4	0	1.4
VAR P1	0.5	0	0	12.5	0	0	0	0	8.9	0	0.1	0	0	0.1	0	2.2
VAR P2	0	0	0	14.8	0	0	0	0	12	0	0	0.1	0	0.3	0	1.7
VAR P3	0	0	0	33.3	0.39	0	0	0	24	0	0	0	0	7.7	0	3.1
Mean	0.9	0	0	1.96	0.09	0	0	0	93.1	0	0.05	0.7	0	1.1	0	2.2
VAR	0.3	0	0	17.5	0.13	0	0	0	13.7	0	0.03	0	0	3.4	0	2.3
F									0.24							

Movement 9: Drumming in Counterphase

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	0.7	0	0	0	0	0	0	0	0	94.9	0	0	0.4	3.4	0.6	0
Mean P2	1.1	0	0	0	0	0	0	0	0	94.7	0	0	0.3	3.2	0.7	0
Mean P3	0.7	0	0	0	0	0	0	0	0	93.6	0	0	0.7	4.2	0.8	0
VAR P1	0	0	0	0	0	0	0	0	0	3.57	0	0	0.2	3.2	0.3	0
VAR P2	1	0	0	0	0	0	0	0	0	3.58	0	0	0.2	7.7	2.5	0
VAR P3	0	0	0	0	0	0	0	0	0	8.82	0	0	0	6.7	0.3	0
Mean	0.9	0	0	0	0	0	0	0	0	94.4	0	0	0.5	3.6	0.7	0
VAR	0.3	0	0	0	0	0	0	0	0	4.92	0	0	0.1	5.2	0.9	0
F										0.28						

Movement 10: Waving Right Hand

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	2	0	0	0.7	0	0	0	0	0	0	97.3	0	0	0	0	0
Mean P2	3.1	0	0	0.9	0	0	0	0	0	0	91.9	0	0	0.6	3.6	0
Mean P3	1.8	0	0	3.4	0	0	0	0	0	0	94.2	0	0	0	0.6	0
VAR P1	1.5	0	0	0.7	0	0	0	0	0	0	0.92	0	0	0	0	0
VAR P2	5.7	0	0	1.4	0	0	0	0	0	0	29.5	0	0	0.6	29	0
VAR P3	3.2	0	0	15	0	0	0	0	0	0	12.6	0	0	0	0.8	0
Mean	2.3	0	0	1.7	0	0	0	0	0	0	94.4	0	0	0.2	1.4	0
VAR	3.3	0	0	6.6	0	0	0	0	0	0	17.6	0	0	0.2	11	0
F											1.56					

Movement 11: Waving Both Hands in Phase

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	1	0	0	0	0	0	0	0	0.4	0	0.8	92.7	0	0.1	0	4.9
Mean P2	1.1	0	0	0	0	0	0	0	0.1	0	0.3	92.4	0	0.9	0	5.2
Mean P3	1	0	0	0	1.3	0	0	0	0	0	0.3	93.3	0	2.7	0.4	1
VAR P1	0.2	0	0	0	0	0	0	0	0.9	0	0.8	1.63	0	0.1	0	3.4
VAR P2	0.2	0	0	0	0	0	0	0	0.1	0	0.4	4.24	0	1.7	0	7.9
VAR P3	0.2	0	0	0	2.2	0	0	0	0	0	0.4	4.25	0	5.6	1	1.5
Mean	1	0	0	0	0.4	0	0	0	0.2	0	0.5	92.8	0	1.2	0.2	3.7
VAR	0.1	0	0	0	1	0	0	0	0.3	0	0.5	3.02	0	3.4	0.3	7.5
F												0.16				

Movement 12: Drumming in Phase

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	0.7	0	0	0	0	0	0	0	0	0	0	0	95.2	4	0.1	0
Mean P2	0.7	0	0	0	0.6	0	0	0	0	0	0	0	95.1	3.3	0.3	0
Mean P3	1	0	0	0	0	0	0	0	0	0.6	0	0	93.3	4.6	0.4	0.1
VAR P1	0	0	0	0	0	0	0	0	0	0	0	0	1.91	2	0.1	0
VAR P2	0	0	0	0	1.7	0	0	0	0	0	0	0	4.97	4	0.2	0
VAR P3	0.4	0	0	0	0	0	0	0	0	1.7	0	0	1.71	2.3	0.4	0.1
Mean	0.8	0	0	0	0.2	0	0	0	0	0.2	0	0	94.5	4	0.3	0.1
VAR	0.1	0	0	0	0.6	0	0	0	0	0.6	0	0	3.27	2.7	0.2	0
F													1.2			

Movement 13: Vertically Waving Left Hand

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	1	0	0	0	0	0	0	0	0	0	0	0	0	98.7	0.3	0
Mean P2	4.6	0	0	3.1	0	0	0	0	0	0	0.7	0	0	90	0.1	1.4
Mean P3	2.6	0	0	1.3	0	0	13	0	0	0	0	0	0	80.3	0.1	2.4
VAR P1	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0.34	0.2	0
VAR P2	56	0	0	20	0	0	0	0	0	0	1	0	0	98.9	0.1	7.8
VAR P3	6.6	0	0	3.2	0	0	354	0	0	0	0	0	0	217	0.1	18
Mean	2.7	0	0	1.5	0	0	4.4	0	0	0	0.2	0	0	89.7	0.2	1.3
VAR	20	0	0	8.3	0	0	143	0	0	0	0.4	0	0	151	0.1	8.5
F														2.42		

