Name	:	Handoko Setyawan
Degree	:	Master of Science
Dept	:	School of Computing
Thesis title	:	Model-based Human Detection in Images

Abstract

This thesis proposes a model-based system for detecting human body in static cluttered scene images. We first build a distribution-based model of the objects by manually marking the human body countour in training images. This approach is capable of overcoming the difficulties in representing human body, a complex and non-rigid object, using traditional geometrical models. In order to localize human body in the images, we propose matching procedures based on edge information. We will also prove in this thesis that the system can extract human bodies of various poses from various cluttered backgrounds.

Since the model is created using a learning approach, the performance of the system depends heavily on the quality of the training examples. Fortunately one can improve the system performance, without having to rebuild the whole system, by feeding more examples during the learning process.

Keywords : model based object detection
human body detection
image localization
deformable templates
pedestrian detection

MODEL-BASED HUMAN DETECTION IN IMAGES

HANDOKO SETYAWAN

NATIONAL UNIVERSITY OF SINGAPORE

2001

MODEL-BASED HUMAN DETECTION IN IMAGES

HANDOKO SETYAWAN

(B.Sc.(Hons.), ITB)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2001

Acknowledgement

I wish to thank all the people for the help in completing this thesis.

I would like to thank Dr. Sung Kah Kay as my first supervisor. It was an honour to have a chance to work closely with such a person like him. I praised the time you spent with me to discuss my project and to help me whenever I faced problems. My only regret is that I did not have the opportunity to finish the work with him. I would also like to thank Dr. Leow Wee Kheng who supported me during the hard times and was willing to become my supervisor. This thesis would not be finished without his help. I thank my graduation committee and my advisory board for the time they spent in reviewing my work. I also thank Dr. Rudy Setiono, who introduced me to NUS, for giving me the opportunity to study in NUS.

To my labmates in CHIME/DIVE lab, thank you for providing me with wonderful and friendly environment during my research days. To Rini who has always been there for me, I could not find any words that could show how much I appreciated what she has done for me. To Fan Lixin who acted as my virtual supervisor, I really appreciated the time he spent to discuss my project. To Terence and Annie who introduced me to life in NUS and taught me many things during my early days. To Jack who spent the time with me in doing crazy stuffs together. Without him I would not notice the fun side of our lab. I would not forget how glad I was seeing Starcraft map on his monitor. My thank also goes to Tang Mengting, Long Huizhong, Li Rui, Indri, Lu Ping, and all the people that I could not mention here one by one.

Table of Contents

List of Figures

List of Tables

Summary

Object detection is one of the most common computer vision problems. Basically it requires us to differentiate our object of interest from the rest of the world. The goal of this thesis is to propose an automatic, robust method for detecting human bodies in static images. Some difficulties faced here are how to define a model that is able to characterize the object of interest (in our case, human body), that is robust enough to handle the variability of the object, and at the same time capable of differentiating human bodies from other objects. Formulating a good measurement function is also another challenge faced by this thesis.

Although there have been many object detection systems developed, we believe that existing systems are still far short of human abilities. While mostly working on rigid-body object, their performance also depends heavily on specific constraints of the object to be detected under special scene conditions. Thus it is not easy to implement them in real life applications.

Unlike rigid objects, human body has high variability in size, shape, pose, color and texture. It has higher variability than other non-rigid objects such as human face. For instance, frontal human faces have similar patterns in shape and spatial layout with limited color variation. On the other hand, the clothes worn and the various poses of human bodies (even if limited to the frontal view) produce a complicated combination of shapes, colors and patterns.

There are several previous research works done on human body detection. Most of the systems relied on motion information to eliminate clutters in the images [2–5,9,19–21,32,34–36], as well as to simplify the detection proccess. There is even one interesting research done in University of Berkeley which eliminates the difficulties of various patterns and colors produced from clothing by constraining their objects to be naked people [10]. Unlike those previous research, this thesis proposes a method of detecting human body in static images. Hence no motion information is available. Another method proposed by Papageorgiou [26] uses wavelets for human body detection in static images. Unlike their method, which is only able to localize the human figure, our method also tries to capture the pose of the human body.

Our method uses a statistical model to overcome the difficulties in representing human body as a complex and non-rigid object. The model is built by first manually marking a series of human body contours as training examples. Our choice of using 2-dimensional (2-D) model to detect human body is based on several experiments done by other scientists. It has been proven that human visual system tends of using 2-D representation of objects instead of using 3-D representation [30]. In order to localize human body from images, we propose a matching procedure which relies on edge information. Since the model is created by learning from examples, the performance of the system depends heavily on the quality of the training examples. Fortunately, the system's performance can be improved by feeding more examples during the learning process without having to rebuild the whole system. We also believe that our framework is general enough to be applied to other non-rigid objects.

In this thesis, we will discuss the details of our approach in modelling human body as well as the model matching method. We will also describe the experiments conducted to measure the system's performance. We will conclude the thesis with suggestions for future work.

Chapter 1

Introduction

1.1 Motivation

Among all of our basic senses, vision is probably the most important sense that we have. Eyes as our visual sensors enable us to explore the environment and gather information without having to have physical contact with the objects. One way to gather information is by identifying objects in the surrounding. Humans have the ability to recognize an object regardless of its shape and appearance from various viewpoints based on their prior knowledge of the object. An example of object detection would be human body detection. While it may look very easy for us to detect human body in its various shapes and poses, it is very difficult to design a machine with similar capability.

Human body detection has many possible applications. Automatic image indexing is an example where images are indexed based on its contents. Images can also be indexed based on the human bodies found in them. Other examples include pedestrian detection in mobile environment and surveillance system that triggers an alarm at the appearance of a person.

This thesis mainly focuses on building a system for detecting and localizing human bodies in static images. Furthermore, we also try to capture the shapes and poses of the human bodies detected by deforming a model to fit their contours.

Many existing object detection systems restrict the inputs to specific rigid-body objects, where there will be no shape transformation as well as unpredictable movements on the objects. By these simplifications, the systems perform well using hand-crafted fixed geometric model.

Unfortunately these limitations cannot be applied when we are dealing with human body which is a non-rigid object. Non-rigid object can undergo non-rigid shape transformation, hence cannot be adequately described by a fixed geometric model. To make things more complicated, an object can have many totally different physical appearances.

Another challenge in working with 3-D non-rigid objects in real life is that they will appear in unpredictable shapes and poses from different points of view and under different imaging conditions, especially after being projected onto a 2-D image plane. Under unrestricted environment, cluttered scenes will also become a problem. Even if we do not allow any major occlusion, many variety of image patterns occur in cluttered scenes. Furthermore, it is also possible that some non-human objects in the scene may have similar appearance as human figures.

Compared with other non-rigid objects such as human face, human body contains higher variabilities in size, shape, pose, color and texture. For instance, frontal human faces have similar patterns in shape and spatial layout with limited color variation. On the other hand, clothes worn and various poses of human bodies (even when limited to one point of view) produce a complicated combination of shape, colors and patterns. In this sense, we can conclude that human body detection is a more difficult task.

In order to perform human body detection well, we have to define a precise model that can represent most varieties and yet is capable of correctly rejecting similar looking non-human objects in images. Here a good matching method plays a major role in classifying human figures.

Figure 1.1: The human figure in the images vary in color, texture, pose and viewpoint. The images have different kinds of background.

Figure 1.1 highlights the difficulties of human body detection by showing several example images used in this project. One can see that the human figures in the images vary greatly in color, texture, pose and shape. One can also see that there is no consistency in the background of the images.

Instead of using traditional geometrical model to represent human body, we use a statistical model. To form our training database, we manually mark the countour of human bodies in various shapes and poses found in a set of images. We also develop a method which allows us to measure how good our model matches human bodies in the images.

Since the method uses a learning approach, the performance of the method depends heavily on the quality of the training examples. We see this as an advantage since it allows us to improve system performance without having to rebuild the overall system from scratch. The other advantage is that the system built is general enough and one can apply it to detect other non-rigid objects by re-training the system using different sets of training samples.

1.2 Problem Definition

To make this thesis easier to understand, first of all we want to highlight some definitions and constraints of the system proposed. The goal of this research is to propose a method for detecting human bodies in static images.

We divide the detection process into 2 major sub-tasks, detection and registration. Both processes utilize the human body templates defined in our training phase. The detection process will detect the appearance and location of human bodies in an image, while the registration process will refine the detection results by deforming the templates to fit the contours of the human figures found.

There are several constrains applied to our system:

1. The humans are standing and facing the camera or back to the camera. Although we do not allow other positions like sitting, crawling, and other possible poses, there is no constrain on the posture of each human body parts. As is shown in Figure 1.1, there are various hand, body and leg postures in the images.

2. There should be no other object in front of the human and the image should show the whole profile of the human. Our matching process uses edge information and therefore major occlusion will give misleading edge information to the system. Although we restrict the appearance of human body in images, there is no restriction in the condition of the scene. As will be shown later, our system is capable of detecting humans in cluttered scenes.

1.3 Related Work

There are several similar research works done in object detection, particulary in human body detection or sometimes more commonly referred to as pedestrian detection. This section describes some of them which we feel most related to what we have done, as well as compare them with our system.

In general, we can classify the existing systems based on the kind of images used in the system, namely static images and video sequences. Video sequences provide very important information compared to static images, which is motion information. Motion information can be used to help us simplify object detection tasks. In human body detection, for example, motion information can be used to boost system performance by implementing background substraction and optical flow analysis to overcome the cluttered scene problem, which is most likely to occur in real application. On the other hand, their systems will not be able to perform well on static images. Therefore in this sense, we can conclude that object detection in video sequences is simpler than detecting object in static images.

1.3.1 Human Body Detection in Video Sequences

Most of the existing human detection systems fall into this category, and most of them are mainly focused on using motion information to "avoid" cluttered background or/and applying simple shapes and background constrains to detect human figure appearances. Early systems usually also assume that a stationary camera is used to capture the sequences.

One example of the first human detection system built was the work done by Tsukiyama and Shirai in 1985 [35], which detected humans based on their leg movement. They defined a simple shape to represent leg movement against plain background, and measured the correspondence distance between moving regions in consecutive images. Their system was capable of handling multiple human figures in video sequences. However, there were several drawbacks in their proposed method which made their system not very practical in real life application. The first drawback was that the system will only work in a specific condition where a static camera and plain background were used to capture the sequence. The other drawback was that the system can only detect human figures with leg movement since it only relies on leg movement.

Among existing human body detection systems, some proposed methods not only localize but also label human body parts as well. In 1993, K. Rohr [32] developed a system based on cylindrical model of human body and kinematic motion data. The model was built by measuring real person's size and shape, and then modeled each major part as a combination of 3-D cylindrical shapes. The pose parameter was designed by learning from real-life sequences of walking human. By assuming that a static camera was used to capture the sequence, he eliminated the background scene using motion information. The model itself was fitted into the human figure using edge information of the images. While it was capable of labelling human figure parts in video sequences, the system could only detect human figures that move parallel to the image plane and was limited by the assumption that static camera was used to capture the sequences.

Leung and Yang [19, 20] also proposed an algorithm to label the outline of a human body contour. Unlike Rohr, they used 2-D template to represent human figure. A human figure was modeled as a combination of a set of ribbons (which represented major body parts) and a set of joint points (which determined connections among the ribbons). The system then analyzed the image's edge-map to determine the moving parts of the figure as well as to determine the joint points. Compared to other methods which usually have limitation on human poses, this approach was able to label a wide range of poses. While capable of detecting various human poses, this system was not generally applicable because some special conditions need to be fulfilled. It was assumed that there was only a single moving human in the image, captured using a stationary camera. It was also assumed that the intensity of the moving object was different from that of the background.

Wren et al. [36] proposed a different method of tracking a human body called Pfinder. Pfinder used a maximum a posteriori (MAP) approach to segment a human figure into a set of blobs corresponding to different regions of the body. The system modelled and segmented human body parts based on their local color information. Pfinder worked as follows: first, it predicted human appereance in an image with initial model parameters and calculated the likelihood that each predicted pixel was a member of the blob. Next, it resolved these pixel-by-pixel likelihoods into a support map and updated the statistical models of all blob models to fit the human figure. In order to differentiate human figure's color with other scene objects' colors, Pfinder assumed that a pre-defined background setting is used. Prior to the detection process, the system analyzed the scene color map. This information was used during the detection process to eliminate the background scene. Here, it was also assumed that a static camera was used to obtain the images.

Similar to Pfinder, Heisele's method [12, 13] also worked based on color and motion information of images. Initially the system clustered an image based on its color information and then a user manually chose which clusters belong to the human in the picture. By assuming that a fixed number of clusters appear in the sequence, the system detected human movement by re-computing the clusters' centroid in subsequent images. Unlike other methods mentioned above, this technique did not assume a stationary camera. However, the detection process needed user intervention to initially label human clusters in the image. Since it was assumed that there was a fixed number of clusters in the sequence, it followed that no changes were expected to appear in the background from its initial state.

Instead of only tracking and detecting, there were some methods proposed that also recognized the person detected. Lakany [18] proposed a very interesting method to recognize a person based on the way he walked. During the proccess, he attached lights to critical points of a human body such as joints points. The system then detected and learned the movement of each individual during the training proccess. While this method was not generally practical, it opened new possibilities of more applications based on human body detection.

Other research done by McKenna [21] combined human detection and face detection to perform recognition which was more applicable in real life. The method was capable of handling several moving humans in image sequences. However, like most existing systems, it assumed a fix camera and a controlled background scene, which did not change during the detection process. Human body appearance was detected solely using motion information. Since its recognition proccess was based on face detection, the performance relied heavily on the quality of the images taken.