Movement 14: Vertically Waving Right Hand

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	3.7	0	0	0.6	0	0	0	0	0	0	0	0	0	0.4	95.3	0
Mean P2	2.4	0	0	0.6	0	0	0	0	0	0	0	0	0	0.6	96.4	0
Mean P3	0.7	0	0	3.1	0.1	0.1	0	6.9	0	0	1.2	0	0	1.3	86.6	0
VAR P1	35	0	0	1.7	0	0	0	0	0	0	0	0	0	0.4	30.2	0
VAR P2	1.5	0	0	0.8	0	0	0	0	0	0	0	0	0	1.7	0.77	0
VAR P3	0	0	0	20	0.1	0.1	0	91	0	0	2.5	0	0	1.3	193	0
Mean	2.3	0	0	1.4	0.1	0.1	0	2.3	0	0	0.4	0	0	0.8	92.8	0
VAR	12	0	0	7.9	0	0	0	37	0	0	1	0	0	1.1	84.7	0
F															1.16	

Movement 15: Waving Left Hand

Mov. #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean P1	2.6	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	97
Mean P2	1.3	0	0	0.9	0	0	0	0	0	0	0	0	0	3	0	94.9
Mean P3	1.6	0	0	0.1	0	0	0	0	0	0	0	0	0	3.6	0.6	94.2
VAR P1	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0.65
VAR P2	0.3	0	0	2.4	0	0	0	0	0	0	0	0	0	17	0	10.6
VAR P3	0.9	0	0	0.1	0	0	0	0	0	0	0	0	0	18	1.7	10.5
Mean	1.8	0	0	0.3	0	0	0	0	0	0	0	0	0	2.3	0.2	95.4
VAR	0.8	0	0	0.9	0	0	0	0	0	0	0	0	0	12	0.6	7.81
F																0.91

Appendix B

All Results

In this section, all the results are shown in fifteen tables. The numbers shown in the tables are the percentages of classification into the classes (shown in the top row) by the algorithm. The total mean and variance are given in the bottom two rows. Subject 1 is shown in movies *, A, B, C and D; subject 2 is shown in E through I, and subject 3 is shown in J through N.

M1: Jumping																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*	2.2	97.1	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0
A	4.3	95	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0
B	11	87.1	1.5	0	0	0	0	0	0	0	0	0	0	0	0	0
C	5	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	1.4	97.9	0	0	0.7	0	0	0	0	0	0	0	0	0	0	0
E	3.6	95	1.4	0	0	0	0	0	0	0	0	0	0	0	0	0
F	1.4	94.3	4.3	0	0	0	0	0	0	0	0	0	0	0	0	0
G	1.4	98.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	3.6	96.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	5.7	92.1	2.2	0	0	0	0	0	0	0	0	0	0	0	0	0
J	4.3	95.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	1.4	95.1	1.4	0	2.1	0	0	0	0	0	0	0	0	0	0	0
L	1.4	97.1	0	0	0	1.5	0	0	0	0	0	0	0	0	0	0
M	0.9	94.1	1.4	0	0	0	0	0	0	0	3.6	0	0	0	0	0
N	3.8	93.9	0	0	0	0	0	0	0	0	0.8	0	0	0	1.5	0
Mean	3.5	95	0.9	0	0.2	0.1	0	0	0	0	0.3	0	0	0	0.1	0
VAR	7.2	7.57	1.4	0	0.3	0.2	0	0	0	0	0.9	0	0	0	0.2	0

M2: Kneebending																
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*		18.6	0.7	80.7	0	0	0	0	0	0	0	0	0	0	0	0	0
A		3.6	0	96.4	0	0	0	0	0	0	0	0	0	0	0	0	0
B		7.9	0	92.1	0	0	0	0	0	0	0	0	0	0	0	0	0
C		4.3	0	95.7	0	0	0	0	0	0	0	0	0	0	0	0	0
D		6.4	2.9	90.7	0	0	0	0	0	0	0	0	0	0	0	0	0
E		5.7	0	94.3	0	0	0	0	0	0	0	0	0	0	0	0	0
F		2.9	0	97.1	0	0	0	0	0	0	0	0	0	0	0	0	0
G		7.1	0.7	92.2	0	0	0	0	0	0	0	0	0	0	0	0	0
H		12.1	0	87.9	0	0	0	0	0	0	0	0	0	0	0	0	0
I		3.6	0	96.4	0	0	0	0	0	0	0	0	0	0	0	0	0
J		7.9	0	90.7	0	0	0	0	0	0	0	0	0	0	0	1.4	0
K		8.6	0	90	0	0	0	0	0	0	0	0	0	0	1.4	0	0
L		7.9	0	91.4	0	0	0	0	0	0	0	0	0	0	0.7	0	0
M		0.7	0.7	93.6	0	5	0	0	0	0	0	0	0	0	0	0	0
N		7.9	0	91.4	0	0	0	0	0	0	0	0	0	0	0.7	0	0
Mean		7.01	0.33	92.04	0	0.33	0	0	0	0	0	0	0	0	0.19	0.09	0
VAR		18.24	0.59	17.06	0	1.67	0	0	0	0	0	0	0	0	0.17	0.13	0