Baumberg [2–5,9] proposed a method which builds a human prototype based on learning data. The approach built a model semi-automatically. First, he created a training data consisting of a set of human figures. These figures were obtained by removing background information from each image in the training data. He then ran snake algorithm to fit a curve into those human figure contours. After the process, he manually fixed some misalignment that were not recovered during the process. Using Pricipal Component Analysis, he reduced the dimensionality of the training data and determined the shape parameters of the model. This model was then used to detect the appearance of moving human bodies in the sequences. The system was capable of detecting multiple moving humans as well as capturing their contours. Recently, Ken Tabb [34] proposed a method with similar approach and added human parts detection capability in order to improve the performance of Baumberg's system. He used neural network not only to detect the human body based on the edges, but also to add a function to determine the "goodness" of the shape detected. However, both systems still relied on motion information to help them remove the cluttered background. Their systems also assumed that a static camera was used to capture the sequences.

1.3.2 Human Body Detection in Static Images

To the best of our knowledge, there are few human body detection systems that work on static images. However, there are prior works designed for detecting other objects, such as faces. This section will briefly describe some object detection systems which we feel are related to our work and will also describe some human body detection systems known to be able to work on static images.

The lack of motion information is one of the big drawbacks in working with static images. Since a cluttered scene cannot be simply avoided using background substraction and optical flow analysis methods, one have to come up with "good" model description and measurement function to differentiate human from the rest of the scene.

Template-based approach and image-invariant approach are the most common methods used in object detection [26] that work on static images. Template-based systems usually try to fit a prototype template of an object to the real object througout the image. Object location is then defined as the position where this prototype returned the highest match value. Image-invariant method, on the other hand, tries to match a set of image pattern relationships, which ideally could uniquely represent the object of interest. Unfortunately these ideal conditions is hardly found in real life applications.

Yullie et. al [37], in 1989, proposed a method which developed a geometrical model for object detection. Although the method worked well on rigid and semi-rigid objects like eyes and lips, it was not easily extandable to non-rigid objects. A much more enhanced method, proposed by Cootes et. al [7, 8], overcame this difficulty by building an example-based model. The approach taken was to learn from manually marked control points on the training data. He then ran statistical analysis on the data to form a face prototype and its shape parameter. His modelling method was commonly known as active shape method. To perform face detection, he used edge information of the image. By placing the template in an initial position, he began to deform the template based on the nearest edge pixel along the control points' normals, to fit the edge contour of a human face. There had been many systems developed using this algorithm to perform non-rigid object detection. Baumberg [2–5, 9], for instance, also implemented Cootes' method in building his human body template.

Sung [33] proposed an example-based general object and pattern detection system. He built a distribution-based model of the target objects pattern class to describe object's image appearances. The system then measured the distance between a pattern and the pattern prototypes to differentiate object patterns from non-object patterns. Implemented for a face detection system, this view-based approach could successfully handle frontal-view faces in the images. However, the pattern complexity of human body caused by the clothes worn made it difficult to classify human body based on pattern analysis. Therefore, this approach was not very suitable for human body detection.

Forsyth [10] proposed an interesting method of finding naked people in the image. The system detected a human appearance based on his skin color information. By constraining the human to be naked, he could avoid the pattern complexity caused by clothing. However as one could imagine, this method was not generally practical in our daily applications.

Papageorgiou [24–26] proposed a new approach of human body detection based on wavelet information. The system learned by observing object's wavelet representation in order to create a compact representation of object classes. The wavelet representation was derived from a statistical analysis of a dictionary of the class instances. Implementations of this method for face and human body detection had shown a reasonable degree of success in dealing with static images. To enhance system performance, video sequences were used to provide motion information. However, using wavelet representation, their approach could only detect human appearance without providing any pose and contour information.

Recently, based on a similar approach, Mohan [22] proposed a new method to detect human appearances based on the number of its components found in images. Since the method did not look for full human body profile, it was capable of detecting partially occluded human in static images. As relative location and size of components to be detected had to be manually provided, its performance relied heavily on the system designer's knowledge of the object. Moreover, it only provided the location of each component without any pose or contour information.

1.4 Our Approach

While all the systems mentioned in the previous section have succeeded to varying degrees, we can conclude that most of them rely on one or all of these following restrictions:

1. Stationary camera must be used to produce static background.
2. Images must be taken in a controlled environment with a fixed background.
3. Initial position must be defined prior to the detection proccess.
4. Motion information is required to remove cluttered background and to differentiate an object from other objects.

The proposed solution aims at building a more general human body detection sytem that overcome these limitations. Our system is able to work without motion information and does not have any restriction on clothes worn, pose, shape, and condition in which the images are taken. Furthermore, our system also provides pose and contour information of the human figures detected.

Our approach can be divided into 2 parts. In the first part we focus on developing a prototype to represent human shapes and poses. This prototype will be used later during the matching proccess as a basis for locating human figures in the images. The second part focuses on developing an algorithm for the matching proccess itself. Here we propose an edge based function to measure how good the prototype matches the real objects in images. During the matching proccess, we also deform the prototype to refine the detection result by fitting it to human figures' contours. Locations and poses of human figures found are determined by the positions and poses of the templates that give the highest goodness values.

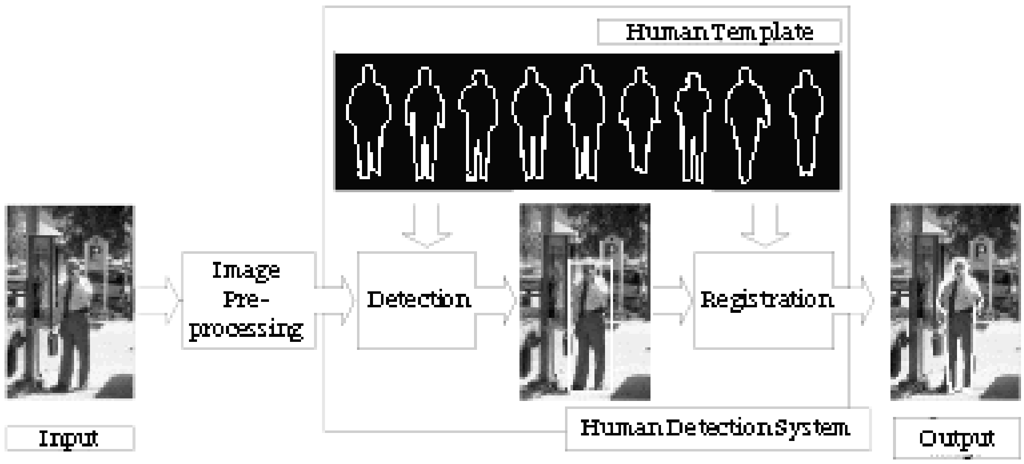


Figure 1.2: The picture above illustrates our human body detection system architecture. There are two major sub-tasks performed, namely detection and registration. The first process detects the appearance and location of human body in the image, which is represented as white boxes surrounding the human figure. Registration process refines the detection result and deforms the model to fit the human figure.

1.4.1 Modelling Algorithm

In order to overcome the human figure's high variability of shapes and poses, we choose to use an example-based model. This method, commonly known as active shape model, derives shapes and poses of manually drawn templates of human figure found in our training database. By doing statistical analysis on the data, we then define several so-called *basic* templates to represent most common poses found. Each template has its own *shape parameter vector* which is used to control the deformation of its shape and pose. Some examples of basic templates used in our system can be found in Figure 1.3.

One advantage of this method compared to traditional geometric-based method is that the templates derived using statistical analysis are not biased by the designer's knowledge of the object. The other advantage is that the system's performance can be improved by simply adding more training data. This approach also allows us to apply our system to detect other non-rigid objects without having to rebuild the whole detection system from scratch, which in this sense makes it more general.

1.4.2 Matching Algorithm

Our detection function is based on edge information gathered using Canny edge detection method [6]. We divide the matching process into two major subtasks, namely detection and registration.

In the detection stage, using previously defined human figure templates, we perform an exhaustive search to measure the possibilities of human existence in each possible location of the image. This stage will provide information on the number of human bodies detected as well as their exact location and size in the image. We then define these parameters as our initial guess for the registration process.

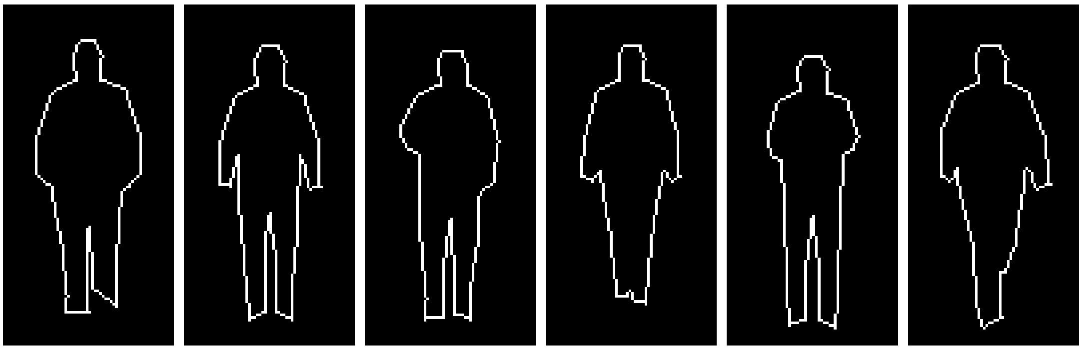


Figure 1.3: Example of some basic templates derived from training data.

Based on the parameters given at each position, we deform the human figure prototype to fit the human figure contour. Here we perform an exhaustive search to find the best combination of template shape parameters that produces the highest match value.

Both of the processes are illustrated in Figure 1.2. The white box surrounding the human figure in the middle image represents the detection result and the white deformed template that fits the human figure contour on the right most image represents the registration result.

1.5 Thesis Outline

The remainder of the thesis is organized as follows. Chapter 2 describes our modelling method in detail as well as the training and clustering processes used in our system. Chapter 3 discusses more about the matching method used in the detection and registration phase. The experiments conducted to analyze system performance is described in Chapter 4. Also in Chapter 4, we give detailed explanation of how we measured the system's performance, as well as discuss the experimental results. We then conclude this thesis by summarizing the project, followed by suggestions for further research.

Chapter 2

Building a Contour Model

Cootes' "Point Distribution Model" [7, 8] has proven to be a good mechanism for representing non-rigid object, such as human face and human body. Therefore, in this thesis we use a similar approach to build our model to represent a human figure. The approach taken can be described as follows. First, we manually place control points on the human figure contour to obtain a representation vector of each training sample. The distribution of all sample representation vectors are then used to model human figure by fitting the sample distribution with a single multi-dimensional Gaussian cluster, consisting of one centroid location and one full covariance matrix [27–29]. Due to the high variabilities in shapes and poses that it has, a single multi-dimensional Gaussian cluster is inadequate for representing all the variants of human figure. Therefore in this thesis, we model human figure by fitting the sample distribution with multiple multi-dimensional Gaussian clusters. We then run principle component analysis (PCA) on each cluster to choose a number of most significant components of the cluster.

This chapter provides detailed explanations of our method in building sample representation vectors, clustering sample distribution, and building model representation of human figure. This chapter will also discuss our method in deforming the models for the registration process.

2.1 Sample Representation

In order to represent various shapes and appereances of the human figure, we build a set of statistical templates based on the distribution of manually marked training data. The steps involved in obtaining a representation vector for each training sample can be briefly described as follows:

1. A set of control points are manually placed on pre-defined positions along the human figure contour to represent the shape and the appearance of the human body found in that particular sample image.

2. Each set of control points is stored in the form of a vector $\mathbf{m}_e$ with $2n$ components, where n represents the number of control points in each set (in our case, n is 33) and $e = 1, 2, \ldots, N_e$ represents the sample image index (N_e is equal to the number of sample images).

3. Each vector $\mathbf{m}_e$ is then normalized to form a unit vector $\vec{\mathbf{m}}_e$. This step is necessary because human figures in sample images have different size and shape.

The template, as mentioned previously, is a set of 2D points along the human body contour. In our case, our template consists of 33 control points.

Figure 2.1 shows how we label control points distribution along human figure. Each control point corresponds to a specific part of the human figure. By maintaining the position of each control point, we are able to tell which part is being represented by which set of control points.

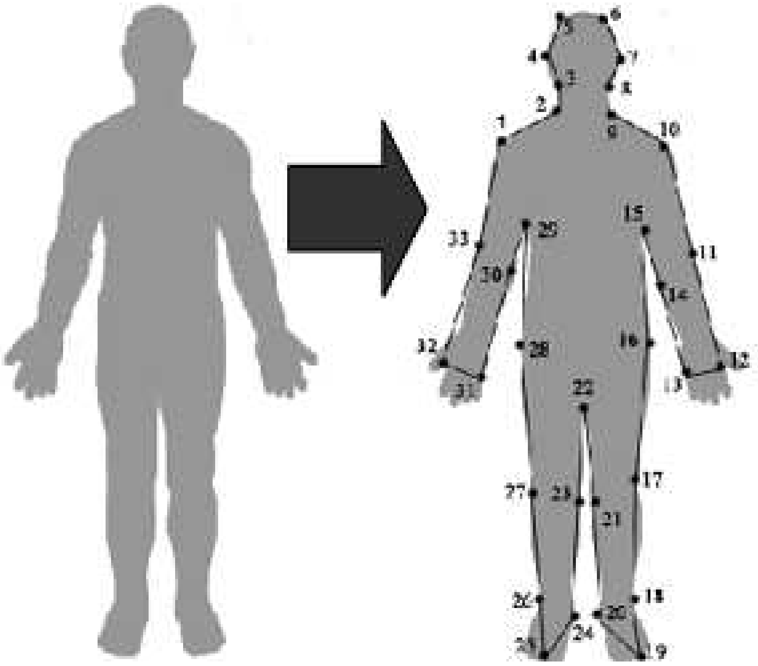


Figure 2.1: The image above shows control points distribution along the human figure contour. Each control point corresponds to a particular part of the human body. Most of them correspond to the joints of human body part such as the ankle, knee and elbow.

In order to build a good model, we have to carefully choose where to put the control points in the training data. Here, we choose to put the control points on the joints of the human body parts such as the ankle, knee, and elbow. As an exception because of its special shape, the head is represented by several control points.

The template shown in Figure 2.1 does not precisely represent the details of the human figure, but it is good enough for our application. Furthermore, the resolution of the images used in our project does not provide detailed information of each part. For applications that need more details, we can always include more control points. Bear in mind that the number of control points used is also significant in determining the number of training data needed. The more control points involved, the more training data will be needed to analyze the distribution.

To give a better understanding of our approach, Figure 2.2 illustrates how we label some training images. The pictures in the first column show images used as our training data, while the second and third show marked images and the associated templates of each image respectively. The control points are placed manually on the contours of the templates.

As mentioned in the beginning of this section, the human figure template from each image sample is stored in the form of a vector $\mathbf{m}_e$ that consists of $2n$ components. By maintaining an ordered coordinate list, we are able to define the direction of each template's splines. Let the control point coordinate along the figure be labeled as follows: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where the index refers to a point order in Figure 2.1. The vector $\mathbf{m}_e$ is then defined as follows:

$$\mathbf{m}_e = (x_1, \ldots, x_n, y_1, \ldots, y_n)^T$$

To build a model, we perform statistical analysis of the training data. However, before we can perform the data analysis, we have to make sure that each sample is not position dependent. We also have to make sure that they lie in the same coordinate frame. Therefore, prior to data analysis, we perform an alignment process on each training set to form a unit vector $\vec{\mathbf{m}}_e$.

Here, we implement a very simple alignment algorithm as follows:

1. Calculate the centre of gravity of each sample.

2. Shift each sample so that its centre of gravity is at the origin.

3. Scale each sample into a unit vector:

$$\vec{\mathbf{m}}_e = \frac{\mathbf{m}_e}{|\mathbf{m}_e|}$$

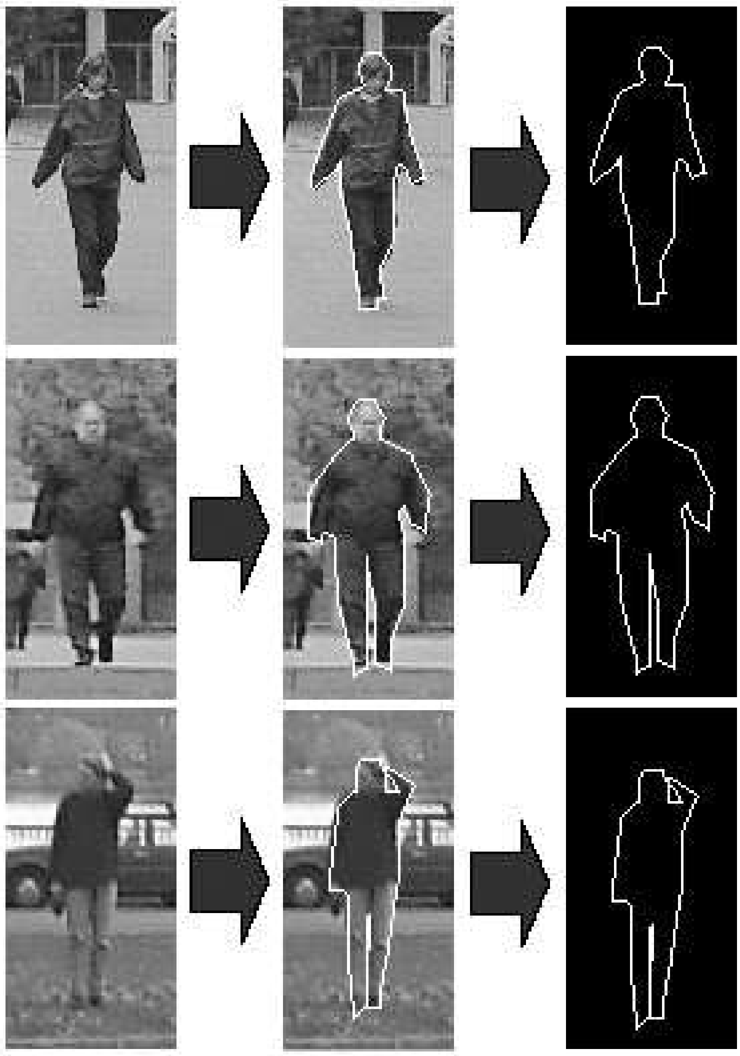


Figure 2.2: The images above show examples of training data used in this project. Images in the first column are examples of input images used to form our training data. Images in the second and third column are manually marked images and the associated templates.

Our training database consists of 210 manually marked images. All data are gathered using images with 64x128 pixels. In order to increase the number of data, we also include a mirror set of each sample. Thus we have a total of 420 training templates. The set of 420 samples form a distribution in $2n$-dimensional space. Our model is then built based on this data distribution.

2.2 Clustering and Model Representation

Our approach in representing human figure based on data distribution of training sample shares a similar idea with Pentland's "view-based eigen space" approach [27–29]. The key idea of the approach is to fit the sample data distribution with a multi-dimensional Gaussian cluster, consisting of cluster centroid and full covariance matrix of the cluster. However, we believe that a single Gaussian multi-dimensional cluster cannot provide a good model if the actual distribution is not unimodal. The human shape distribution can be too complex to be represented using single multi-dimensional Gaussian cluster. Therefore, to solve this problem, instead of assuming the distribution as one big cluster, we divide the distribution into several small locally linear clusters. Each prototype cluster is a multi-dimensional Gaussian with a centroid location and a covariance matrix that describes the local data distribution. In this sense, one can view these human figure clusters as a coarse distribution-based representation of the human figure manifold. Figure 2.3 shows our hypothetical example of how multiple multi-dimensional Gaussian clusters can represent the distribution better if the actual distribution is not unimodal.

To cluster the sample data distribution, we run K-means clustering algorithm which can be described as follows:

1. Define k initial cluster centers. Divide the data set into k clusters by assigning each data sample to the nearest cluster. The distance between a sample and a cluster is defined as the Eucledian distance between the sample and the cluster centroid.

2. Initialize the covariance matrices of all k clusters to be the identity matrix.

3. Compute the cluster centers as the centroid of the samples in the cluster.

4. With the current k cluster centroids and the associated cluster covariance matrices, re-compute data partition by re-assigning each data sample to the nearest cluster. If data partitions remain unchanged, or the number of

iteration performed has exceeded a maximum number allowed, proceed to Step 5. Otherwise return to Step 3.

5. Return current set of k cluster centroids $\bar{\mathbf{m}}_c$ and their cluster covariance matrices $\mathbf{S}_c$, where $\bar{\mathbf{m}}_c$ and $\mathbf{S}_c$ are defined as follows:

$$\bar{\mathbf{m}}_c = \frac{1}{N_c} \sum_{e=1}^{N_c} \vec{\mathbf{m}}_e \tag{2.1}$$

$$\mathbf{S}_c = \frac{1}{N_c - 1} \sum_{e=1}^{N_c} (\vec{\mathbf{m}}_e - \bar{\mathbf{m}}_c) (\vec{\mathbf{m}}_e - \bar{\mathbf{m}}_c)^T \tag{2.2}$$

and N_c represents the number of unit vectors $\vec{\mathbf{m}}_e$ in cluster c.

In order to obtain the eigenvectors of each cluster, we perform Principal Component Analysis (PCA) on each cluster individually. The algorithm is described as follows:

1. Compute all the eigenvectors $\mathbf{e}_i$ with corresponding eigenvalues λ_i of $\mathbf{S}_c$ (sorted so that $\lambda_i \geq \lambda_{i+1}$).

2. Each eigenvalue gives the variance of the data about the mean in the direction of the corresponding eigenvector. Compute the total variance as follows:

$$V_T = \sum_t \lambda_i$$

3. Choose the first t largest eigenvalues such that

$$\sum_{i=1}^{t} \lambda_i \geq f_v V_T$$

where f_v represents the proportion of total variance we wish to account for from each cluster c (in our case, 95 percents).

4. Store the choosen eigenvectors in matrix $\mathbf{E}_c$, such that

$$\mathbf{E}_c = (\mathbf{e}_1 | \mathbf{e}_2 | \ldots | \mathbf{e}_t)$$

where t indicates the number of eigenvectors choosen for cluster c. Since the size of each Gaussian cluster can be different, the size of t of each cluster can also be different.

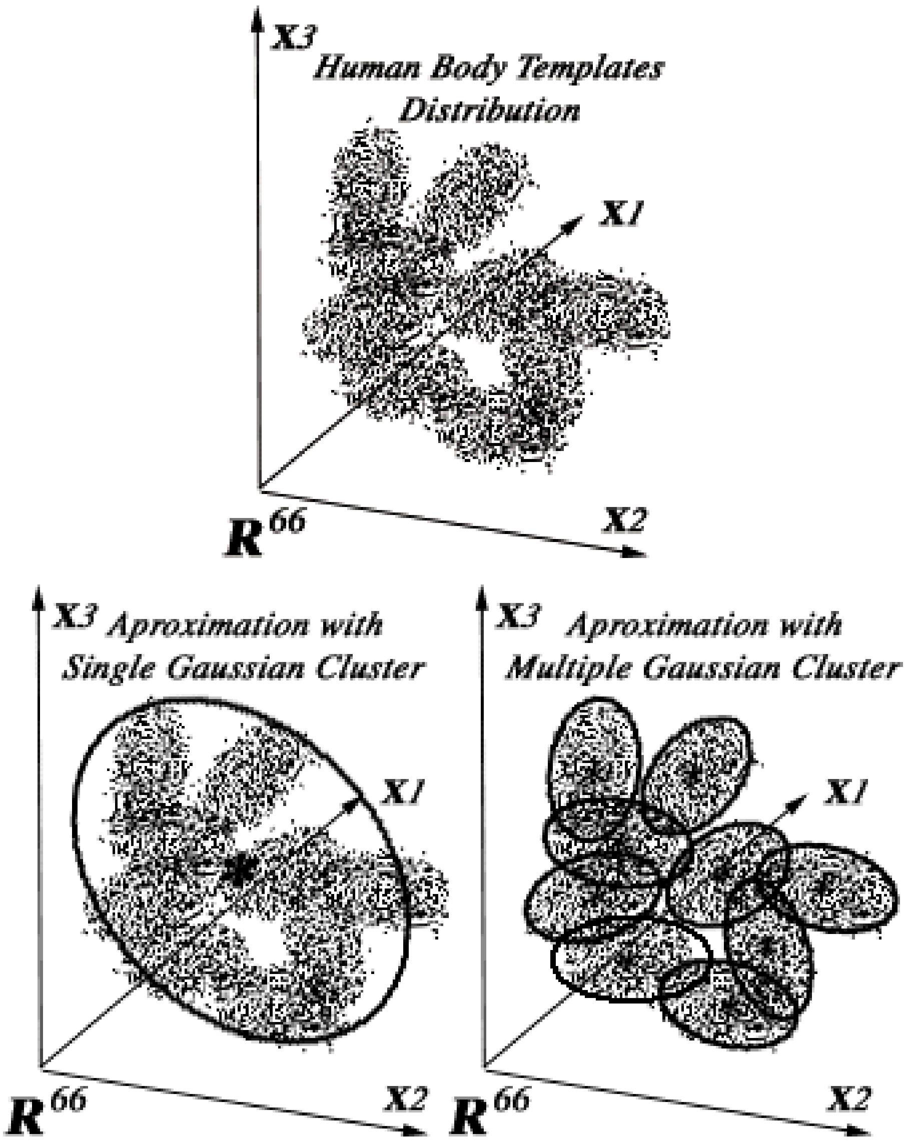


Figure 2.3: The images above illustrate a hypothetical plot of human body shape distribution found in our database. The two images in the second row give graphical interpretations of how we can fit all sample data into a single multi-dimensional Gaussian cluster or into multiple multi-dimensional Gaussian clusters. The ellipses in both pictures illustrate multi- dimensional Gaussian clusters used, and star sign in the middle of each ellipse denotes the centroid of each cluster. Here we can conclude that multiple multi-dimensional Gaussian cluster can represent the distribution better if the actual distribution is not unimodal.

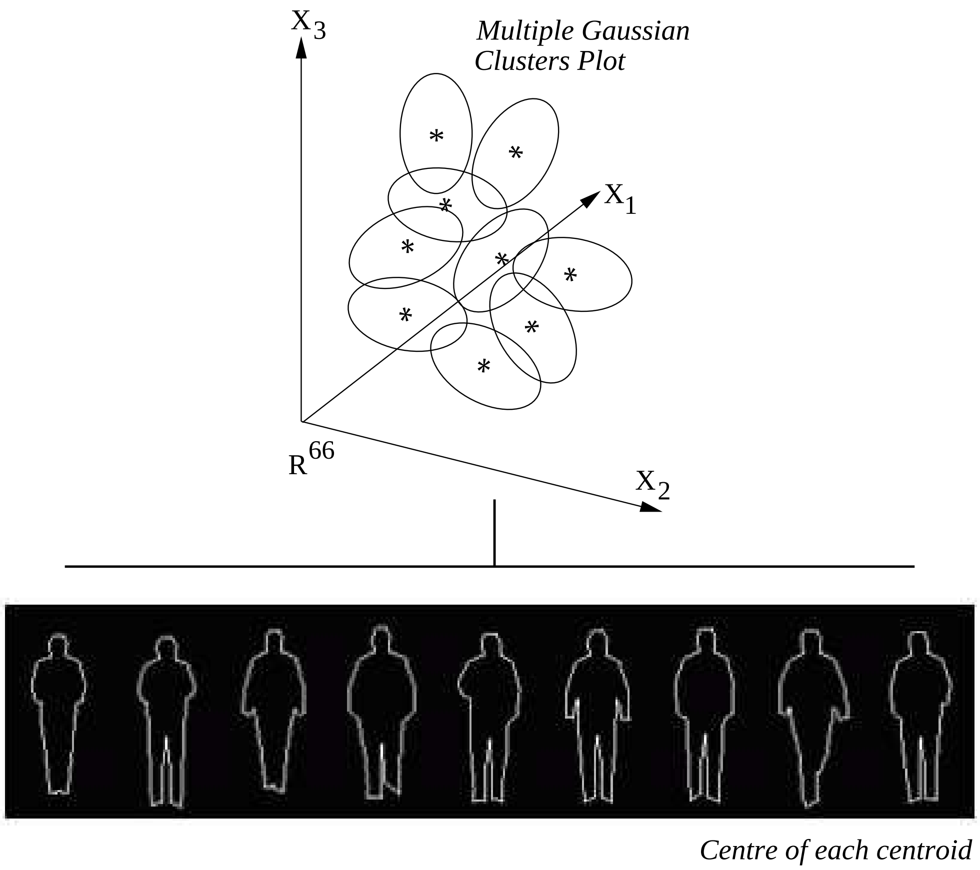


Figure 2.4: The image above illustrates a representation of the data sample distribution using 9 multi-dimensional Gaussian clusters. Human figure plots on the bottom of the image represent the centroids $\bar{\mathbf{m}}_c$ in the clusters c.