M3: Clapping																	
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*		42.9	0	0	50	0	0	0	0	0	0	3.6	0	0	0	3.5	0
A		0.7	0	0	97.9	0	0	0	0	0	0	0	0	0	0	1.4	0
B		0.7	0	0	77.1	0	0	0	0	0	0	0.7	3.6	0	0	0	17.9
C		0	0	0	99.3	0	0	0	0	0	0	0	0.7	0	0	0	0
D		17.9	0	0	57.9	0	0	0	0	0	0	0.7	2.9	0	0.7	0	19.9
E		2.9	0	0	93.6	0	0	0	3.5	0	0	0	0	0	0	0	0
F		2.9	0	0	96.4	0	0	0	0	0	0	0.7	0	0	0	0	0
G		4.3	0	0	95	0	0	0	0	0	0	0.7	0	0	0	0	0
H		0.7	0	0	90	0	0	0	0	0	0	9.3	0	0	0	0	0
I		3.6	0	0	96.4	0	0	0	0	0	0	0	0	0	0	0	0
J		10	0	0	89.3	0	0	0	0	0	0	0	0	0	0.7	0	0
K		4.3	0	0	95	0	0	0	0	0	0	0.7	0	0	0	0	0
L		4.3	0	0	95.7	0	0	0	0	0	0	0	0	0	0	0	0
M		0	0	0	99.3	0	0	0	0	0	0	0	0.7	0	0	0	0
N		1.4	0	0	98.6	0	0	0	0	0	0	0	0	0	0	0	0
Mean		6.44	0	0	88.77	0	0	0	0.23	0	0	1.09	0.53	0	0.09	0.33	2.52
VAR		123.39	0	0	232.9	0	0	0	0.82	0	0	6	1.3	0	0.06	0.9	44.4

M4: Vert. Waving 2 Hands in Phase																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*	0	0	0	0	95	0.7	0	0	0	0	0	0	0	0.7	0	3.6
A	0	0	0	0	95.7	0	0	0	0	0	0	0	0	3.6	0	0.7
B	0	0	0	0.7	97.2	0	0	0	0	0	0	0	0	2.1	0	0
C	0	0	0	0	96.5	0	0	0	0	0	0	0	0	2.1	0.7	0.7
D	0	0	0	0	95.7	0	0	0	0	0	0	0	0	3.6	0	0.7
E	0	0	0	0	46.4	10	0	0	4.3	0	19.3	0	12.9	1.4	0	5.7
F	11.4	0	0	0	45	10	0	0	0	0	10.7	0	12.9	2.9	5	2.1
G	0	0	0	0	90.7	0	0	0	0	0	0	0	4.3	0	0	5
H	0	0	0	0	91.5	0	0	0	0	0	0	0	4.3	2.1	0	2.1
I	0	0	0	0	92.9	0	0	0	0	0	0	0	4.3	2.1	0	0.7
J	0	0	0	0	97.2	0	0	0	0.7	0	0	0	0	1.4	0	0.7
K	0	0	0	7.9	74.3	0	0	0	0.7	0	0	0	17.1	0	0	0
L	0	0	0	0	67.9	15	0	0	0	0	0	0	0	12.9	0	4.2
M	0	0	0	0	97.9	0	0	0	0	0	0	0	0	0.7	0	1.4
N	0	0	0	0	70	15	0	0	0	0	0	0	0	10.7	0	4.3
Mean	0.76	0	0	0.57	83.59	3.38	0	0	0.38	0	2	0	3.72	3.09	0.38	2.13
VAR	8.66	0	0	4.14	339.6	34.22	0	0	1.24	0	30.5	0	33.8	13.9	1.67	3.71

M5: Vert. Waving 2 Hands in Counterphase																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*	0	0	0	0	0	97.2	0	0	0	0	0	0	0	2.1	0	0.7
A	0	0	0	0.7	0.7	95.7	0	0	0	0	0	0	0	2.9	0	0
B	0	0	0	0	0	94.3	0	0	0	0	0	0	0	2.1	2.9	0.7
C	0.7	0	0	0	0	95.8	0	0	0	0	1.4	0	0	0	2.1	0
D	0.7	0	0	0	0	95.1	0	0	0	0	0	0	0	1.4	2.1	0.7
E	0	0	0	0	0	97.2	0	0	0	0	0	0	0	0.7	0	2.1
F	0.7	0	0	0	0	87.2	0	0	10	0	0	0	0	0	2.1	0
G	0	0	0	0	0	94.3	0	0	5.7	0	0	0	0	0	0	0
H	1.4	0	0	0	0	77.1	0	0	10.7	0	0	0	0	5.4	5.4	0
I	1.4	0	0	0	0	93.6	0	0	0	0	0	0	0	2.9	2.1	0
J	0	0	0	3.6	0	77.1	0	0	7.1	0	0	0	0	10.7	1.5	0
K	0	0	0	1.4	0	95	0	0	0.7	0	0	0	0	0	2.9	0
L	0	0	0	1.4	0	76.4	0	0	8.6	0	0	0	0	13.6	0	0
M	1.4	0	0	1.4	0	93.7	0	0	0	0	1.4	0	0	0	2.1	0
N	0	0	0	2.1	0	76.4	0	0	8.6	0	0	0	0	10.7	2.2	0
Mean	0.42	0	0	0.71	0.05	89.74	0	0	3.43	0	0.19	0	0	3.5	1.69	0.28
VAR	0.34	0	0	1.16	0.03	71	0	0	19.26	0	0.24	0	0	20.5	2.29	0.34

M6: Clockwise Circular Movement																	
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*		5	0	0	0	0	0	74.3	0	0	0	17.1	0	0	0	3.6	0
A		5	0	0	7.1	0	0	77.1	0	0	0	0	0	0	0	10.8	0
B		5	0	0	0	0	0	77.1	0	0	0	6.5	0	0	0	11.4	0
C		3.6	0	0	1.4	0	0	77.2	0	0	0	11.4	0	0	0	6.4	0
D		2.9	0	0	0	0	0	76.4	0	0	0	18.6	0	0	0	2.1	0
E		0.7	0	0	3.6	0	0	72.9	0	0	0	2.1	0	0	2.1	18.6	0
F		0	0	0	5.7	0	0	77.1	0	0	0	0	0	0	0	17.2	0
G		1.4	0	0	4.3	0	0	75	0	0	0	4.3	0	0	3.6	11.4	0
H		0	0	0	1.4	0	0	77.2	0	0	0	0	0	0	5	16.4	0
I		0	0	0	0.7	0	0	72.9	0	0	0	0	0	0	5.7	20.7	0
J		0	0	0	0.7	0	0	75.7	0	0	0	0	0	0	1.5	22.1	0
K		0.7	0	0	1.4	0	0	77.1	0	0	0	2.1	0	0	2.9	15.8	0
L		0	0	0	0.7	0	0	76.4	0	0	0	2.1	0	0	2.1	18.7	0
M		0	0	0	2.1	0	0	75.8	0	0	0	2.1	0	0	0.7	19.3	0
N		0	0	0	0.7	0	0	77.2	0	0	0	0	0	0	2.1	20	0
Mean		1.62	0	0	1.99	0	0	75.96	0	0	0	4.42	0	0	1.71	14.3	0
VAR		4.28	0	0	4.81	0	0	2.34	0	0	0	39.55	0	0	3.63	40.5	0

M7: Counterclockwise Circular Movement																	
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*		2.1	0	0	0	0	0	0	77.2	0	0	12.1	0	0	0	8.6	0
A		2.9	0	0	0	0	0	0	75.7	0	0	6.4	0	0	0	15	0
B		0.7	0	0	0	0	0	0	77.1	0	0	7.9	0	0	3.6	10.7	0
C		3.6	0	0	0	0	0	0	75.7	0	0	10.7	0	0	0	10	0
D		2.9	0	0	0	0	0	0	77.1	0	0	7.9	0	0	0	12.1	0
E		2.9	0	0	0	0	0	0	75.7	0	0	0	0	0	0	21.4	0
F		0.7	0	0	0	0	0	0	72.9	0	0	0	0	0	5	21.4	0
G		0.7	0	0	6.4	0	0	0	77.1	0	0	5.7	0	0	0	10.1	0
H		0.7	0	0	5.7	0	0	0	77.3	0	0	0	0	0	0.7	15.6	0
I		1.4	0	0	6.4	0	0	0	77.1	0	0	0	0	0	0.7	14.4	0
J		0.7	0	0	2.1	0	0	0	75.1	0	0	1.4	0	0	0	20.7	0
K		3.6	0	0	0	0	0	0	75	0	0	0	0	0	0	21.4	0
L		3.6	0	0	0	0	0	0	75.7	0	0	6.4	0	0	0	14.3	0
M		1.4	0	0	2.9	0	0	0	77.1	0	0	0.7	0	0	0	17.9	0
N		0.7	0	0	0	0	0	0	75.7	0	0	0	0	0	1.5	22.1	0
Mean		1.91	0	0	1.57	0	0	0	76.1	0	0	3.95	0	0	0.77	15.7	0
VAR		1.49	0	0	6.46	0	0	0	1.49	0	0	19.21	0	0	2.31	23.2	0