We conducted an experiment to see the effect of varying the number of cluster used on the system's performance. The experiment was conducted by trying to fit each set of cluster centroids into 300 human figure images using our detection algorithm. Based on the result, we concluded that our system performs well using 9 multi-dimensional Gaussian clusters. Increasing the number of cluster beyond 9 did not give us any significant improvement of performance. The detailed explanation of the detection algorithm used in this experiment can be found in Chapter 3.

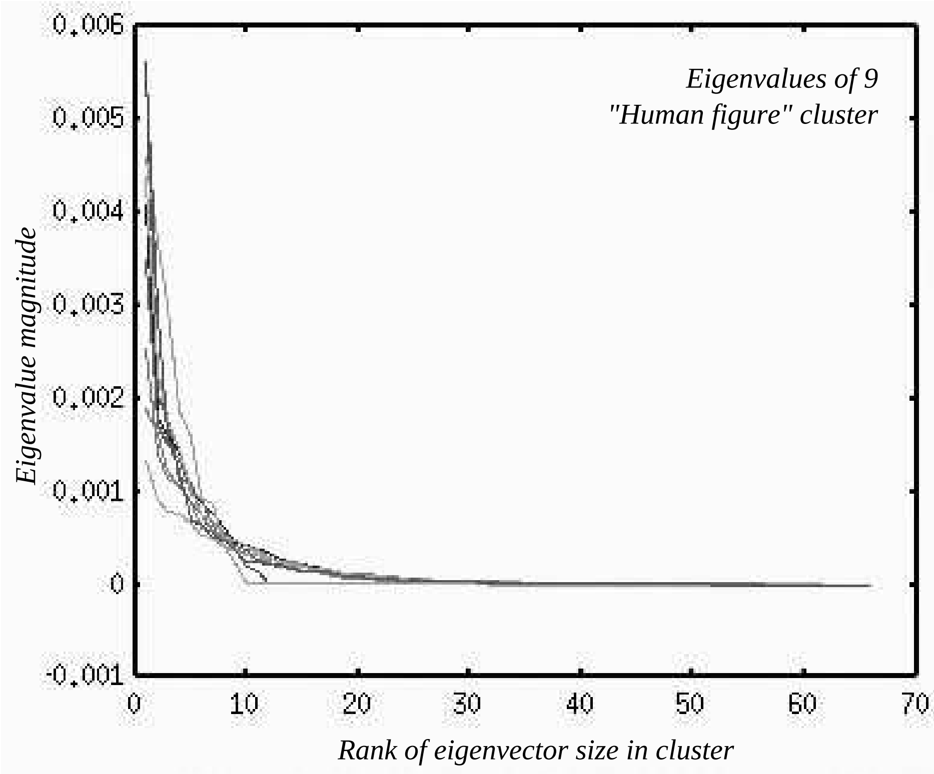


Figure 2.5: A plot of the eigenvalues of our 9 "human figure" clusters in order of the rank of eigenvector size in the associated cluster. Notice that for all 9 curves, only a small number of eigenvectors have significant eigenvalues. From the plot we can also conclude that the number of "significant" eigenvectors of the clusters are different.

Figure 2.5 plots the eigenvalues of our 9 "human figure" clusters decreasing in order of magnitude. As we see for all eigenvalue curves, only a small member of the eigenvectors have significant eigenvalues. As stated previously, from each cluster we choose the t largest eigenvalues such that the sum of all choosen eigenvalues is at least 95 percent of the sum of all eigenvalues in that particular cluster. Table 2.1 shows the number of significant eigenvectors of each cluster derived using this criterion.

Table 2.1: The table below shows the number of significant eigenvectors of each human figure cluster. Notice that each cluster has a different number of significant eigenvectors.

Cluster	Number of significant eigenvectors
1	9
2	20
3	19
4	24
5	29
6	22
7	19
8	7
9	18

2.3 Model Transformation

During detection and registration process, the models are transformed to fit the contour of human figures in the images. This section describes all the transformation processes performed in our system.

Basically there are three transformation processes performed:

1. **Translation**. During the detection and the registration processes, a model is moved by shifting its center of gravity in the x-direction and y-direction.
2. **Scaling**. During the detection and registration processes, a model is scaled by multiplying its unit vector with a constant.
3. **Deformation**. A model is deformed by varying its shape parameter vector $\mathbf{b}$. The deformation process is only performed during the model registration process.

The overall model transformation process is formulated as follows:

$$\mathbf{m}_c^* = T_{u,v,s}\left(D_{\mathbf{b}}\left(\bar{\mathbf{m}}_c\right)\right) \tag{2.3}$$

where $\mathbf{m}_c^*$ represents a transformed model and $\bar{\mathbf{m}}_c$ represents a cluster centroid, $D_{\mathbf{b}}(\bar{\mathbf{m}}_c)$ denotes the deformation function and $T_{u,v,s}(\mathbf{m}_c)$ denotes translation and scaling function.

2.3.1 Deformation

A model is deformed by adding some amounts of deformation on the cluster centroids $\bar{\mathbf{m}}_c$. Refer to Figure 2.3, we can see that each shape variation of the model is represented by one point in our "human figure" eigenspace. Determining the centroid of the cluster in the eigenspace $\bar{\mathbf{m}}_c$, we can represent shape variation of the model as a linear combination of the cluster's eigenvector.

$$D_{\mathbf{b}}(\bar{\mathbf{m}}_c) = \bar{\mathbf{m}}_c + \mathbf{E}_c\mathbf{b} \tag{2.4}$$

where $\mathbf{E}_c = (\mathbf{e}_1|\mathbf{e}_2|\ldots|\mathbf{e}_t)$ represents the eigenvector matrix consisting of t largest eigenvector $\mathbf{e}_i$ of cluster c and $\mathbf{b} = (b_1, b_2, \ldots, b_t)$, is a t by 1 vector, representing the cluster's shape parameter. Here b_i represents a weighting factor along eigenvector $\mathbf{e}_i$. Let $\bar{\mathbf{m}}_{cj}$ to be a control point of cluster centroid $\bar{\mathbf{m}}_c$, we can then perform a deformation process $D_{\mathbf{b}}$ on each point as follows:

$$\mathbf{m}_{cj} = \bar{\mathbf{m}}_{cj} + \sum_i e_{ij} b_i \tag{2.5}$$

Figure 2.6 illustrates the effect of varying each of first three shape parameters in turn between $\pm 1.5\sqrt{\lambda_i}$ and leaving all other parameters unchanged.

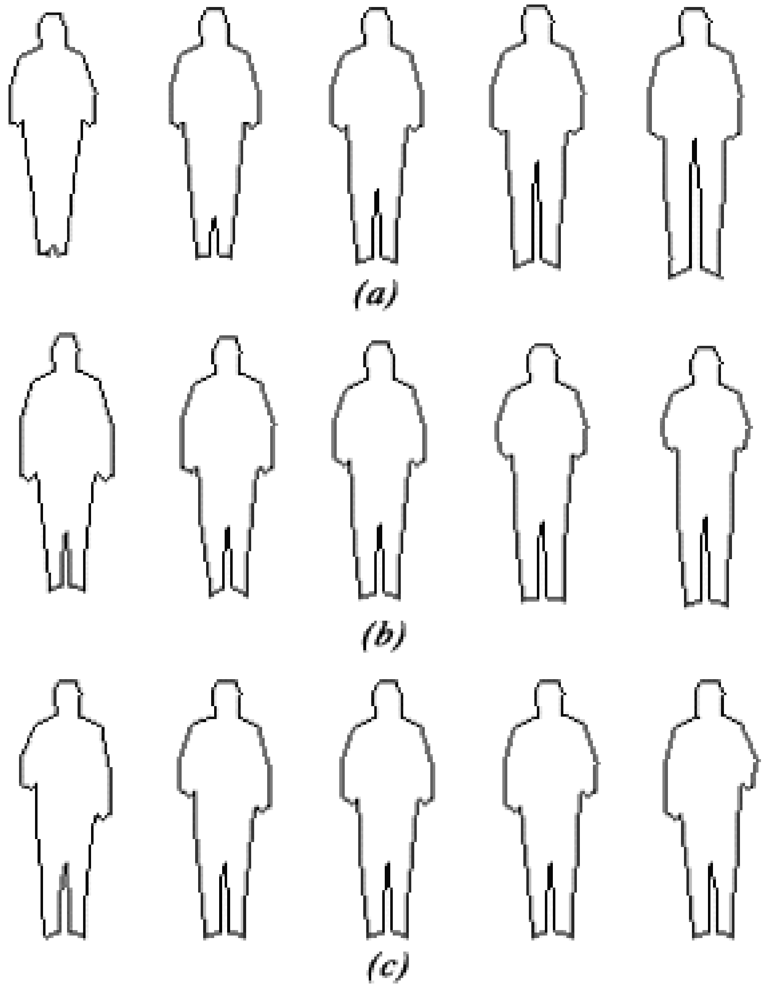


Figure 2.6: The above images illustrate the effect of varying each of first three deformation parameters in turn between $\pm 1.5\sqrt{\lambda_i}$ and leaving all other parameters unchanged. The midle image of each row is the original human figure shape, while the first, the second, the fourth and the fifth are deformed shapes by varying shape parameter in turn between $\pm 1.5\sqrt{\lambda_i}$. **(a)** illustrate the effect of varying the first shape parameter, **(b)** illustrate the effect of varying the second shape parameter and **(c)** illustrate the effect of varying the third shape parameter.

2.3.2 Translation and Scaling

Let a point (x, y) be a control point of a deformed shape, we can then perform a translation and scaling process $T_{u,v,s}$ for each points as follows:

$$T_{u,v,s}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + s\begin{pmatrix} x \\ y \end{pmatrix} \tag{2.6}$$

where the control point (x, y) is translated by u pixels and v pixels in the x- and y-directions respectively, and is scaled by scaling factor s.

More details about the implementation of both $T_{u,v,s}$ and $D_{\mathbf{b}}(\bar{\mathbf{m}}_c)$ processes in our system are discussed in Chapter 3.

2.4 Summary

Chapter 2 provides detailed explanations of our approach in building 9 multi-dimensional Gaussian clusters from sample data distribution, as well as the method used in obtaining the sample data distribution itself from training images. This chapter also describes the method used in choosing the most significant eigenvectors of each cluster and the transformation formula used to perform translation, scaling and deformation of the model.

The cluster centroids derived from the 9 clusters are used as our basic human figure models. These models are then used to perform both model detection and model registration. The eigenvector matrices derived from the 9 clusters are used to perform deformation on the basic models, which is used solely in model registration.

Chapter 3

Detection and Registration

This chapter describes how we utilize the models discussed in Chapter 2 on our human body detection system. Referring to Figure 1.2, basically there are three processes involved in our system.

1. **Edge Detection**. This process performs noise reduction and edge detection of the input image. The outputs of this process are the image gradient-map and the image edge-map. These outputs will be used in the next process, namely, human figure detection process and human figure registration process.

2. **Human Figure Detection**. The goals of human figure detection are to detect the appearance of human figures in the image and to provide an initial guess of the position and the size of the figure. These initial guesses are the inputs to the next process, human figure registration.

3. **Human Figure Registration**. Based on the output of the previous process, human figure registration refines these initial guesses by deforming the models to fit the human figure contours in the image. The output of this process are deformed models that are similar in appearance with the human figures detected in the image. In this process the models undergo translation, scaling and deformation.

The rest of this chapter will discuss the above processes in detail, including the methods and the measurement functions used.

3.1 Edge Detection

There are two processes involved in this phase, namely noise reduction and edge detection. The method used for noise reduction in our project is by filtering the input images by a median filter. Instead of changing pixel intensity according to the average local neighbourhood intensity, which tends to blur sharp discontinuities in intensity values, a median filter [17] replaces it with the median of the local neighbourhood intensity values. Since the new values are not significantly different from their neighbourhood, this method is very effective in removing small noises while retaining image details.

The process can be described as follows:

1. Apply a 3x3 window in the image to compute new intensity of the pixel at the center of the window.

2. Sort the pixels in the window into increasing order of intensity.

3. Select the median value from the list as the new value of the pixel in the center of the window.

As mentioned previously in the beginning of this chapter, edge information of the input image is very important because it is used as a basis of our detection and registration process. The algorithm used to gather edge information in our system is the Canny edge detector [6]. Canny edge detector is a well known edge detector which works on the first derivative of a Gaussian and closely approximates the operator that optimizes the product of signal-to-noise ratio and localization. The detector provides information on the image gradient map as well as the image edge map.