M8: Waving 2 Hands in Counterphase																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*	0.7	0	0	0	0	0	0	0	96.4	0	0	0.7	0	0.7	0	1.5
A	1.4	0	0	0	0	0	0	0	92.9	0	0.7	0.7	0	0	0	4.3
B	2.1	0	0	0	0	0	0	0	94.3	0	0	0.7	0	0	0	2.9
C	0.7	0	0	0	0	0	0	0	95	0	0	0.7	0	0	0	3.6
D	2.1	0	0	7.9	0	0	0	0	88.6	0	0	0.7	0	0	0	0.7
E	0.7	0	0	0	0	0	0	0	95.7	0	0	0.7	0	0	0	2.9
F	0.7	0	0	8.6	0	0	0	0	87.9	0	0	0.7	0	1.4	0	0.7
G	0.7	0	0	0	0	0	0	0	95	0	0	0.7	0	0	0	3.6
H	0.7	0	0	0	0	0	0	0	96.4	0	0	0.7	0	0.7	0	1.5
I	0.7	0	0	0	0	0	0	0	95	0	0	0	0	0.7	0	3.6
J	0.7	0	0	12.9	0	0	0	0	83.6	0	0	0.7	0	0.7	0	1.4
K	0.7	0	0	0	0	0	0	0	94.3	0	0	0.7	0	0	0	4.3
L	0.7	0	0	0	0	0	0	0	92.2	0	0	0.7	0	6.4	0	0
M	0.7	0	0	0	0	0	0	0	96.4	0	0	0.7	0	0.8	0	1.4
N	0.7	0	0	0	1.4	0	0	0	92.9	0	0	0.7	0	4.3	0	0
Mean	0.93	0	0	1.96	0.09	0	0	0	93.11	0	0.05	0.65	0	1.05	0	2.16
VAR	0.26	0	0	17.51	0.13	0	0	0	13.68	0	0.03	0.03	0	3.4	0	2.29

M9: Drumming in Counterphase																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*	0.7	0	0	0	0	0	0	0	0	96.4	0	0	0	2.2	0.7	0
A	0.7	0	0	0	0	0	0	0	0	92.2	0	0	0.7	5	1.4	0
B	0.7	0	0	0	0	0	0	0	0	93.6	0	0	0	5.7	0	0
C	0.7	0	0	0	0	0	0	0	0	95.8	0	0	0.7	2.1	0.7	0
D	0.7	0	0	0	0	0	0	0	0	96.4	0	0	0.8	2.1	0	0
E	0.7	0	0	0	0	0	0	0	0	97.1	0	0	0.7	1.5	0	0
F	0.7	0	0	0	0	0	0	0	0	96.4	0	0	0.7	2.2	0	0
G	0.7	0	0	0	0	0	0	0	0	92.9	0	0	0	6.4	0	0
H	0.7	0	0	0	0	0	0	0	0	93.6	0	0	0	5.7	0	0
I	2.9	0	0	0	0	0	0	0	0	93.6	0	0	0	0	3.5	0
J	0.7	0	0	0	0	0	0	0	0	93.6	0	0	0.7	4.3	0.7	0
K	0.7	0	0	0	0	0	0	0	0	94.3	0	0	0.7	2.9	1.4	0
L	0.7	0	0	0	0	0	0	0	0	88.6	0	0	0.7	8.6	1.4	0
M	0.7	0	0	0	0	0	0	0	0	95	0	0	0.7	2.9	0.7	0
N	0.7	0	0	0	0	0	0	0	0	96.4	0	0	0.7	2.2	0	0
Mean	0.85	0	0	0	0	0	0	0	0	94.39	0	0	0.47	3.59	0.7	0
VAR	0.32	0	0	0	0	0	0	0	0	4.92	0	0	0.12	5.2	0.91	0

M10: Waving Right Hand																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*	1.4	0	0	0.7	0	0	0	0	0	0	97.9	0	0	0	0	0
A	1.4	0	0	0	0	0	0	0	0	0	98.6	0	0	0	0	0
B	2.9	0	0	0.7	0	0	0	0	0	0	96.4	0	0	0	0	0
C	0.7	0	0	2.1	0	0	0	0	0	0	97.2	0	0	0	0	0
D	3.6	0	0	0	0	0	0	0	0	0	96.4	0	0	0	0	0
E	7.1	0	0	0	0	0	0	0	0	0	87.1	0	0	0	5.8	0
F	2.9	0	0	0	0	0	0	0	0	0	85	0	0	0	12.1	0
G	2.9	0	0	0.7	0	0	0	0	0	0	95	0	0	1.4	0	0
H	0.7	0	0	2.9	0	0	0	0	0	0	95	0	0	1.4	0	0
I	2.1	0	0	0.7	0	0	0	0	0	0	97.2	0	0	0	0	0
J	1.4	0	0	5	0	0	0	0	0	0	93.6	0	0	0	0	0
K	5	0	0	0	0	0	0	0	0	0	92.9	0	0	0	2.1	0
L	1.4	0	0	0	0	0	0	0	0	0	98.6	0	0	0	0	0
M	0.7	0	0	9.3	0	0	0	0	0	0	89.3	0	0	0	0.7	0
N	0.7	0	0	2.9	0	0	0	0	0	0	96.4	0	0	0	0	0
Mean	2.33	0	0	1.67	0	0	0	0	0	0	94.44	0	0	0.19	1.38	0
VAR	3.33	0	0	6.64	0	0	0	0	0	0	17.6	0	0	0.24	11.2	0

M11: Waving 2 Hands in Phase																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*	0.7	0	0	0	0	0	0	0	0	0	0	91.4	0	0	0	7.9
A	1.4	0	0	0	0	0	0	0	0	0	2.1	91.5	0	0	0	5
B	1.4	0	0	0	0	0	0	0	0	0	0.7	94.3	0	0.7	0	2.9
C	0.7	0	0	0	0	0	0	0	2.1	0	0	92.9	0	0	0	4.3
D	0.7	0	0	0	0	0	0	0	0	0	1.4	93.6	0	0	0	4.3
E	1.4	0	0	0	0	0	0	0	0.7	0	0	92.9	0	0	0	5
F	0.7	0	0	0	0	0	0	0	0	0	0	92.1	0	2.9	0	4.3
G	1.4	0	0	0	0	0	0	0	0	0	0	95	0	0	0	3.6
H	0.7	0	0	0	0	0	0	0	0	0	0	89.3	0	0	0	10
I	1.4	0	0	0	0	0	0	0	0	0	1.4	92.9	0	1.4	0	2.9
J	1.4	0	0	0	0	0	0	0	0	0	1.4	95	0	0.8	0	1.4
K	0.7	0	0	0	0.7	0	0	0	0	0	0	90.7	0	5.7	2.2	0
L	0.7	0	0	0	2.9	0	0	0	0	0	0	92.9	0	2.8	0	0.7
M	1.4	0	0	0	0	0	0	0	0	0	0	95.7	0	0	0	2.9
N	0.7	0	0	0	2.9	0	0	0	0	0	0	92.1	0	4.3	0	0
Mean	1.03	0	0	0	0.43	0	0	0	0.19	0	0.47	92.82	0	1.24	0.15	3.68
VAR	0.13	0	0	0	1.04	0	0	0	0.31	0	0.54	3.02	0	3.38	0.32	7.53

M12: Drumming in Phase																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*	0.7	0	0	0	0	0	0	0	0	0	0	0	95.7	3.6	0	0
A	0.7	0	0	0	0	0	0	0	0	0	0	0	93.6	5.7	0	0
B	0.7	0	0	0	0	0	0	0	0	0	0	0	95	3.6	0.7	0
C	0.7	0	0	0	0	0	0	0	0	0	0	0	97.2	2.1	0	0
D	0.7	0	0	0	0	0	0	0	0	0	0	0	94.3	5	0	0
E	0.7	0	0	0	0	0	0	0	0	0	0	0	92.9	5.7	0.7	0
F	0.7	0	0	0	0	0	0	0	0	0	0	0	94.3	5	0	0
G	0.7	0	0	0	0	0	0	0	0	0	0	0	97.9	1.4	0	0
H	0.7	0	0	0	0	0	0	0	0	0	0	0	97.1	1.4	0.8	0
I	0.7	0	0	0	2.9	0	0	0	0	0	0	0	93.5	2.9	0	0
J	0.7	0	0	0	0	0	0	0	0	0	0	0	95	4.3	0	0
K	0.7	0	0	0	0	0	0	0	0	0	0	0	93.6	5.7	0	0
L	0.7	0	0	0	0	0	0	0	0	0	0	0	93.6	5.7	0	0
M	0.7	0	0	0	0	0	0	0	0	2.9	0	0	92.9	2.1	1.4	0
N	2.1	0	0	0	0	0	0	0	0	0	0	0	91.4	5.1	0.7	0.7
Mean	0.79	0	0	0	0.19	0	0	0	0	0.19	0	0	94.5	3.95	0.29	0.05
VAR	0.13	0	0	0	0.56	0	0	0	0	0.56	0	0	3.27	2.65	0.2	0.03

M13: Vert. Waving Left Hand																
	0	**1**	**2**	**3**	**4**	**5**	**6**	7	**8**	**9**	**10**	**11**	12	**13**	**14**	**15**
*	1.4	0	0	0	0	0	0	0	0	0	0	0	0	97.9	0.7	0
A	0.7	0	0	0	0	0	0	0	0	0	0	0	0	99.3	0	0
B	0.7	0	0	0	0	0	0	0	0	0	0	0	0	98.6	0.7	0
C	0.7	0	0	0	0	0	0	0	0	0	0	0	0	99.3	0	0
D	1.4	0	0	0	0	0	0	0	0	0	0	0	0	98.6	0	0
E	0.7	0	0	0	0	0	0	0	0	0	0	0	0	98.6	0.7	0
F	2.1	0	0	0	0	0	0	0	0	0	1.4	0	0	96.5	0	0
G	0.7	0	0	6.4	0	0	0	0	0	0	0	0	0	92.9	0	0
H	17.9	0	0	0	0	0	0	0	0	0	2.1	0	0	73.6	0	6.4
I	1.4	0	0	9.3	0	0	0	0	0	0	0	0	0	88.6	0	0.7
J	0.7	0	0	0	0	0	40	0	0	0	0	0	0	58.6	0	0.7
K	5.7	0	0	0	0	0	0	0	0	0	0	0	0	94.3	0	0
L	5	0	0	3.6	0	0	0	0	0	0	0	0	0	90.7	0	0.7
M	0.7	0	0	0	0	0	26.4	0	0	0	0	0	0	72.2	0	0.7
N	0.7	0	0	2.9	0	0	0	0	0	0	0	0	0	85.7	0.7	10
Mean	2.7	0	0	1.48	0	0	4.43	0	0	0	0.23	0	0	89.7	0.19	1.28
VAR	20.2	0	0	8.28	0	0	143.1	0	0	0	0.4	0	0	151	0.1	8.45