The method used is described as follows:

1. Apply a 1-dimension Gaussian filter to image I, in the x- and y-direction to smoothen the image. I_x and I_y are the result of the convolution ($*$) between the image and the Gaussian filter in each direction.

$$\begin{aligned} G(x) &= \frac{1}{2\pi}\exp(\frac{x^2}{2\sigma^2}) \qquad I_x = I * G(x) \\ G(y) &= \frac{1}{2\pi}\exp(\frac{y^2}{2\sigma^2}) \qquad I_y = I * G(y) \end{aligned} \tag{3.1}$$

2. Compute the gradient magnitude and orientation using Gaussian first derivative on each direction. ∇I_x and ∇I_y are the result of the convolution ($*$) between I_x and I_y and the first derivative of the Gaussian filter in each direction. $\parallel \nabla I(i,j) \parallel$ and $\theta(i,j)$ are the gradient magnitude and gradient orientation of the pixel (i,j) in the image.

$$\begin{aligned} \frac{dG(x)}{dx} &= \frac{-x}{2\pi\sigma^2}\exp(\frac{-x^2}{2\sigma^2}) \qquad \nabla I_x = I_x * \frac{dG(x)}{dx} \\ \frac{dG(y)}{dy} &= \frac{-y}{2\pi\sigma^2}\exp(\frac{-y^2}{2\sigma^2}) \qquad \nabla I_y = I_y * \frac{dG(y)}{dy} \end{aligned} \tag{3.2}$$

$$\begin{aligned} \parallel \nabla I(i,j) \parallel &= \sqrt{\nabla I_x^2(i,j) + \nabla I_y^2(i,j)} \\ \theta(i,j) &= \arctan(\frac{\nabla I_y(i,j)}{\nabla I_x(i,j)}) \end{aligned} \tag{3.3}$$

3. Apply nonmaxima suppression to the gradient magnitude. Nonmaxima suppression process nullifies the gradient value at a pixel if its gradient magnitude is not a local maximum among the neighbour along the perpendicular to the pixel's gradient orientation. The result is a thinned edge map, where only the dominant edges are present.

4. Use the double thresholding algorithm to detect and link edges. This method involves two different threshold values which varies from picture to picture. The low threshold value is chosen so that the number of pixels having intensity lower than that value is 5 percent of the total number of image pixels. A

similar but opposite criterion is applied to choose the higher threshold value. Any pixel in an edge list that has a gradient greater than the higher threshold value are classified as a valid edge point. Any pixels connected to these valid edge points that have a gradient value above the lower threshold value are also classified as edge points. That is, once you have started an edge you don't stop it until the gradient of the edge has dropped considerably. This process will produce an edge map Z of the image.

Figure 3.1 shows the results of this edge detection process. The image's gradient information are stored into two maps, gradient map in x-direction ∇I_x and y-direction ∇I_y. Both maps are used to compute gradient magnitude $\| \nabla I\,(i,j) \|$ and gradient orientation $\theta\,(i,j)$ during the detection and registration process.

3.2 Human Figure Detection

The goals of the human figure detection process are to detect the appearance of human figures in the images and to provide initial guesses of the locations and sizes of the human figures. The idea here is to place a model on top of the image, and measure how likely the image underneath the model is a human. Therefore we can detect the appearance of human figures in the image by finding the location which gives a high degree of match. For this purpose, we introduce two match measurements namely edge direction match and edge point match.

Figure 3.1: Edge Detection results: (a) Input image, (b) smoothened image, (c) the image's gradient magnitude map and (d) the image's edge map.

Edge direction match measures image gradient magnitude and orientation along the model contour. This function assumes that the image gradient magnitude along the model contour is high if the model is placed on top of a human figure that has a similar appearance. This function also assumes that the image gradient orientation along the contour is perpendicular to the orientation of the model contour whenever the model is perfectly placed on top of the human figure with a similar shape.

Edge point match measures the percentage of the model contour points that have edge pixels in their neighbourhoods. This function assumes that there are K members of the model contour points that will have at least one nearby edge pixel in their neighbourhoods, if the model is placed on top of the human figure with a similar shape.

Referring to Figure 3.1c and 3.1d, we can see why both the above functions are needed and how they complement each other. The image gradient map, which is needed by edge direction matching, contains a lot of noise. Therefore, the detection process based solely on edge direction matching will give us a lot of false alarms. The edge map, on the other hand, provides too litle information about the human figure. With this condition, the detection process based solely on edge point matching will not perform well either. By combining the two, we expect that the edge point matching function will eliminate false alarms caused by noise in edge direction matching.

3.2.1 Edge Direction Matching

Like all other edge-based detection methods, we expect edge-pixels along human figure contour to have high gradient magnitude compared to other parts of the image. Therefore, we design the edge direction matching function to measure the sum of the gradient pixels' magnitude underneath the model. This function will attain its maximum value when the model falls right on top of the high gradient pixels.

Unfortunately, without any additional information, it is not possible for the function to differentiate which pixels belong to human contour and which ones belong to other objects. In a noisy image, which is usually the case if the image is taken in real life environment, this function can give us a lot of false detections.

To overcome the problem, we improve the function to also compute the gradient pixels' orientation along the model. We assume that the orientations of the gradient pixels along the contour are perpendicular to the direction of the contour. The new function will attain its maximum value when the model falls right on top of the high gradient pixels that also have perpendicular orientation to the model's contour.

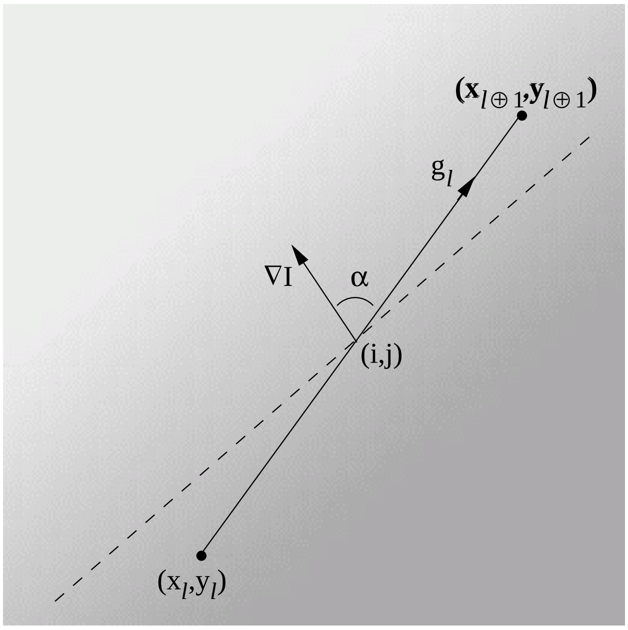


Figure 3.2: The image above illustrates the edge direction matching process. The solid line represents l-th segment of the model $\bar{\mathbf{m}}_c(u, v, s)$ and the dash line represents the edge in the image gradient map. In the picture, $\mathbf{g}_l$, ∇I, and α represent the model segment's unit vector, the gradient pixel's unit vector, and the angle between these two vectors. Edge direction matching function reachs its maximum value if the magnitude of the gradient pixel $\| \nabla I(i, j) \|$ is maximum and α is equal to 90 degrees.

Consider the model $\bar{\mathbf{m}}_c(u, v, s)$ that consists of n (which in our case is equal to 33) segments $\mathbf{g}_l$ and n pairs of control points (x, y), where

$$\bar{\mathbf{m}}_c(u, v, s) = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n\}$$

and l-th segment is given by

$$\mathbf{g}_l = ((x_l, y_l), (x_{l\oplus 1}, y_{l\oplus 1})) \qquad \text{and } l \oplus 1 = l + 1 \bmod 33$$

The edge direction matching function is formulated as follows:

$$\mathcal{U}(\bar{\mathbf{m}}_c(u, v, s)) = \sum_{l=1}^{G} \frac{\mathcal{U}_l(\bar{\mathbf{m}}_c(u, v, s))}{G} \tag{3.4}$$

where

$$\mathcal{U}_l(\bar{\mathbf{m}}_c(u, v, s)) = \frac{\sum_{(i,j)\in\mathbf{g}_l} \left[\| \nabla I(i,j) \| \, h\left(\frac{\mathbf{g}_l \cdot \nabla I(i,j)}{\| \mathbf{g}_l \| \| \nabla I(i,j) \|} \right) \right]}{N_{\mathbf{g}_l}} \tag{3.5}$$

and $N_{\mathbf{g}_l}$ represents the number of gradient pixels involve in segment $\mathbf{g}_l$. Component $h(.)$ is formulated as follows:

$$h(x) = \begin{cases} \dfrac{2}{1 + \exp(10\left(|x| - \frac{1}{2}\right))} - 1 & \text{if} \quad |x| \leq 1 \\ 0 & \text{if} \quad |x| > 1 \end{cases} \tag{3.6}$$

The first component of the nominator of Equation 3.5, $\| \nabla I(i,j) \|$, represents the magnitude of the gradient pixel (i, j). The second component measures the angle between $\nabla I(i,j)$ and model segment $\mathbf{g}_l$. The denominator of the function normalizes the result. The function, $h(x)$ attains its maximum value when x is equal to zero, indicating that $\nabla I(i,j)$ is perpendicular to model segment $\mathbf{g}_l$. Equation 3.5 attains its maximum value if component $\| \nabla I(i,j) \|$ and $h(x)$ are maximum.

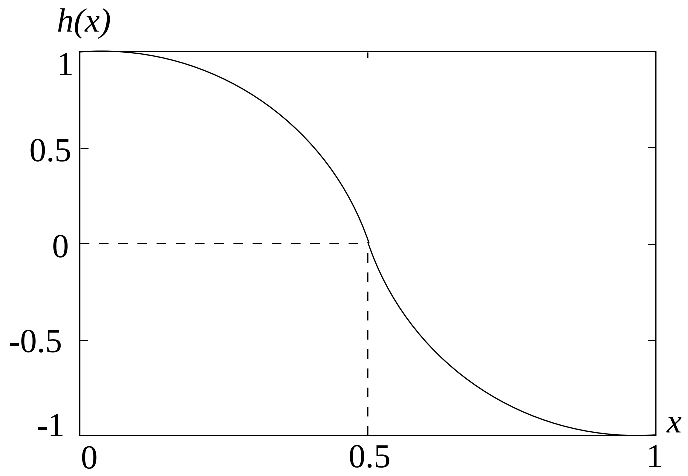


Figure 3.3: The image above shows the plot of $h(x)$.

3.2.2 Edge Point Matching

The goal of edge point matching is to measure the degree of resemblance between the points on model $\bar{\mathbf{m}}_c(u, v, s)$ and the human figure in the edge map. This function measures the percentage of the model points that have at least one edge pixel in their neighbourhoods. The function attains its maximum value when the model is placed right on top of the image's edge map area that contains edge pixels. For this purpose, we design the edge point matching function based on the idea used for Hausdorff distance measurement algorithm [23]. In order to explain our function, we will start by giving a short review of the original Hausdorff distance measurement algorithm. This is followed by an explanation on how we modify the algorithm to suit our purpose.

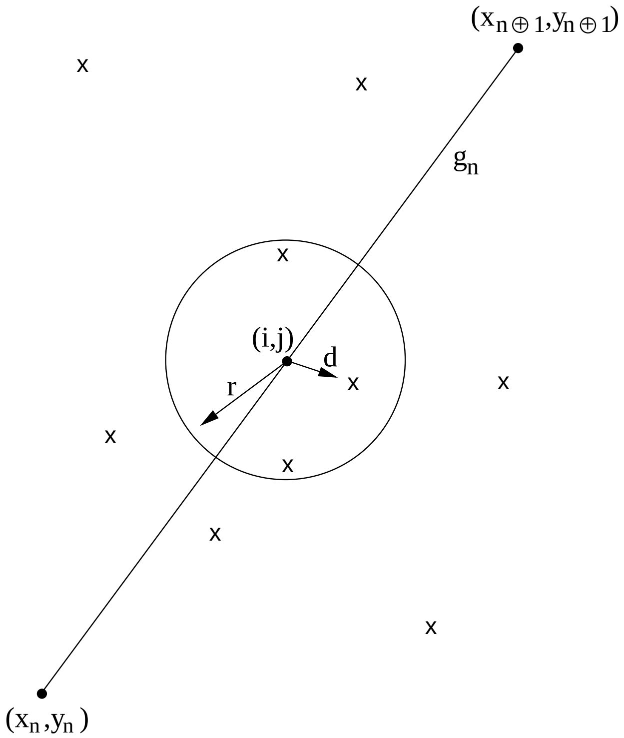


Figure 3.4: The image above illustrates the edge point matching process. The solid line represents segment n of model $\bar{\mathbf{m}}_c(u, v, s)$ and the crosses denote the edge pixels in the edge map. In the picture, r represents the maximum distance allowed for an edge pixel to be accounted in the function, while d represents the distance between the model point and its nearest neighbour. The circle represents the neighbour area of the model point.

Hausdorff distance is a well known measurement used to determine the degree of resemblance between two sets of points A and B that are at fixed positions with respect to one another. Given two sets of points $A = \{\mathbf{a}_i, \ldots, \mathbf{a}_p\}$ and $B = \{\mathbf{b}_i, \ldots, \mathbf{b}_p\}$, the original Hausdroff algorithm measures the distance as follows:

$$H(A, B) = \max(h(A, B), h(B, A)) \tag{3.7}$$

where

$$h(A, B) = \max_{a \in A} \min_{b \in B} \| a - b \| \tag{3.8}$$

Equation 3.8 identifies the point $a \in A$ that is farthest from any point in B and measures the distance from a to its nearest neighbour in B. The distance is defined as directed Hausdorff distance from A to B. Function $H(A, B)$ defines the largest value among $h(A, B)$ and $h(B, A)$ to be the degree of mismatch between the two sets.