M14: Vert. Waving Right Hand																
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*		1.4	0	0	0	0	0	0	0	0	0	0	0	0	0	98.6	0
A		14.3	0	0	0	0	0	0	0	0	0	0	0	0	0	85.7	0
B		0.7	0	0	0	0	0	0	0	0	0	0	0	0	0.7	98.6	0
C		0.7	0	0	0	0	0	0	0	0	0	0	0	0	1.4	97.9	0
D		1.4	0	0	2.9	0	0	0	0	0	0	0	0	0	0	95.7	0
E		0.7	0	0	0	0	0	0	0	0	0	0	0	0	2.9	96.4	0
F		2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	97.9	0
G		3.6	0	0	0	0	0	0	0	0	0	0	0	0	0	96.4	0
H		2.1	0	0	2.1	0	0	0	0	0	0	0	0	0	0	95.8	0
I		3.6	0	0	0.7	0	0	0	0	0	0	0	0	0	0	95.7	0
J		0.7	0	0	6.4	0	0	0	19.3	0	0	2.9	0	0	0	70.7	0
K		0.7	0	0	0	0.7	0.7	0	0	0	0	0	0	0	2.1	95.8	0
L		0.7	0	0	0	0	0	0	0	0	0	0	0	0	2.1	97.2	0
M		0.7	0	0	9.3	0	0	0	15	0	0	2.9	0	0	0	72.1	0
N		0.7	0	0	0	0	0	0	0	0	0	0	0	0	2.1	97.2	0
Mean		2.27	0	0	1.43	0.05	0.05	0	2.29	0	0	0.39	0	0	0.75	92.8	0
VAR		12.11	0	0	7.87	0.03	0.03	0	37.08	0	0	1.04	0	0	1.11	84.7	0

M15: Waving Left Hand																	
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
*		3.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.4
A		2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.9
B		2.9	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	96.4
C		2.1	0	0	0	0	0	0	0	0	0	0	0	0	1.4	0	96.5
D		2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.9
E		0.7	0	0	0	0	0	0	0	0	0	0	0	0	7.9	0	91.4
F		0.7	0	0	3.6	0	0	0	0	0	0	0	0	0	0	0	95.7
G		1.4	0	0	0	0	0	0	0	0	0	0	0	0	7.1	0	91.5
H		1.4	0	0	0.7	0	0	0	0	0	0	0	0	0	0	0	97.9
I		2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.9
J		0.7	0	0	0	0	0	0	0	0	0	0	0	0	10	0	89.3
K		1.4	0	0	0	0	0	0	0	0	0	0	0	0	2.1	2.9	93.6
L		2.1	0	0	0.7	0	0	0	0	0	0	0	0	0	0	0	97.2
M		0.7	0	0	0	0	0	0	0	0	0	0	0	0	5.7	0	93.6
N		2.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97.1
Mean		1.79	0	0	0.33	0	0	0	0	0	0	0	0	0	2.33	0.19	95.4
VAR		0.82	0	0	0.88	0	0	0	0	0	0	0	0	0	12.2	0.56	7.81

Bibliography

Aggarwal, J.K., Davis, L.S., and Martin, W.N. (1981). Correspondence process in dynamic scene analysis. *Proc. of the IEEE*, 69(5):562-572.

Badler, N., Phillips, C., and Webber, B. (1993). Simulating Humans. *Oxford University Press*, Oxford.

Beek, P.J.L. van, Reinders, M.J.T., Sankur, B., and Van der Lubbe, J.C.A. (1992). Semantic segmentation of videophone image sequences. *Proc. of SPIE International Conference on Visual Communications and Image Processing*, pages 1182-1193.

Bharatkumar, A.G., Daigle, K.E., Pandy, M.G., Cai, Q., and Aggarwal, J.K. (1994). Lower limb kinematics of human walking with the medial axis transformation. *Proc. of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, pages 70-76, Austin, TX.

Bradski, G.R. (1998). Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal, 2nd Quarter*.

Cai, Q., Mitiche, A., and Aggarwal, J.K. (1996). Tracking human motion using multiple cameras. *Proc. of International Conference on Pattern Recognition*, pages 68-72, Vienna, Austria.

Chen, Z., and Lee, H. (1992). Knowledge-guided visual perception of 3-D human gait from a single image sequence. *IEEE Transactions on Systems, Man and Cybernetics*, 22(2):336-342.

Cheng, Y. (1995). Mean Shift, mode seeking and clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17:790-799.