Instead of only considering one point in measuring the degree of resemblance between two sets, we use Equation 3.8 to measure the distance of all points in the model to their nearest neighbours in the image edge map. The degree of resemblance between the model and the human figure in the image is defined as the percentage of the points in the model that have an edge pixel as their neighbour within an acceptable distance r. Based on our experiment we define $r = 5$ as it give us the best system performance. Using this criteria, our function reaches its maximum value if the model lies exactly on top of the human figure which must also have well defined edges.

However, in images taken in the real-life environment, most of the time the edges of the human figure in the image is not complete. Therefore, to measure the degree of resemblance, it is reasonable for us to only consider some points of the model whose distances to the edge pixels are the shortest. Let $\bar{\mathbf{m}}_c(u, v, s)$ be a model that consists of P points (equal to the number of pixels along the model contour) and d_p be the distance of point $p \in \bar{\mathbf{m}}_c(u, v, s)$ to its nearest edge neighbour. We sort all the distances $\{d_1, \ldots, d_P\}$ such that $d_1 < d_2 < \cdots < d_P$ and consider only the first K points where $K < P$. Based on our experiment K equal to 95 percent of P give us the best result.

The degree of resemblance between the model and the human figure in the image is defined as follows:

$$\mathcal{H}(\bar{\mathbf{m}}_c(u, v, s)) = \frac{\sum_{i=1}^{K} \mathcal{C}(d_i)}{K} \tag{3.9}$$

where

$$C(x) = \begin{cases} 1 & \text{if} \quad x \leq r \\ 0 & \text{if} \quad x > r \end{cases} \tag{3.10}$$

and K represents the number of points considered. The distance d is measured using Eucledian distance.

3.2.3 The Complete Human Figure Detection

Combining edge direction matching function and edge point matching function, the complete match measurement function used in the human figure detection process is defined as follows:

$$\mathcal{F}_{\mathcal{D}}\left(\bar{\mathbf{m}}_c\left(u,v,s\right)\right)=\mathcal{U}\left(\bar{\mathbf{m}}_c\left(u,v,s\right)\right)\mathcal{H}\left(\bar{\mathbf{m}}_c\left(u,v,s\right)\right) \tag{3.11}$$

where $\mathcal{U}\left(\bar{\mathbf{m}}_c\left(u,v,s\right)\right)$ represents the edge direction matching function and $\mathcal{H}\left(\bar{\mathbf{m}}_c\left(u,v,s\right)\right)$ represents the edge point matching function. Notice in Equation 3.11 that during the human figure detection process, cluster centroid $\bar{\mathbf{m}}_c$ undergoes translation of u pixels and v pixels in the x- and y-directions, and scaling of scaling factor s. To find the combination that produces the highest match value, we use an exhaustive search approach. The search ranges in the x- and y-directions are determined by the size of the model used and the size of the image. The criterion is that no part of the model is allowed to be positioned outside of the image boundary. Scaling factor s is set to allow the size of the model to increase and decrease by a maximum of 10 percent of the average size of the human figures found in the image database.

3.3 Human Figure Registration

The goal of the human figure registration process is to create models that have similar appearance with the human figures found in the image. This process refines the detection result produced by the human figure detection process, by deforming a model to fit a human figure contour in the image. During the process, the model undergoes a translation, scaling and deformation transformation.

There are three measurement functions involved in human figure registration process:

1. Edge direction matching function

2. Edge point matching function

3. Deformation measurement function

Since edge direction matching function and edge point matching function have been explained in the previous section, they will not be discussed again here.

Deformation measurement function measures how well the deformed model resembles a human figure by computing the distance between the deformed model and its cluster centroid. The distance is measured using Mahalanobis distance. This function is needed to prevent the model from deforming into an ambiguous shape. Maintaining the shape of the model is very important since it is possible for the model to deviate from the shape of a human figure and fits into a non-human object in the image. If this condition occurs during human figure registration process, it will lead to a wrong registration result.

3.3.1 Deformation Measurement Function

Since our models are derived from clustered data distribution, we can determine how well the deformed model resembles a human figure by measuring the Mahalanobis distance between the deformed model and its original cluster centroid. Let $\mathbf{m}_c^*$ be a deformed model and $\bar{\mathbf{m}}_c$ be the cluster centroid, the Mahalanobis distance is defined as follows:

$$\mathcal{M}\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right) = \left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right)^T \boldsymbol{\Sigma}_c^{-1} \left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right) \tag{3.12}$$

where $\boldsymbol{\Sigma}_c$ represents the covariance matrix of cluster c.

Due to the high dimensionality of model space and the limitation of data sample involved in deriving the model, it is dificult for us to recover the full covariance matrix of each cluster accurately [33]. In order to simplify the distribution, we reduce the dimensionality of the covariance matrix based on its most significant eigenvectors. The number of significant eigenvectors of each cluster is defined by performing Principal Component Analysis on the cluster. The detailed algorithm of Principal Component Analysis has been discussed in Chapter 2. Referring to Table 2.1, the number of significant eigenvectors for each cluster is different. To measure the Mahalanobis distance using covariance matrix whose dimension has been reduced, we use 2-Value Distance Metric function [33].

2-Value Distance Metric function consists of two components:

1. A component that measures the distance between the deformed model and its cluster centroid within the reduced dimensional sub-space.

   This component accounts for the shape difference between the deformed model and its cluster centroid in the directions of the cluster's most significant eigenvectors. The measurement process is very similar to the normal Mahalanobis distance, except for the size of the covariance matrix used. The distance $\mathcal{D}_1\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right)$ is defined as follows:

$$\mathcal{D}_1\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right) = \left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right)^T \mathbf{\Sigma}_{c,t}^{-1}\left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right) \tag{3.13}$$

   where $\mathbf{\Sigma}_{c,t}$ represents the covariance of cluster c in a t-dimensional subspace. Using eigenvector matrix $\mathbf{E}_c$ and eigenvalue matrix $\mathbf{\Lambda}_c$ of cluster c, Equation 3.13 is rewriten as follows:

$$\begin{aligned}\mathcal{D}_1\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right) &= \left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right)^T \mathbf{\Sigma}_{c,t}^{-1}\left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right) \\ &= \left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right)^T \left[\mathbf{E}_c\mathbf{\Lambda}_{c,t}^{-1}\mathbf{E}_c^T\right]\left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right) \\ &= \mathbf{b}^T\mathbf{\Lambda}_{c,t}^{-1}\mathbf{b}\end{aligned}$$

   where $\mathbf{b}$ is a t-dimensional deformation parameter vector derived from Equation 2.4:

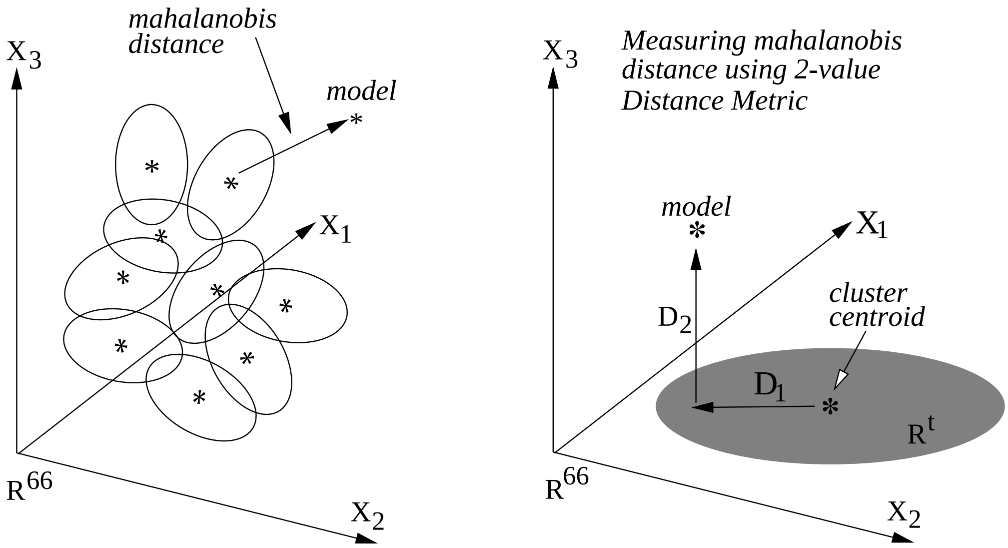
$$D_{\mathbf{b}}\left(\bar{\mathbf{m}}_c\right) = \bar{\mathbf{m}}_c + \mathbf{E}_c\mathbf{b}$$

$$\mathbf{m}_c^* = \bar{\mathbf{m}}_c + \mathbf{E}_c\mathbf{b}$$

$$\mathbf{b} = \mathbf{E}_c^T\left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right)$$

   Because of the diagonalized form, $\mathcal{D}_1\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right)$ can also be expressed as follows:

$$\mathcal{D}_1\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right) = \sum_{i=1}^{t}\frac{b_i^2}{\lambda_i} \tag{3.14}$$



(a) (b)

Figure 3.5: The image above illustrates the Deformation Measurement Function. Figure (a) illustrates the Mahalanobis distance between deformed model $\mathbf{m}_c^*$ and its original cluster centroids $\bar{\mathbf{m}}_c$. Figure (b) illustrates how we measure the distance using 2-Value Distance Metric function. The grey area represents a sub-space spanned by the t most significant eigenvectors of cluster c, while D_1 and D_2 represent the first and the second component of the 2-Value Distance Metric respectively.

2. A component that measures the distance between the deformed model and its projection in the reduced dimensional sub-space.

   This distance component accounts for the shape difference along other eigenvector directions between the deformed model and its cluster centroid that are not captured by the first component. Since we may not have a reasonable estimate of the smaller eigenvalues, we simply assume an isotropic Gaussian distribution along the smaller eigenvector directions. Therefore, instead of using Mahalanobis measurement, we use a Euclidean distance measurement which can be expressed as follows:

$$\mathcal{D}_2\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right) = \parallel \left(I - \mathbf{E}_c\mathbf{E}_c^T\right)\left(\mathbf{m}_c^* - \bar{\mathbf{m}}_c\right) \parallel \tag{3.15}$$

   where $\mathbf{E}_c$ is an eigenvectors matrix which consists of t significant eigenvectors of cluster c.

Let T be the size of the original model space dimension, and t_c be the number of significant eigenvectors in cluster c, the 2-Value Distance Metric function is defined as follows:

$$\mathcal{M}_{\mathcal{R}}\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right) = \mathcal{D}_1\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right) + \frac{T - t_c}{\sum\limits_{i=t_c+1}^{T} \lambda_i} \mathcal{D}_2\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right) \tag{3.16}$$

where $\mathcal{D}_1\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right)$ and $\mathcal{D}_2\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right)$ represents the first and the second component of the 2-Value Distance Metric function respectively. λ_i represents the i-th eigenvalue of the cluster. Notice that in this function, the better the model resembles a human figure, the lower is the $\mathcal{M}_{\mathcal{R}}\left(\mathbf{m}_c^*, \bar{\mathbf{m}}_c\right)$ value. This function attains its minimum value when the deformed model $\mathbf{m}_c^*$ and the cluster centroids $\bar{\mathbf{m}}_c$ are exactly the same, or the model is actually not deformed.

3.3.2 The Complete Human Figure Registration

Combining all the measurement functions involved in this process, the complete match measurement function used in human figure registration process is defined as follows:

$$\begin{aligned} &\mathcal{F}_{\mathcal{R}}\left(\bar{\mathbf{m}}_c\left(u, v, s, \mathbf{b}\right)\right) = \\ &\mathcal{U}\left(\bar{\mathbf{m}}_c\left(u, v, s, \mathbf{b}\right)\right) \mathcal{H}\left(\bar{\mathbf{m}}_c\left(u, v, s, \mathbf{b}\right)\right) \left(\mathrm{C} - \mathcal{M}_{\mathcal{R}}\left(\bar{\mathbf{m}}_c\left(u, v, s, \mathbf{b}\right)\right)\right) \end{aligned} \tag{3.17}$$

where $\mathcal{U}\left(\bar{\mathbf{m}}_c\left(u, v, s, \mathbf{b}\right)\right)$ represents the edge direction matching function, $\mathcal{H}\left(\bar{\mathbf{m}}_c\left(u, v, s, \mathbf{b}\right)\right)$ represents the edge point matching function, and $\mathcal{M}_{\mathcal{R}}\left(\bar{\mathbf{m}}_c\left(u, v, s, \mathbf{b}\right)\right)$ represent deformation measurement function. Notice in Equation 3.17 that during the human figure registration process cluster centroid $\bar{\mathbf{m}}_c$ undergoes translation of u pixels and v pixels in the x- and y-directions, scaling with scaling factor s, and deformation of amount $\mathbf{b}$. To find the combination that produces the highest match value, we use an exhaustive search approach similar to that described in Section 3.2.3.

3.4 Summary

Chapter 3 provides detailed explanations of our detection and registration processes, as well as the measurement functions involved in the processes. There are three measurement functions introduced, namely edge direction matching function, edge point matching function and deformation measurement function.

Human figure detection process produces an initial guess of the location and the size of the human figure detected. In this process, cluster centroid $\bar{\mathbf{m}}_c$ undergoes translation of u pixels and v pixels in the x- and y-directions, and scaling with scaling factor s, in order to find the combination that produces the highest match value.

Human figure registration process produces a model whose contour fits the human figure contour in the image by refining the initial guess produced by the detection process. In this process, cluster centroid $\bar{\mathbf{m}}_c$ undergoes translation of u pixels and v pixels in the x- and y-directions, scaling with scaling factor s, and deformation of amount $\mathbf{b}$ in order to find the combination that produces the highest match value.

Chapter 4

Experimental Results and Analysis

This chapter explains in detail the experiments conducted to measure the performance of our system. Two experiments were conducted:

1. **Human figure detection experiment.** This experiment measured the performance of our system in detecting human figure appearance in the images. The performance was measured by analyzing the system's initial guess of human figure's size and location in the image.

2. **Human figure registration experiment.** This experiment measured the performance of our system in deforming the model to fit the human body detected in the detection process. The performance was measured by comparing the deformed model produced with the shape of the real human figure detected. In order to measure the performance of this process independently of the performance of the detection process, we decided to perform the experiment by using only the good detection results.

Our system performance was then judged based on the analyzed results of both experiments. These experiments were conducted on Linux platform using various Intel processor computers ranging from PII-350 to PIII-800.

Figure 4.1: The images above highlight the variation of human shapes and poses as well as the variation of background scenes and lighting conditions found in our testing data. Images D, G, H, and I show slightly rotated human figures. Images A, B, C, F, and J show humans wearing a long dress, overcoat, or carrying a bag. In these images, some parts of the human bodies are covered. Images C, D, H, and L have relatively complex scenes.

A database of 807 grey-level images was used in both experiments. The image size varied from 64x128 to 400x600 pixels. Each picture contains the frontal or the rear view of standing human figures. Most of the pictures were gathered from a database of real life pedestrian pictures taken at various places and under varying lighting conditions. Some samples of the images are shown in Figure 4.1.

The detailed setting used in conducting each experiment and the rationale behind the measurement criteria used in measuring the system performance are explained separately in Section 4.1.1 and Section 4.2.1. The results of both experiments are also discussed and analyzed separately in Section 4.1.2 and Section 4.2.2

4.1 Human Figure Detection Experiment

The detection process detects the human figure appearance in an image. It also predicts the location and the size of the human figure detected. The result is represented as a *detection box* surrounding the human figure detected. This section discusses the criterion used in measuring the system's performance as well as the experimental results.

4.1.1 Measurement Criterion

In measuring the performance of the detection process, we set up a goodness criterion as follows:

1. **Good detection result.** The system was considered as giving a good detection result if and only if it correctly detected a human figure in the image and correctly predicted the size of the human figure detected.

2. **Bad detection result.** The system was considered as giving a bad detection result if it correctly detected a human figure in the image but it gave a wrong prediction of the size of the human figure detected.

3. **False detection result.** The system was considered as giving a false detection result if it detects a non-human object as a human figure

To help us differentiate good and bad detection results, we defined an additional goodness criterion. Let w_l be the distance between the leftmost point of an object's smallest bounding box and the left side of the detection box, and w_r be the distance between the rightmost point of the object's bounding box and the right side of the detection box. Let us also consider l_t to be the distance between the topmost point of the object's bounding box and the top side of the detection box, and l_b to be the distance between the bottommost point of the object's bounding box and the bottom side of the detection box. If the object's bounding box had the length of L and width of W, we defined the goodness criterion $\mathcal{G}_\mathcal{D}$ as follows:

$$\mathcal{G}_\mathcal{D} = \begin{cases} \text{good} & \text{if} \quad (w_r + w_l \leq 0.05W) \bigwedge (l_t + l_b \leq 0.05L) \\ \text{bad} & \text{if} \quad (w_r + w_l > 0.05W) \bigvee (l_t + l_b > 0.05L) \end{cases} \tag{4.1}$$

We believed that it was reasonable to measure the performance using the above-mentioned criterion since the criterion guarantees that the detection box encloses the human figure, and therefore relevant to the objective of the detection process which was to detect and locate the appearance of human figure in the image and to provide an initial guess of the size and the location of the human.

Prior to the experiment, we defined u_i, v_i and s_i as the initial parameters. Parameters u_i and v_i were defined based on the size of the picture, which were the topmost and the leftmost possible model position in the image. The scaling factor s_i was defined manually based on the prior knowledge of the estimated human body size in the image. We believed that it was reasonable for us to estimate the human body size prior to the experiment because even in a practical application, one would also have to define the range of the object size one wanted to detect.

During the experiment, we performed translation of the model by u and v pixels in the x- and y-directions respectively to all possible model positions in the image. The model was scaled by a scaling factor s, ranging from s_{min} to s_{max}, which were defined manually prior to the experiment. We used an exhaustive search approach similar to that described in Section 3.2.3, to find the combination of u, v, and s that gave us the best result.

4.1.2 Experiment Results

A total of 807 images were used in the human figure detection experiments. This number consisted of 210 training images and 597 testing images. The training images were the images used to build our data sample distribution. Recall that an image might contain more than one human figure. Due to false matches, it was also possible that there were more number of detections than the actual number of human figures in the testing set. The system's performance was measured based on 2 measurements, namely detection rate and detection accuracy. These 2 measurements were formulated as follows:

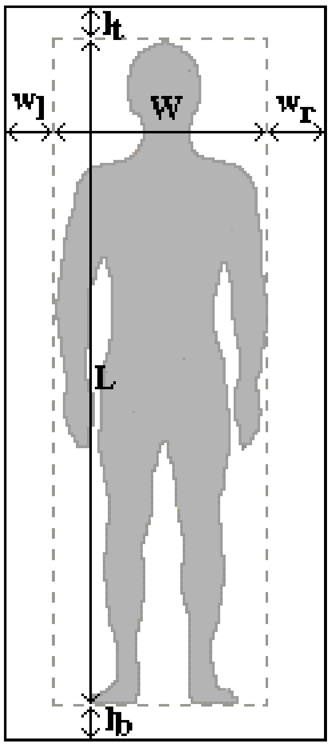


Figure 4.2: The image above illustrates how we measured the goodness criterion to differentiate between good and bad detection results. The box with dashed line represents the object's smallest bounding box, while the outer box with solid line represents the detection box. The distance between the inner and the outer boxes are represented by l_t, l_b, w_l and w_r. The width and length of the inner box are represented by L and W respectively.

$$detection\ rate = \frac{number\ of\ correct\ detections}{total\ number\ of\ human\ figures\ found\ in\ the\ image\ database}$$

and

$$detection\ accuracy = \frac{number\ of\ correct\ detections}{number\ of\ detections}$$

The human bodies that were accounted in measuring the detection rate were the ones that fulfill our object's criterion (please refer to Section 1.2).

Table 4.1 shows the detection rate of our human figure detection system. The table is divided into 3 column which shows the number of good detection results, the total number of human figures in the image database and the system's detection rate. The system's detection rate is measured as the ratio of good detection results to the total number of human bodies found in the image database. Table 4.2 shows the detection accuracy of our human figure detection system. The detection results are divided into three categories, namely good, bad, and false detections. The last column shows our system's detection accuracy which is measured as a ratio of the number of good detection to the total number of detections. In both tables, we separate the results of the detection experiment using training images from the result of detection experiment using testing images.

Table 4.1: The table shows the result of our human figure detection experiment. The detection rate is measured as a ratio of the number of good detections to the total number of detections.

	Number of good detections	Total number of human bodies found	Detection rate
Training data	198	210	94.28%
Test data	579	607	95.38%
Total	777	817	95.10%

Table 4.2: The table shows the result of our human figure detection experiment. The detection accuracy is measured as a ratio of the number of good detections to the total number of detections.

	Detection results			Detection
	Good	Bad	False	accuracy
Training data	198	9	8	92.09%
Test data	579	12	22	94.45%
Total	777	21	30	93.84%

The result shows that our system has a very good performance with 95.1 percent detection rate and 93.84 percent detection accuracy. The reason that the detection rate is slightly higher than the detection accuracy is because in measuring detection rate we neglected the bad and the false detection produced. Surprisingly, our detection accuracy and detection rate using its own training data are slightly lower than its performance using non-training data. Although theoretically it is very unlikely, we found that this result was still possible since we selected the training data randomly from the same set of data. The average time required to complete the human figure detection process for a 128x64 image is 98.1 seconds.

Figure 4.3 shows some examples of the results of the detection experiment. The results show that our system was capable of detecting human figure in various shapes, lighting conditions, and image sizes (4.3(a) - 4.3(k)). The results also show that the system was capable of detecting human figures in cluttered scenes (4.3(g), 4.3(j), 4.3(k)). Figure 4.3(k) shows that our system was capable of handling multiple human figures in one image. Some of the pictures even show that it was capable of handling slightly out of focus images (4.3(f) and 4.3(h)).

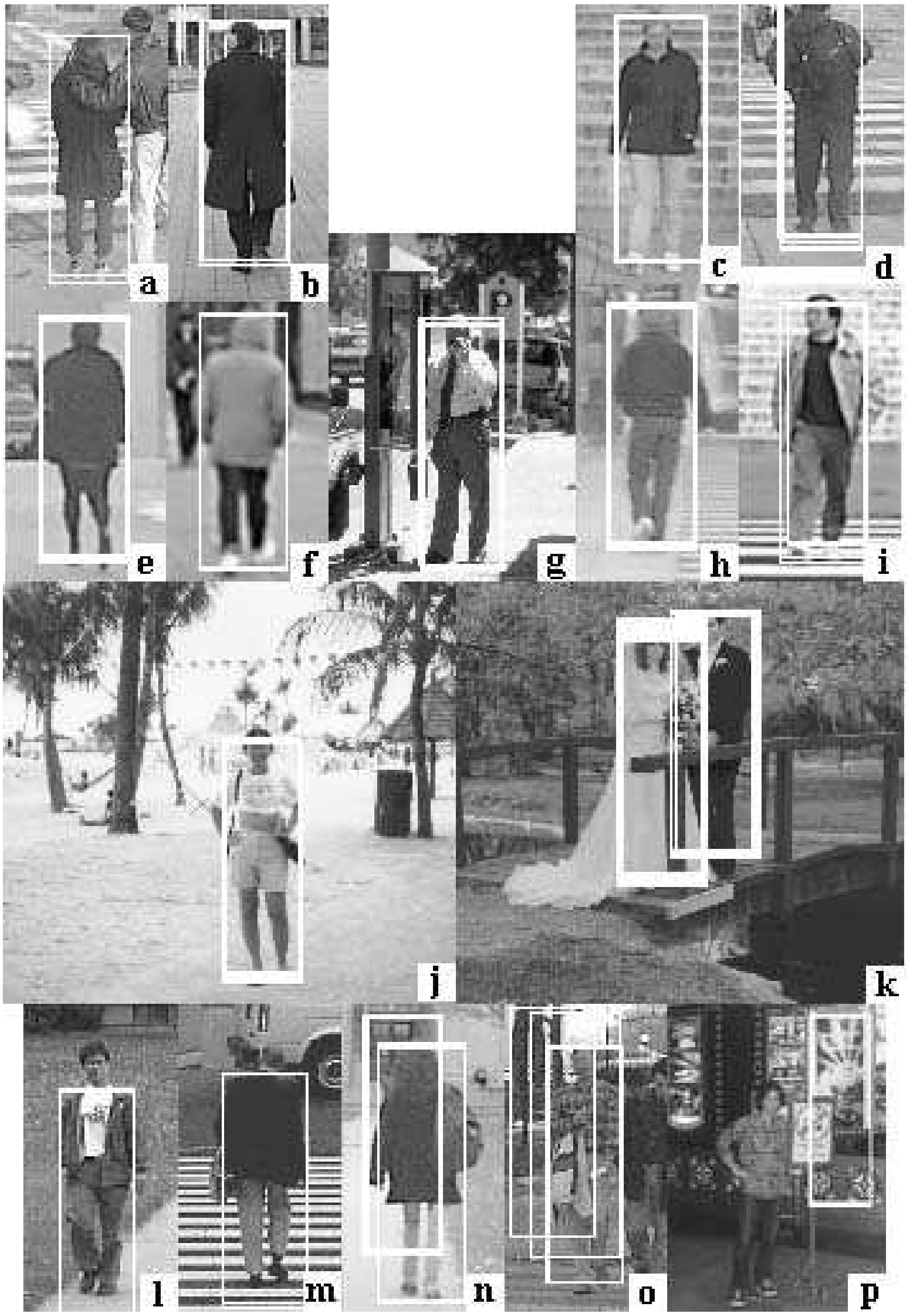


Figure 4.3: The images above show some samples of our human figure detection results. Image (a) to (k) show examples of good detection results, and image (l) and (m) show examples of bad detection results. Image (n) and (o) show the condition where the detetion experiment produced both good detection results and false detection results in one image. Image (p) shows false detection result produced by the system.

As we can see in Figure 4.3(l) and 4.3(m), most of our bad detection results were caused by the system's failure in detecting the head location of the human figures. We believed that these happened due to the lack of edge information at the top and bottom parts of the human figures. The size of human head and feet also made our system sometimes mistakenly define them as parts of the cluttered background.

4.2 Human Figure Registration Experiment

As described in Chapter 3, the goal of the human figure registration system is different from that of the detection system. Here we are more concerned with capturing the contours of the human figures detected, and therefore we use a different criterion to measure the system's registration performance.

4.2.1 Measurement Criterion

The criterion applied to human figure registration experiment is described as follows:

1. **Good registration result.** The result was considered as a good registration result if and only if it correctly ressembled the shape of the human figure detected and the mean distance of all of its control points to its designated landmarks in the human figure was less than 3 pixels. We believe that this was a reasonable criterion as it constrained the deformed model to fit a human figure contour, and was therefore relevant to the objective of human figure registration.

2. **Bad registration result.** The result was considered as a bad registration result if it did not satisfy the above criterion.

As mentioned previously in the beginning of Chapter 4, to make sure that the performance of the detection process did not affect the performance of the registration process, we only performed registration using initial parameters from good detection results. Therefore, unlike the criterion used in measuring detection results, there were no false registration category defined.

Prior to the experiment we defined initial parameters u_i, v_i and s_i based on the system detection results. During the experiment we performed translation on the model by u and v pixels in the x- and y-directions respectively. The model was also scaled by the scaling factor s. The range of u, v, and s were defined manually prior to the experiment.

The accuracy of our human registration algorithm was defined as the ratio of the number of good registration results to the total number of registration processes performed.