Craw, I., Ellis, H., and Lishman, J.R. (1987). Automatic extraction of face-features. *Pattern Recognition Lett.*, pages 183-187.

Darrell, T., Gordon, G., Harville, M., and Woodfill, J. (1998). Integrated person tracking using stereo, color, and pattern detection. *IEEE Proc. of International Conference on Computer Vision and Pattern Recognition*.

Darrell, T., and Pentland, A. (1993). Space-time gestures. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 335-340, New York.

De Silva, L.C., Aizawa, K., and Hatori, M. (1995). Detection and tracking of facial features by using a facial feature model and deformable circular template. *IEICE Trans. Inform. Systems*, E78-D(9), pages 1195-1207.

Freund, Y., and Schapire, R. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory: Eurocolt '95*, pages 23-37.

Goddard, N.H. (1994). Incremental model-based discrimination of articulated movement from motion features. *Proc. of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, pages 89-95, Austin, TX.

Graf, H.P., Cosatto, E., Gibson, D., Petajan, E., and Kocheisen, M. (1996). Multi-modal system for locating heads and faces. *Proc. of 2nd International Confgerence on Automatic Face and Gesture Recognition*, pages 277-282.

Gunn, S.R., and Nixon, M.S. (1994). A dual active contour for head and boundary extraction. *IEEE Colloquium on Image Processing for Biometric Measurement*, London, pp 6/1.

Hogg, D. (1983). Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5-20.

Huang, C.L., and Chen, C.W. (1992). Human facial feature extraction for face interpretation and recognition. *Pattern Recognition* 25, pp. 1435-1444.

Hunke, M., and Waibel, A. (1994). Face locating and tracking for human-computer interaction. *28th Asilomar Conference on Signals, Systems and Computers*. Monterey, CA.

Jeng, S.H., Liao, H.Y.M., Han, C.C., Chern, M.Y., and Liu, Y.T. (1998). Facial feature detection using geometrical face model: An efficient approach. *Pattern Recognition*, 31.

Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14(2):201-211.

McKenna, S., Gong, S., and Collins, J.J. (1996). Face tracking and pose representation. *British Machine Vision Conference*, Edinburgh, Scotland.

Kjeldsen, R., and Kender, J. (1996). Toward the use of gesture in traditional user interfaces. *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 151-156, Killington.

Lam, K.M., and Yan, H. (1994). Facial feature location and extraction for computerised human face recognition. *International Symposium on information theory and its applications*. Sydney, Australia.

Lam, K.M., and Yan, H. (1996). Locating and extracting the eye in human face images. *Pattern Recognition* 29, pp. 771-779.

Leung, M.K., and Yang, Y.H. (1995). First sight: A human body outline labeling system. *IEEE Trans. on PAMI*, 17(4):359-377.

O'Rourke, J., and Badler, N. (1980). Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):522-536.

Polana, R., and Nelson, R. (1994). Low level recognition of human motion. *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 77-82, Austin, TX.

Poritz, A.B. (1988). Hidden Markov Models: A guided tour. *Proc. of IEEE International Conference on Acoustic Speech and Signal Processing*, pages 7-13.

Propp, M., and Samal, A. (1992). Artificial neural network architecture for human face detection. *Intelligent Eng. Systems Artificial Neural Networks*, 2:535-540.

Quek, F. (1995). Eyes in the interface. *Image and Vision Computing*, 13(6):511-525.

Rashid, R.F. (1980). Towards a system for the interpretation of moving light displays. *IEEE Trans. on PAMI*, 2(6):571-581.

Rohr, K. (1994). Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94-115.

Rossi, M., and Bozzoli, A. (1994). Tracking and counting people. *First International Conference on Image Processing*, pages 212-216, Austin, Texas.

Rowley, H.A., Baluja, S., and Kanade, T. (1998). Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 20:23-38.

Sakai, T., Nagao, M., and Kanade, T. (1972). Computer analysis and classification of photographs of human faces. *Proc. First USA-Japan Computer Conference*, p. 2-7.

Segen, J., and Pingali, S. (1996). A camera-based system for tracking people in real-time. *Proc. of International Conference on Pattern Recognition*, pages 63-67, Vienna, Austria.

Shio A., and Sklansky, J. (1991). Segmentation of people in motion. *IEEE Workshop on Visual Motion*, pages 325-332.

Sirovich, L., and Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America*, 4:519-524.

Takahashi, K., Seki, S., Kojima, H., and Oka, R. (1994). Recognition of dexterous manipulations from time-varying images. *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 23-28, Austin.
Turk, M., and Pentland, A. (1991). Eigenfaces for Recognition. *J. Cog. Neurosci.* 3:71-86.

Viola, P., and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*.

Wren, C., Azerbayejani, A., Darrel, T., and Pentland, A. (1995). Pfinder: Real-time tracking of the human body. *Proc. SPIE*, Bellingham, WA.

Yamamoto, J., Ohya, J., and Ishii, K. (1992). Recognizing human action in time-sequential images using Hidden Markov Models. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 379-385.

Yang, J., and Waibel, A. (1996). A real-time face tracker. *Proc. of the 3rd Workshop on Applications of Computer Vision*, Florida.