4.2.2 Experiment Results

A total of 571 images with 571 good initial detection results were used in human figure registration experiments. This number consisted of 194 training images and 377 testing images. The training images were the images used to build our data sample distribution. Table 4.3 shows the results of our human figure registration experiment. The registration results were classified into two categories, namely good and bad registration. We separated the registration results for the training images from that of the testing images. The last column shows our system's accuracy which was measured as a ratio of the number of good registration results to the total number of registration processes performed.

The result shown in Table 4.3 indicates good performance with the system's registration accuracy at 84.23 percent. As in the detection experiment, the system's performance using its own training data was slightly lower than the performance using testing data. The average time required to complete the human figure registration process for a 128x64 image is 340.7 seconds.

Table 4.3: The table below shows the result of our human figure registration experiment. The system's accuracy was measured as a ratio of the number of good registration results to the total number of registration processes performed.

	Registration results		System's accuracy
	Good	Bad	
Training data	162	32	83.50%
Test data	319	58	84.61%
Total	481	90	84.23%

Referring to Figure 4.4, we could see that the system successfully deformed the model shapes to fit the human figure contours. As for the bad registration results, we see that although it missed several parts of the human body such as the leg and the hand, the system was able to deform most of the other body parts to fit the human shapes. In these cases, usually the "missing" parts were represented by other parts of the human model. For example, if both legs were superimposed on one another, sometimes the system would register both legs as one leg, and therefore unable to find the other leg of the human figure.

We conclude that the performance of our registration system is slightly lower compared to the performance of the detection process. However, considering the limited number of the control points used in our training data to represent human shape contour, we found that the result is very promising.

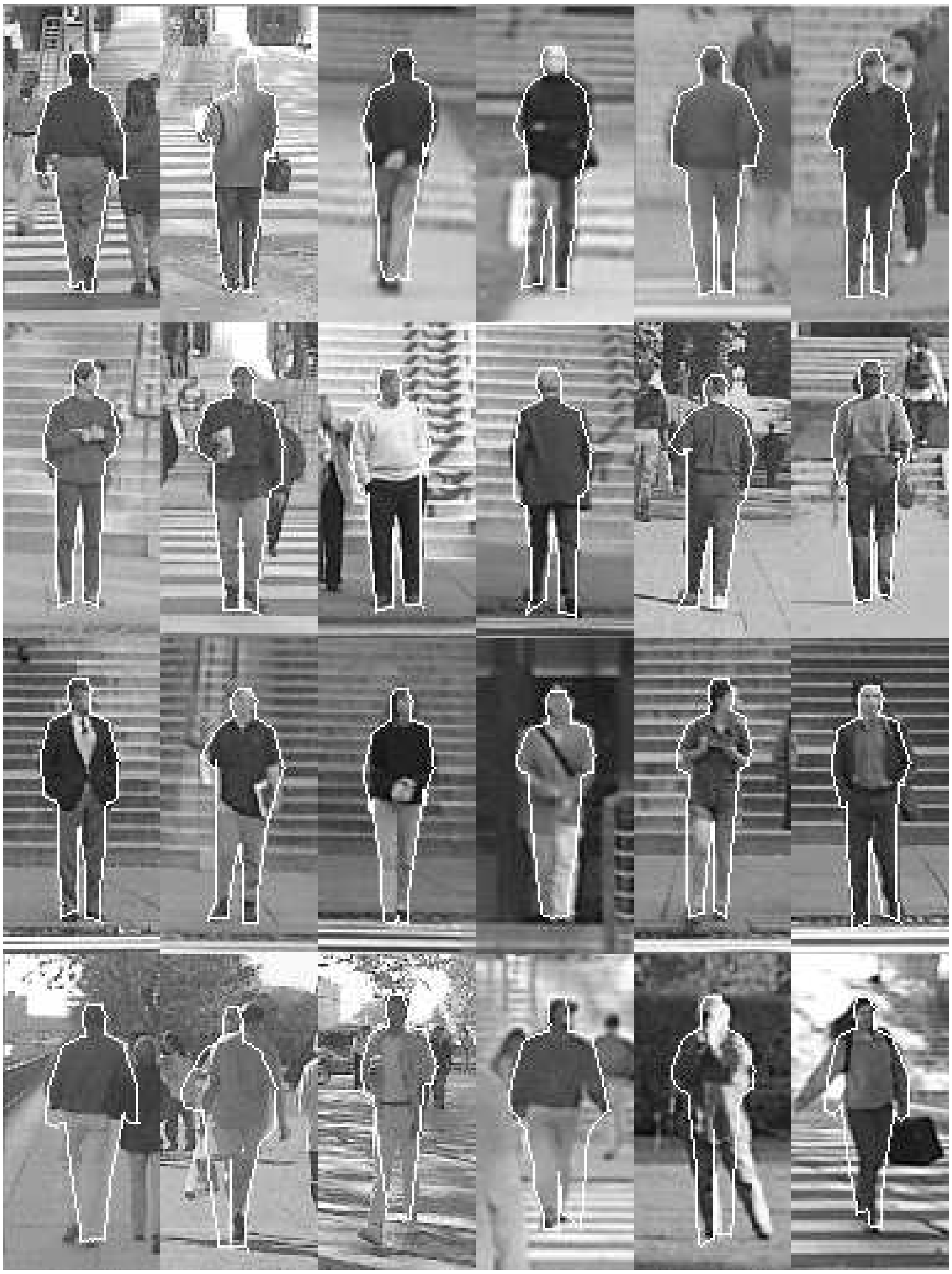


Figure 4.4: Sample of the human figure registration results. Images on the first three rows show examples of good registration results, while the images on the last row show examples of bad registration results. We could see that our registration system was capable of deforming the models to fit the human figure contours detected on the images

Chapter 5

Conclusion

5.1 Discussion and Future Direction

In this thesis, we have succesfully built a simple 2-D model that represents human figures which are non-rigid 3-D objects. The model is built based on manually marked human figure data samples. The advantage of the model is that one can further improve the model's performance by adding training data without having to rebuild the overall system.

Analyzing the overall system performance, we found some drawbacks and limitations which can be overcome by introducing and implementing new methods in our system.

1. **Unweighted edge matching.** In most cases, bad detection results were caused by the system's failure in detecting small parts of the human body, such as the head and the feet, since their edge information is not as strong as that of other parts of the human body. By defining a new weighting algorithm to represent the possibilities of edge appearance of each human body part, we believe that the system's accuracy can be increased. Furthermore, there are possibilities of implementing a similar weighting scheme to detect partially-occluded human figures in the images.

2. **Poor thresholding algorithm.** Since the detection process is based solely on the highest goodness value of match, we have to constraint the input

image to contain at least one human figure to be eligible for the detection process. One can eliminate this constrain by applying a better thresholding algorithm. One method is to learn the thresholding function based on the sample distribution of the goodness of match. This can also be used to reduce false detection rate.

3. **Time-consuming process.** One major drawback of our detection and registration system is the long time taken to complete the processes. If one apply an appropriate optimization algorithm, our method can be efficiently implemented.

5.2 Summary of Work

This thesis addresses the problem of detecting human figures in an image. In order to differentiate human body from non-human object, we have to formulate a model that is able to characterize the human figure. The model have to be robust enough to handle the variability of human body, and at the same time capable of differentiating human bodies from other non-human objects. However, it is difficult to formulate a good human figure model, since human body has high variability in size, shape and pose. To add to the complexity of the problem, the clothes and the accessories worn by the people produce unpredictable patterns and color combinations in human body. Furthermore, some parts of the human figures may be occluded.

In order to detect the appearance of human in images, we build a 2-D human shape model derived from manually marked real human figure images. Here, we overcome the difficulties in describing human shapes by introducing 9 prototypes that represent the main variants of the human figure's shapes and poses. For the matching purpose, we propose an edge-based model matching algorithm that is capable of detecting the appereance of human figures and deforming the models to fit the human figure contours detected in the images. The matching algorithm consists of 2 stages, namely, human figure detection and human figure registration.

We show that our system is capable of handling multiple human figures with various shapes and poses under various lighting conditions. The system is also capable of detecting human figures in cluttered background scenes. During verification experiments, our human figure detection system produces a detection accuracy of 93.84 percent and a detection rate of 95.10 percent. The human figure registration system produces a registration accuracy of 84.23 percent.

Based on the analysis of the system's performance, we believe that the overall system architecture is robust enough and therefore can be used to detect other non-rigid objects by re-training the system using different sets of data samples.

Bibliography

[1] Howard Anton. *Elementary Linear Algebra.* Wiley, New York, 7th edition, 1994.

[2] Adam Michael Baumberg. *Learning Deformable Models for Tracking Human Motion.* PhD thesis, The University of Leeds, October 1995.

[3] A.M. Baumberg and D.C. Hogg. Learning flexible models from image sequences. Technical report, Division of Artificial Intelligence, School of Computer Studies, University of Leeds, October 1993.

[4] A.M. Baumberg and D.C. Hogg. An efficient method for contour tracking using active shape models. Technical report, Division of Artificial Intelligence, School of Computer Studies, University of Leeds, April 1994.

[5] A.M. Baumberg and D.C. Hogg. Learning spatiotemporal models from training examples. Technical report, Division of Artificial Intelligence, School of Computer Studies, University of Leeds, March 1995.

[6] John F. Canny. Finding edges and lines in image. Technical Report 720, Artificial Intelligence Laboratory, MIT, June 1983.

[7] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *Proceedings of European Conference on Computer Vision*, pages 484–498, Springer, 1998.

[8] T.F. Cootes and C.J. Taylor. Active shape models. Technical report, Wolfson Image Analysis Unit, Dept. of Medical Biophysics, University of Manchester, Manchester M13 9PT, U.K., August 18 1998.

[9] Baumberg et al. An adaptive eigenshape model. In *Proceedings of the British Machine Vision Conference*, Birmingham, September 1995.

[10] David A. Forsyth and Margaret Fleck. Finding naked people. In *Proceedings of European Conference on Computer Vision*, pages 593–602, 1996.

[11] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Addison-Wesley, 1992.

[12] B. Heisele, U. Krebel, and W. Ritter. Tracking non-rigid, moving objects based on color cluster flow. In *Proceedings of IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, pages 257–260, 1997.

[13] B. Heisele and C. Wohler. Motion-based recognition of pedestrian. In *Proceedings of IEEE 14th International Conference of Pattern Recognition*, volume 2, pages 1325–1330, 1998.

[14] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. Comparing image using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9), September 1993.

[15] Daniel P. Huttenlocher and Clark F. Olson. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1), January 1997.

[16] Anil K. Jain. *Fundamentals of digital image processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.

[17] Ramesh Jain, Rangachar Kasturi, and Brian G. Schunck. *Machine vision*. McGraw-Hill, New York, 1995.

[18] H. M. Lakany and G. M. Hayes. An algorithm for recognising walker. In Josef Bigun, Gerard Chollet, and Gunilla Borgefors, editors, *Audio and Video based Biometric Person Authentication*, pages 112–118. IAPR, Springer, 1997.

[19] M.K. Leung and Yee Hong Yang. A model based approach to labelling human body outlines. In *Proceedings of the 1994 IEEE Workshop on Motion and non-rigid articulated objects*, pages 57–62, 1994.

[20] M.K. Leung and Yee Hong Yang. A human body outline labeling system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):359–377, April 1995.

[21] Stephen J. McKenna and Sahogang Gong. Non-intrusive person authentication for access control by visual tracking and face recognition. In Josef Bigun, Gerard Chollet, and Gunilla Borgefors, editors, *Audio and Video based Biometric Person Authentication*, pages 177–183. IAPR, Springer, 1997.

[22] Anuj Mohan. Object detection in images by components. Technical Report 1664, Artificial Intelligence Laboratory, MIT, June 1999.

[23] Clark F. Olson and Daniel P. Huttenlocher. Recognition by matching dense, oriented edge pixels. In *Proceedings of the International Symposium on Computer Vision*, pages 91–96, 1995.

[24] C. Papageorgiou, M. Oren, and Tomaso Poggio. Pedestrian detection using wavelet templates. In *Proceedings of IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 1997.

[25] C. Papageorgiou, M. Oren, and Tomaso Poggio. A general framework for object detection. In *Proceedings of IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, Bombay, India, 1998.

[26] Constantine Phaedon Papageorgiou. Object and pattern detection in video sequences. Master's thesis, Massachusetts Institute of Technology, May 1997.

[27] Alex Pentland and Baback Moghaddam. Probabilistic visual learning for object detection. In *Proceedings of IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, pages 786–793, 1995.

[28] Alex Pentland and Baback Moghaddam. Beyond eigenfaces: Probabilistic matching for face recognition. In *Proceedings of 3rd IEEE International Conference on Automatic Face and Gestur Recognition*, pages 30–35, 1998.

[29] Alex Pentland and Matthew A. Turk. Probabilistic visual learning for object detection. In *Proceedings of IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[30] Tomaso Poggio and S. Nayar. *Early Visual Learning.* Oxford University Press, 1996.

[31] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.

[32] K. Rohr. Incremental recognition of pedestrians from image sequences. In *Proceedings of IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, pages 8–13, 1993.

[33] Kah-Kay Sung. *Learning and Example Selection for Object and Pattern Detection.* PhD thesis, Massachusetts Institute of Technology, January 1996.

[34] K. Tabb, S. George, and R. Adams. Human shape recognition from snakes using neural network. In *Proceedings of the ICCIMA'99 Third International Conference on Computational Intelligence and Multimedia Applications*, pages 292–296, 1999.

[35] T. Tsukiyama and Y. Shirai. Detection of the movements of persons from a sparse sequence of tv images. In *Pattern Recognition, 18(3/4)*, pages 207–213, 1985.

[36] Christopher R. Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[37] Alan Yuille and Peter Hallinan. Deformable templates. In Andrew Blake and Alan Yuille, editors, *Active Vision*, pages 21–38. The MIT Press, Cambridge, Massachussetts, 1992.